

# Methoden zur geometrischen Suche in Proteinstruktursammlungen

Kumulative Dissertation

zur Erlangung des akademischen Grades

*Dr. rer. nat.*

an der Fakultät

für Mathematik, Informatik und Naturwissenschaften der  
Universität Hamburg

eingereicht beim Fach-Promotionsausschuss Informatik

von

**Joel Graef**

geboren in Hamburg

Hamburg, März 2024



Erstgutachter: Prof. Dr. Matthias Rarey  
Zweitgutachter: Prof. Dr. Andrew Torda  
Drittgutachter: Dr. Didier Rognan

Tag der Disputation: 04.06.2024



# Kurzfassung

Für die Funktion und Stabilität von Proteinen und Proteinkomplexen ist die intermolekulare Wechselwirkung zentraler Bestandteil. Durch sie werden biologische Prozesse in Gang gesetzt, reguliert oder gestoppt. Die Bestimmung und Erforschung dieser Wechselwirkungen in Bindestellen bildet deshalb ein Kernelement des computergestützten Wirkstoffentwurfs. Grundlage hierfür sind Protein-Ligand- und Protein-Protein-Komplexe, die in Datenbanken wie der öffentlich zugänglichen *Protein Data Bank* (PDB) vorliegen. Zur Analyse und systematischen Durchsuchung der stetig zunehmenden Anzahl verfügbarer Strukturen bedarf es effizienter Programme. Insbesondere die Suche nach ähnlichen Bindestellen ermöglicht die Vorhersage von Proteinfunktionen oder Interaktionspartnern. Dies basiert auf der Suche nach geometrischen Anordnungen von Atomen. Hierfür ist eine präzise Repräsentation der strukturellen und chemischen Eigenschaften vonnöten. Verfügbare Software zur systematischen Durchsuchung von Bindestellen deckt jedoch zumeist nur ein kleines Set von Eigenschaften ab und eignet sich nur eingeschränkt zur Analyse großer Datenmengen.

Zur Durchführung nutzerspezifischer Suchanfragen bei hoher Flexibilität und umfangreicher Datenabdeckung in Bindestellen wurden im Rahmen dieses Promotionsprojekts drei Verfahren entwickelt. Zunächst wurde eine Methode zur Detektion von Bindestellen kleiner Moleküle für die Anwendung auf große Strukturkollektionen optimiert und reimplementiert. Die Identifizierung von vergrabenen Stellen auf der Proteinoberfläche basiert auf der Diskretisierung des Proteins mithilfe eines Gitters und stellt eine der präzisesten geometrischen Bindetaschenvorhersagemethoden dar. Das zweite Verfahren generiert eine Datenbank der Geometrie und Eigenschaften dieser Bindestellen, die durch textuelle, numerische und geometrische Anfragen durchsucht werden kann. Die Angabe von textuellen und numerischen Eigenschaften wie des PDB-Codes oder der Bindestellentiefe ermöglicht die Reduzierung des Suchraums. Mithilfe geometrischer Anfragen können innerhalb kürzester Zeit ähnliche atomare Anordnungen gefunden werden. Anhand zahlreicher Beispiele zur Analyse von medizinisch relevanten Proteinkinasen konnte die Anwendbarkeit der Methode überzeugend demonstriert werden. Die Effizienz dieser Methodologie wird in der dritten Methode genutzt, um die Ähnlichkeit von Protein-Protein-Bindestellen zu bestimmen. Mithilfe automatisch generierter Suchanfragen können so verwandte Bindestellen erfolgreich erkannt und auf Grundlage der identifizierten ähnlichen Eigenschaften überlagert werden, wie ein Vergleich mit gängigen Methoden zeigt. Die im Rahmen dieser Arbeit entwickelten Software-Applikationen ermöglichen umfassende Analysen, die von der Vorhersage möglicher Bindungstaschen über die Analyse von Protein-Ligand-Interaktionen bis hin zur Ähnlichkeitssuche von Protein-Protein-Komplexen reichen.



# Abstract

Intermolecular interactions are key to the function and stability of proteins and protein complexes. Biological processes are initiated, regulated, or stopped by molecular interactions. The determination and investigation of these interactions in binding sites thus forms a fundamental element of computer-aided drug design. It is based on structures of protein-ligand and protein-protein complexes, which are available in databases such as the publicly accessible *Protein Data Bank* (PDB). Efficient programs are needed to analyze and screen the ever-increasing number of available structures. In particular, the search for similar binding sites enables the prediction of protein functions or interaction partners. For this purpose, geometric arrangements of atoms are searched for. This search requires a precise representation of the structural and chemical properties. However, available software for the screening of binding sites usually only covers a small set of properties and is only of limited suitability for analyzing large amounts of data.

To perform user-specific and individualized searches and extensive data coverage in binding sites, three methods were developed in the context of this thesis. First, a method for detecting binding sites of small molecules was optimized and reimplemented for the application to large structure collections. The identification of buried sites on the protein surface is based on the discretization of the protein using a grid. This method was shown to be one of the most accurate geometric binding pocket prediction methods. The second method generates a database of the geometry and properties of these detected binding sites, which can be searched by textual, numerical, and geometrical queries. The specification of textual and numerical properties such as the PDB code or the binding site depth enables the reduction of the search space. Using geometric queries, similar atomic arrangements can be found within reasonable time. The applicability of the method could be convincingly demonstrated by numerous examples for the analysis of pharmaceutically relevant protein kinases. The efficiency of this methodology is exploited in the third method to determine the similarity of protein-protein binding sites. With the help of automatically generated search queries, related binding sites can thereby be successfully detected and superimposed based on the detected similarities, as shown by comparing it with commonly used methods. The software applications developed as part of this work enable comprehensive analyses ranging from the prediction of possible binding pockets to the analysis of protein-ligand interactions and similarity searches in databases of protein-protein complexes.





# Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich während meiner Promotion unterstützt und begleitet haben. Als Erstes möchte ich Matthias Rarey für das interessante Promotionsthema, die ausgezeichnete Betreuung und seine fachliche Unterstützung danken. Des Weiteren danke ich Christiane Ehrt für die vielen hilfreichen Diskussionen und die gute Zusammenarbeit an mehreren Publikationen. Ebenfalls möchte ich mich bei Konrad Diedrich und Martin Poppinga für die Zusammenarbeit an der Methode GeoMine bedanken. Darüber hinaus danke ich dem BMBF für die Finanzierung des GeoMine-Projekts. Auch danke ich all meinen Kollegen aus der AMD-Arbeitsgruppe für die entspannte Arbeitsatmosphäre und die tollen Jahre zusammen. Insbesondere danke ich Jochen Sieg, Patrick Penner, Jonathan Pletzer-Zelgert und Christian Meyenburg für die vielen Diskussionen, die diese Arbeit vorangetrieben haben. Für das Korrekturlesen möchte ich mich besonders bei Christiane Ehrt, Jochen Sieg und Katrin Schöning-Stierand bedanken. Als Letztes möchte ich an dieser Stelle auch meinen Freunden und meiner Familie danken. Besonders danke ich meiner Mutter für ihr immer offenes Ohr und fortwährende Unterstützung.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	1
1.1	Proteinkomplexe und molekulare Interaktionen . . . . .	3
1.2	Suche in Proteinstruktursammlungen . . . . .	5
1.3	Motivation . . . . .	7
<b>2</b>	<b>Proteinbindetaschenvorhersage</b>	11
2.1	Verfahren zur Vorhersage von Proteinbindetaschen . . . . .	12
2.1.1	Strukturbasierte Verfahren . . . . .	12
2.1.2	Sequenzbasierte Verfahren . . . . .	16
2.1.3	Kombinierte Verfahren . . . . .	17
2.1.4	Maschinelles Lernen . . . . .	17
2.2	Proteinbindetaschenvorhersage mit DoGSite . . . . .	18
2.2.1	Methodische Zusammenfassung . . . . .	20
2.2.2	Evaluation . . . . .	23
2.2.3	Ausblick . . . . .	24
<b>3</b>	<b>Geometrische Mustersuche</b>	25
3.1	Herausforderungen der Suche nach ähnlichen Bindetaschen . . . . .	26
3.2	Verfahren zur Suche in Proteindatensammlungen . . . . .	27
3.2.1	Suchverfahren in Proteinstrukturen . . . . .	27
3.2.2	Suchverfahren in Proteinbindetaschen . . . . .	29
3.2.3	Vergleich von Suchverfahren . . . . .	32
3.3	Geometrische Mustersuche mit GeoMine . . . . .	34
3.3.1	Methodische Zusammenfassung . . . . .	36
3.3.2	Evaluation und Performanz . . . . .	42
3.3.3	Anwendungen . . . . .	44
3.3.4	Ausblick . . . . .	46

<b>4 Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen</b>	49
4.1 Vorhersage und Analyse von PPIs . . . . .	50
4.1.1 Maschinelles Lernen . . . . .	51
4.1.2 Hotspots und Bindungsstellen . . . . .	51
4.1.3 Docking und virtuelles Screening . . . . .	53
4.1.4 Netzwerkanalysen und sequenzbasierte Methoden . . . . .	54
4.2 Bestimmung der Ähnlichkeit von Protein-Protein-Bindestellen . . . . .	55
4.3 Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen mit PiMine . . . . .	57
4.3.1 Methodische Zusammenfassung . . . . .	58
4.3.2 Evaluation . . . . .	61
4.3.3 Ausblick . . . . .	65
<b>5 Zusammenfassung</b>	67
<b>Literaturverzeichnis</b>	69
<b>Literaturverzeichnis der kumulativen Dissertation</b>	89
<b>Betreute studentische Arbeiten</b>	91
<b>Anhang</b>	93
<b>A Wissenschaftliche Beiträge</b>	93
A.1 Publikationen . . . . .	93
A.2 (Inter-)nationale Konferenzbeiträge . . . . .	96
A.2.1 Vorträge . . . . .	96
A.2.2 Poster . . . . .	96
<b>B Weitere Analysen</b>	97
B.1 dogsite.v3 . . . . .	97
B.2 GeoMine . . . . .	98
<b>C Suchfilter in GeoMine</b>	105
<b>D Software-Architektur</b>	115
D.1 Bibliotheken . . . . .	117
D.2 Optimierung der Surface-Bibliothek . . . . .	126
D.3 Applikationen . . . . .	127

<b>E</b>	<b>Bedienung der Software</b>	129
E.1	dogsite3 . . . . .	129
E.2	GeoMine . . . . .	131
E.3	PiMine . . . . .	134
E.4	Proteins <i>Plus</i> -Server . . . . .	138
<b>F</b>	<b>Publikationen der kumulativen Dissertation</b>	141
F.1	Binding Site Detection Remastered: Enabling Fast, Robust, and Reliable Binding Site Detection and Descriptor Calculation with DoGSite3 . . . . .	141
F.2	Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures . . .	177
F.3	GeoMine: interactive pattern mining of protein–ligand interfaces in the Protein Data Bank . . . . .	203
F.4	ProteinsPlus: a comprehensive collection of web-based molecular modeling tools . . . . .	207
F.5	Database and Workflow Optimizations for Spatial-Geometric Queries in GeoMine . . . . .	213
F.6	Database-Driven Identification of Structurally Similar Protein-Protein Interfaces . . . . .	227
F.7	SiteMine: Large Scale Binding Site Similarity Searching in Protein Structure Databases . . . . .	261



# Kapitel 1

## Einleitung

Die Wirkstoffentwicklung ist eine komplexe und weltweite Herausforderung. Aufgrund der hohen Kosten, dem großen zeitlichen Aufwand und der enormen Datenmenge ist die Unterstützung mithilfe von computergestützten Analysen kaum mehr wegzudenken [1, 2]. Algorithmisches molekulares Design wird z. B. zur Auswertung von Hochdurchsatz-Screenings, also der automatisierten Suche nach potenziellen Modulatoren von Proteinfunktionen in einer riesigen Anzahl von Substanzen, der Analyse von und Suche in Proteinstrukturen, der Identifikation von Wirkstoffzielen oder der Vorhersage von Interaktionen zwischen Molekülen verwendet. Bereits frühe Phasen des Entwicklungsprozesses von Wirkstoffen werden so erleichtert und reduzieren Probleme im späteren Verlauf [3].

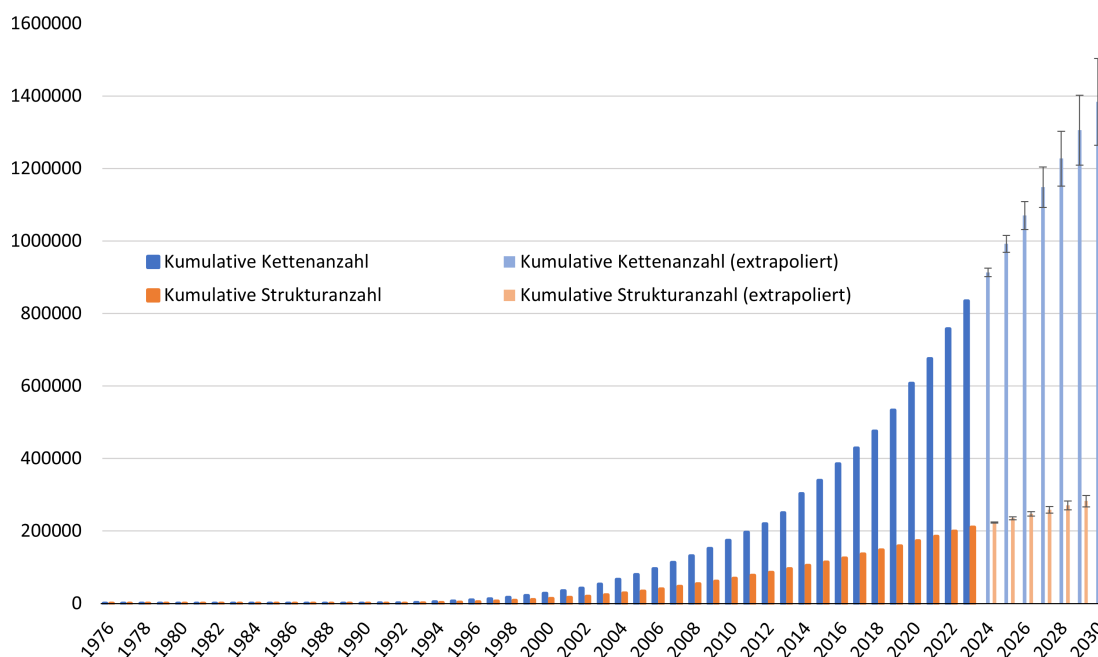
Eine zentrale Rolle für viele Programme zur Wirkstoffentwicklung nehmen Proteinstrukturen ein. Proteine sind unverzichtbar für viele Prozesse in lebenden Organismen. Sie übernehmen als Enzyme die Biokatalyse, regulieren die Funktion anderer Proteine, transportieren Substanzen und spielen als Antikörper eine entscheidende Rolle in der Immunantwort [4]. Zur Analyse werden Proteine als Sequenzen oder dreidimensionale (3D) Strukturen dargestellt. Letztere sind insbesondere deshalb nützlich, da angenommen wird, dass die Struktur eines Proteins in der Evolution länger überdauert als die Sequenz [5]. Somit kann beispielsweise die Struktur bei der Identifizierung der Funktion des Proteins helfen. Die erste Proteinsequenz wurde bereits 1951–1955 durch Frederick Sanger für das Hormon Insulin bestimmt und publiziert [6–10]. Die Frage nach der 3D-Struktur konnte damit allerdings noch nicht beantwortet werden. 1954 beobachtete Anfinsen jedoch, dass sich die Polypeptidkette der Ribonuklease in eine einzigartige 3D-Struktur falten kann [11, 12]. 1958 folgte die erste vollständige

Proteinstrukturbestimmung für Myoglobin durch Röntgenanalysen eines kristallisierten Proteins [13]. Bei dieser wird das Protein in hoher Konzentration durch Abkühlung oder Verdampfung des Lösungsmittels kristallisiert. Anschließend wird der Kristall einem Röntgenstrahl ausgesetzt und die durch Atome verursachte Lichtbrechung analysiert [14]. In den folgenden Jahren wurde die Röntgenanalyse durch technische Fortschritte detailreicher, sodass einzelne Atompositionen erkennbar wurden [15]. Weitere Analysemethoden wie die Kernspinresonanzspektroskopie, die Neutronenstreuungskristallographie und die (Kryo-)Elektronenmikroskopie folgten [16–19]. Diese technischen Durchbrüche erlauben immer höhere Auflösungen und weniger kostspielige Analysen. Auch gibt es seit kurzem computergestützte Struktur-Vorhersagemethoden wie das maschinelle Lernverfahren AlphaFold [20, 21] wodurch die Anzahl an vorhandenen Proteinstrukturmodellen erheblich gestiegen ist.

Proteinstrukturen werden zum Beispiel in Datenbanken wie der *Protein Data Bank* (PDB) [22] bereitgestellt. Diese erlauben eine effiziente Speicherung, einen zentralen Zugriff, einfache Suchmöglichkeiten und eine leichte Erweiterung der Daten [23]. Insbesondere letzteres ist von großem Vorteil angesichts des signifikanten Anstiegs neuer Strukturen. Programme zur Wirkstoffentwicklung können somit auf eine Vielfalt an Strukturen zugreifen und diese analysieren oder weiterverarbeiten. Eine der wichtigsten Informationen, die eine Proteinstruktur liefern kann, sind Bereiche an denen andere Moleküle binden können und deren Eigenschaften. Diese können z. B. verwendet werden, um auf die Funktion des Proteins zu schließen, die Affinität eines Moleküls zum Protein vorherzusagen oder Moleküle zu entwerfen, die dort binden können [24, 25]. Jede Proteinkette bildet im Schnitt vier biologisch relevante Bindetaschen aus [26]. Derzeit gibt es in der PDB etwa 835 000 Proteinketten in 211 000 Proteinstrukturen. 2030 können wir damit rechnen, dass 1,38 Millionen Proteinketten und 294 000 Proteinstrukturen gespeichert sind (siehe Abbildung 1.1). Dementsprechend gäbe es etwa 5,52 Millionen Bindetaschen in experimentell bestimmten Proteinstrukturen. Auch gibt es mehr als 214 Millionen vorhergesagte Proteinstrukturmodelle in der AlphaFold Proteinstrukturdatenbank [21]. Damit steigt der Bedarf an exakten, aber schnellen und effizienten Methoden zur Speicherung und Analyse dieser großen Datenmengen.

In der vorliegenden Dissertation werden drei Programme im Kontext der Proteinbindestellen beschrieben: Das Erste ermöglicht die Vorhersage der Bindestellen anhand von Strukturinformationen; mit dem Zweiten können Proteinstrukturen oder deren vorhergesagte Bindestellen – größtenteils bestimmt mithilfe des ersten Tools – in einer Datenbank sowohl mit textuellen und numerischen Anfragen gefiltert und mit geometrischen Mustern durchsucht werden; das Dritte erlaubt die Bestimmung der Ähnlichkeit





**Abbildung 1.1:** Entwicklung der kumulativen Anzahl an vorhandenen Proteinketten und Strukturen in der PDB von 1976 bis 2030. Die Anzahlen der Jahre 1976 bis 2023 wurden von der PDB übernommen [27], während die Werte der Jahre 2024 bis 2030 auf einer mit Microsoft Excel 365 berechneten Extrapolation beruhen.

von Bindestellen zwischen interagierenden Proteinen, den sogenannten Protein-Protein-Komplexe.

In den folgenden Unterkapiteln wird zunächst auf Proteine und molekulare Interaktionen eingegangen, um deren Bedeutung für die entwickelten Programme hervorzuheben. Im Anschluss werden Anwendungen zur Suche in Proteinstruktursammlungen vorgestellt und damit verbundene Problematiken diskutiert. Zuletzt wird diese Arbeit motiviert und die Struktur der Dissertation dargelegt.

## 1.1 Proteinkomplexe und molekulare Interaktionen

Proteine sind Makromoleküle, die aus über Peptidbindungen kovalent gebundenen Aminosäuren bestehen und Ketten bilden [4]. Oft bestehen Proteine aus mehreren identischen (Homomultimere) oder unterschiedlichen Ketten (Heteromultimere). Die Sequenz der Aminosäuren bedingt die Struktur des Proteins [4]. In der Natur gibt es über 140 Aminosäuren, bisher sind allerdings nur 20 bis 22 Aminosäuren bekannt, die in Proteinen von Lebewesen vorkommen [28]. Diese sind aus zwei Teilen aufgebaut: 1. ein gleicher Teil, das sogenannte Rückgrat bzw. Backbone, bestehend aus dem sogenannten

C $\alpha$ -Atom mit kovalent gebundener Carboxygruppe und einer Aminogruppe, und 2. ein unterschiedlicher Teil, der ebenfalls kovalent an das C $\alpha$ -Atom gebundenen Seitenkette [4]. Das Backbone hat zwei Enden bzw. Termini, welche über Peptidbindungen an weitere Aminosäuren gebunden sein können, während die Seitenkette für die Eigenschaften der Aminosäure verantwortlich ist [4].

Ihre Funktion üben Proteine durch die Bindung von beispielsweise kleinen Molekülen (im Weiteren auch Liganden genannt), anderen Proteinen oder Nukleinsäuren aus. Diese Bindung resultiert in einem sogenannten Proteinkomplex. Die Bindungsstelle ist im Falle einer Protein-Ligand Bindung auch als Proteinbindetasche bekannt. Die Interaktion zweier Bindungspartner findet über eine sterische und physikochemische Komplementarität statt [29, 30]. Die sterischen Eigenschaften beschreiben hierbei die geometrische Passform, während die physikochemischen Eigenschaften energetische Wechselwirkungen zwischen Atomen oder funktionellen Gruppen, z. B. elektrostatische oder van-der-Waals Wechselwirkungen, angeben.

Nicht-kovalente Bindungen zu anderen Molekülen werden durch die negative freie Bindungsenthalpie getrieben. Diese wird mittels der Formel  $\Delta G = \Delta H - T\Delta S$  berechnet und besteht aus einem entropischen Term  $S$ , einem enthalpischen Term  $H$ , sowie der Temperatur  $T$ . Der enthalpische Term beschreibt die innere Energie und die Volumenarbeit bzw. das Produkt des Drucks und Volumens eines Systems [31]. Er gibt damit die Spezifität und Stärke der molekularen Interaktionen wie der ionischen und elektrostatischen Wechselwirkungen oder der Wasserstoffbrückenbindungen an. Der entropische Term stellt hingegen ein Maß für die Dynamik des Systems bzw. für die Freiheitsgrade der Moleküle in der Bindestelle an [32].

Da ein Komplex nur dann gebildet wird, wenn dieser energetisch günstiger ist als der der Moleküle im ungebundenen Zustand, liegt ein Fokus der Wirkstoffentwicklung auf der Optimierung von Molekülinteraktionen bzw. -wechselwirkungen. Die bekanntesten Interaktionstypen sind die Wasserstoffbrückenbindungen, die eine elektrostatische Wechselwirkung eines Wasserstoffbrücken-Donors und Wasserstoffbrücken-Akzeptors darstellen, die Salzbrücken zwischen einem Kation und einem Anion, sowie die hydrophoben Wechselwirkungen. Letztere wirken im Gegensatz zu anderen Wechselwirkungen nicht direktional und werden z. B. zwischen aromatischen Systemen ausgebildet [33]. Neben diesen Interaktionstypen gibt es eine Reihe weiterer Interaktionen wie die Halogen-Brücken zwischen einem Halogenatom wie Chlor und Brom und einem elektronenreichen System [33].

## 1.2 Suche in Proteinstruktursammlungen

Es existieren diverse Proteinstruktursammlungen, die unter anderem helfen, den computergestützten Wirkstoffentwurf zu unterstützen und unser Wissen zu erweitern. Häufig sind diese Sammlungen in Form von Datenbanken auf eine Molekülgruppe spezialisiert. Die PDB beinhaltet beispielsweise hauptsächlich Proteinstrukturen. Von den derzeit etwa 210 000 Strukturen wurden die meisten mithilfe der Röntgenkristallographie bestimmt [27]. Die AlphaFold-Datenbank beinhaltet ebenfalls Proteinstrukturmodelle, welche allerdings mithilfe eines maschinellen Lernverfahrens für Proteinsequenzen vorhergesagt wurden. In der initialen Veröffentlichung von AlphaFold wurden bereits über 360 000 vorhergesagte Strukturen bereitgestellt [21]. Anstelle der großen Proteine finden sich die Strukturen kleiner Moleküle beispielsweise in der *Cambridge Structural Database* (CSD) [34]. Diese beinhaltet über 1 150 000 Moleküle aus Röntgenkristallographie- und Neutronenstreuungsanalysen. Auch gibt es Datenbanken wie die sc-PDB [35], welche Strukturen und Bindestellen bereitstellen, die speziell für die Wirkstoffentwicklung zusammengestellt wurden und Datenbanken wie BioGRID [36], die Protein-Protein-Interaktionsnetzwerke beinhalten.

Mithilfe der stetig wachsenden Anzahl bekannter Strukturen und der immer höheren Qualität ermöglichen die Daten eine nie dagewesene Datendiversität und damit neue Einblicke in biomolekulare Zusammenhänge. Die Nutzung dieses Wissens ist allerdings mit einigen Hürden verbunden. Wo früher zu wenig Strukturen bekannt waren, sind es heute so viele, dass computergestützte Verfahren unverzichtbar geworden sind. Ein wichtiger erster Schritt hierfür ist die Präprozessierung der Daten. So gibt es spezifische Herausforderungen in Abhängigkeit der zur Strukturbestimmung verwendeten Methode. In Röntgenkristallstrukturen als Hauptquelle der PDB sind beispielsweise nur die Teile des Moleküls aufgelöst, die Elektronen beinhalten. Wasserstoffatome können somit nur bei sehr hohen Auflösung dargestellt werden. Aufgrund der hohen Flexibilität der Moleküle kann es zudem dazu kommen, dass Teile des Proteins nicht aufgelöst werden können. Weiter können Kristallartefakte auftreten, bei denen z. B. Fremdkörper in den Kristall eingedrungen sind und mit der Probe in Kontakt treten [37].

Für die Präprozessierung gibt es bereits einige etablierte Methoden. So können beispielsweise die Wasserstoffatome, Aminosäureorientierungen und Protonierungszustände mithilfe des auch in den hier entwickelten Methoden verwendeten Programms Proton [38] oder mit dem Webserver H++ [39] bestimmt werden. Kristallkontakte bzw. die Unterscheidung zwischen echten Bindestellen und Kristallkontakten spielt insbesondere bei Protein-Protein-Interaktionsanalysen eine große Rolle. Vorhersagen, um welchen

Typ von Interaktion es sich handelt, können mithilfe einiger Tools wie mit PRODIGY-crystal [40] und HyPPI [41] getroffen werden.

Im Anschluss an die Präprozessierung können unterschiedlichste Analysen der Strukturen durchgeführt werden. Eine mögliche Analyse ist die Suche nach geometrischen Mustern, die z. B. verwendet wird, um ähnliche Strukturen zu finden und damit Rückschlüsse auf die Proteinfunktion treffen zu können. Programme wie DALI [42], CE [43] und STRUCTAL [44], die die globale Ähnlichkeit der Proteinstruktur berechnen, verwenden häufig die  $C\alpha$ -Atome der Aminosäuren. Andere wie SSM [45] und VAST [46] nutzen neben den  $C\alpha$ -Atomen ebenfalls Sekundärstrukturelemente. Ein Nachteil dieser beiden Verfahrenskategorien ist die häufig deutlich höhere Laufzeit als bei Berechnungen der Sequenzähnlichkeit. Auch vernachlässigt die Betrachtung von  $C\alpha$ -Atomen die Aminosäureseitenketten und damit die pharmakophoren Eigenschaften. Sie bewerten dementsprechend nur die Form und nicht die chemischen Features. Andere Methoden wie 3D-BLAST [47] berücksichtigen nur die Form, versuchen aber die Proteinoberfläche mithilfe anderer Techniken wie der sphärisch polaren Fourier Repräsentation durch mathematische Funktionen zu beschreiben. Dies führt im Vergleich zu anderen Methoden zu einer deutlichen Beschleunigung, allerdings müssen genügend Details mit der Funktion beschrieben werden können bzw. die Diskretisierung der Oberfläche darf nicht so ungenau ausfallen, dass wichtige Bereiche nicht mehr erkennbar sind. Unabhängig der Proteinrepräsentation verwenden die meisten Methoden automatisch generierte Anfragen. Eine nutzerbasierte Fokussierung auf bestimmte Bereiche oder geometrische Eigenschaften des Proteins ist deshalb nicht möglich.

Neben den obigen Nachteilen von Verfahren, die die gesamte Proteinstruktur zur Ähnlichkeitsbestimmung verwenden, gibt es noch einen weiteren entscheidenderen Nachteil: So wurde in Studien gezeigt, dass in einigen Fällen verschiedene Proteinfaltungen ähnliche Funktionen und in anderen Fällen gleiche Faltungen unterschiedliche Funktionen haben können [48]. Hier ist häufig ein kleiner Teil der Oberfläche, nämlich eine Bindestelle, für die Wechselwirkungen mit anderen Molekülen und damit auch für die Funktion des Proteins verantwortlich [49, 50]. In Analysen wurde gezeigt, dass diese Bindestellen weitaus konservierter sind als die gesamte Proteinoberfläche [51].

Bindestellen werden häufig über einen co-kristallisierten Liganden bestimmt. Dabei werden alle Aminosäureatome, die innerhalb eines Radius um die Atome des co-kristallisierten Liganden liegen, als Bindestellenatome annotiert. So können sowohl innerhalb einer Proteinkette als auch zwischen Proteinketten Bindestellen definiert werden, ohne dass eine aufwendige Berechnung stattfinden muss. Da hierfür allerdings zwingend co-kristallisierte Liganden benötigt werden, wurden alternative Methoden zur

Vorhersage von Bindetaschen entwickelt. Diese weisen ihre eigenen Problematiken auf, wie die Anforderung an die Auflösung der Strukturrepräsentation während der Vorhersage bzw. die daraus resultierende Laufzeit und insbesondere die Genauigkeit der Vorhersagen. So kann eine Proteinstruktur beispielsweise durch ein 3D-Gitter beschrieben werden. Je höher die Auflösung des Gitters ist, desto genauer wird das Protein repräsentiert und Taschen vorhergesagt, aber desto länger dauert auch die Berechnung. Ebenso ist die Wahl und Erstellung der Datensätze zum Trainieren der Methoden eine Herausforderung und stellt eine zentrale Aufgabe in der Entwicklung dar.

### 1.3 Motivation

Die Verwendung geometrischer Suchen in Proteinstruktursammlungen ermöglicht umfassende Analysen und damit eine Vielzahl an Einblicken in die chemischen Prozesse zur Unterstützung des Wirkstoffentwurfs. Sie werden zur Analyse von Proteinoberflächen, Protein-Ligand-Bindestellen und deren Interaktionen, und Protein-Protein-Wechselwirkungen verwendet. Durch die Analyse von molekularen Interaktionen können beispielsweise die Selektivität von Liganden und konservierte, relevante Reste in Bindestellen vorhergesagt werden. Unter anderem ermöglicht dies eine Verbesserung der Affinität bzw. die Optimierung von Molekülen, sodass diese selektiver an die Zielbindestelle binden. Auch können Positionen in der Proteinstruktur gefunden werden, die als Startpunkt für den Fragment-basierten Wirkstoffentwurf nützlich sind. Eine weitere bereits im vorhergehenden Unterkapitel genannte Anwendung ist die Suche nach ähnlichen Bindestellen anhand übereinstimmender Muster zur Vorhersage der Proteinfunktion oder zur Erleichterung der Suche nach neuen, bisher unbekanntem Interaktionspartnern. Ebenso werden für viele Anwendungen wie im Protein-Ligand Docking Bewertungsfunktionen zur Vorhersage der Bindungsaffinität benötigt. Meist basieren die Funktionen auf den molekularen Interaktionen und deren Geometrie. Das Wissen über bekannte Interaktionsmuster kann zur Optimierung der Funktionen verwendet werden.

Ziel des in dieser Dissertation beschriebenen Promotionsprojekts war die Entwicklung einer Methode zur geometrischen Analyse von Protein-Ligand-Bindestellen in großen Proteinstruktursammlungen. Da die Menge an verfügbaren Proteinstrukturen und Bindestellen bereits jetzt umfangreich ist und, wie oben in Abbildung 1.1 bereits gezeigt, in den nächsten Jahren weiter erheblich ansteigt, wird ein Programm zu deren Durchsuchung immer wichtiger. Der Fokus der entwickelten Methode lag auf der automatisierten Vorhersage von Bindetaschen, umfangreicher Suchmöglichkeiten, guter Skalierbarkeit und hoher Geschwindigkeit. Als Ausgangspunkt wurde die 2017 veröffentlichte

geometrische Mustersuche PELIKAN [52] verwendet. PELIKAN wurde zur Speicherung und Analyse von Protein-Ligand Bindetaschen entwickelt. Die Taschen wurden hierbei über einen festgelegten Radius definiert und in einer Datenbank hinterlegt. Der PELIKAN Algorithmus ermöglicht dadurch eine schnelle Durchsuchung einer Vielzahl von Bindetaschen mithilfe einer eigens zu diesem Zweck entwickelten Indexstruktur. Diese basiert auf Dreiecken, die aus jeweils drei Suchpunkten und deren Distanzen zueinander erstellt werden. Neben der geometrischen Suche verfügt PELIKAN über eine Reihe textueller und numerischer Filter. Trotz vieler Funktionen weist PELIKAN einige Defizite in der Formulierung von Anfragen auf:

- Die Suche in Protein-Ligand Bindetaschen unterstützt die Arbeit mehrerer Bereiche wie der Optimierung von Wirkstoffen, allerdings setzt dies co-kristallisierte Moleküle voraus. Häufig handelt es sich bei diesen Molekülen allerdings um Lösungsmittelmoleküle. Somit muss zwischen diesen Lösungsmittelmolekülen und „echten“ Liganden unterschieden werden. Mit einer Bindetaschenvorhersage können auch in Abwesenheit von Liganden mögliche Taschen identifiziert werden. Diese verwenden des Öfteren geometrische Eigenschaften wie konkave Bereiche in der Proteinoberfläche oder Vergrabenheit für die Vorhersage von Bindetaschen. Durch die Kombination dieser konkaven Bereiche und der Information vorhandener Ligandenpositionen im Komplex können Bindetaschen bzw. deren Begrenzung außerdem umfassender als mit einem Radius definiert werden.
- Neben den für den Wirkstoffentwurf hauptsächlich genutzten Protein-Ligand Bindetaschen werden Protein-Protein-Bindestellen immer populärer. Aufgrund ihrer Größe und der im Vergleich zu Protein-Ligand Bindetaschen geringeren Erforschung ist deren Analyse eine große, aber vielversprechende Herausforderung. Die Suche nach geometrischen Mustern kann hierbei unterstützen und Information über lokale Ähnlichkeiten liefern.
- PELIKAN ermöglicht die Erstellung von biomolekularen Suchpunkten. Diese können mittels Distanzen und Winkeln zur Formulierung von geometrischen Mustern kombiniert werden. Da eine Unterscheidung von Protein und Nukleinsäuren nicht stattfindet, können diese in Suchen nicht unterschieden werden, welches sowohl die Ergebnismenge als auch die Laufzeit negativ beeinflusst.
- In PELIKAN ist es zwar möglich für Proteinpunkte die Sekundärstruktur anzugeben (Helix oder  $\beta$ -Strang in einem  $\beta$ -Faltblatt), allerdings geht aus diesen Angaben nicht hervor, wo ein Sekundärstrukturelement anfängt bzw. endet, womit eine nützliche Information für die Anfrageformulierung fehlt. Wenn beispielsweise eine

Bindestelle Teil mehrerer Sekundärstrukturelemente ist oder Moleküle insbesondere mit einem Ende eines Sekundärstrukturelementes interagieren [53], können ähnliche Arrangements gefunden werden.

- Eine Substruktursuche für Liganden und Proteine ist in PELIKAN über SMARTS-Muster [54] möglich. SMARTS erlaubt es, Molekülmuster anhand eines Regelsatzes zu beschreiben. Bisher fehlt allerdings mit der Fingerprint-basierten Ähnlichkeitssuche ein Standardverfahren der Chemieinformatik, welches effiziente Filter- und Analysemöglichkeiten von Liganden ermöglicht. Mit ihr können z. B. für ein Molekül Proteinstrukturen identifiziert werden, die mit ähnlichen Molekülen cokrystallisiert wurden, sodass auf mögliche Bindungsmuster oder Zielstrukturen für einen Wirkstoff geschlossen werden kann.
- Um die Verwendung eines Programms möglichst vielen Wissenschaftler und Wissenschaftlerinnen zur Verfügung stellen zu können, wird die Entwicklung von Webtools und serverbasierten Datenbanken immer populärer. PELIKAN ist als installierbares Programm verfügbar und lässt sich mit einer portablen eingebetteten Datenbank verwenden. Dies hat den Vorteil, dass die Datenbanken zum Beispiel auf einer Webseite angeboten werden und sich Nutzer diese herunterladen oder selbst erstellen können. Hierdurch ist es möglich, vorhandene oder neue Datenbanken mit eigenen und möglicherweise nicht öffentlich zugänglichen Proteinstrukturen zu erweitern bzw. zu erstellen. Der Nachteil ist allerdings der benötigte Installationsprozess, Anforderungen an die verwendete Hardware, das dezentrale Einspielen von Updates sowie der häufige Verzicht auf Features der serverbasierten Datenbankentechnologien. So muss jeder Nutzer seine Datenbank selbstständig aktualisieren, was aufgrund der beinhalteten Präprozessierung einen hohen Rechen- und Zeitaufwand verursachen kann, und beim Auftreten von Fehlern die Installation von Programmaktualisierungen notwendig macht. Eine dezentrale, web-basierte Lösung mit interaktiver Benutzeroberfläche legt den Fokus des Nutzers komplett auf die Funktionen der Methode und alle technischen Fragen rücken in den Hintergrund. Hierfür zwingend notwendig ist eine möglichst schnelle Suche, sodass Nutzern effiziente Anfragen und schnelle Suchen ermöglicht werden.

Vor dem Hintergrund dieser Möglichkeiten zur methodischen Erweiterung wurden im Rahmen des Promotionsprojekts drei Programme entwickelt. Zur Umsetzung der Methodiken und Implementierungen wurde die NAOMI-Softwarebibliothek [55–58] für die Chemie- und Strukturbioinformatik verwendet. Der hier entwickelte Code stellt eine Erweiterung dieser Bibliothek dar. Die Ausführung ist in drei Kapitel unterteilt

und behandeln Teilaspekte der entwickelten Verfahren. Zu Beginn wird die Weiterentwicklung einer geometrischen Bindetaschenvorhersagemethode motiviert und beschrieben. Im zweiten Teil erfolgt die Beschreibung der Weiterentwicklung von PELIKAN im Kontext der geometrischen Mustersuche in biomolekularen Datenbanken. Daran anschließend wird die Problematik und der entwickelte Lösungsansatz zur Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen anhand geometrischer und chemischer Eigenschaften auf Basis des im zweiten Teil entwickelten Verfahrens erläutert. Die einzelnen Kapitel beginnen jeweils mit einer einleitenden Heranführung an das Thema, der Diskussion existierender Lösungsansätze und Methoden, sowie Problematiken des dort behandelten Teilaspekts. Darauf folgt eine Zusammenfassung des im Rahmen des Promotionsprojekts entwickelten Verfahrens, deren Evaluation und ein Ausblick. Im Kapitel 5 werden die Ergebnisse und deren Bedeutung in der molekularen Modellierung zusammengefasst.



## Kapitel 2

# Proteinbindetaschenvorhersage

Die biologische Funktion einer Mehrheit der großen und kontinuierlich steigenden Zahl bekannter Proteinsequenzen und -strukturen ist nach wie vor unbekannt [59]. Aus diesem Grund werden häufig computergestützte Verfahren eingesetzt, um die Proteinfunktion vorherzusagen. Beispielsweise können Sequenzen des Zielproteins mit bereits bekannten Proteinen verglichen werden. Eine weitere Methode basiert auf der Annahme, dass Proteine in den meisten Fällen ihre biologische Funktion ausüben, indem sie mit anderen Molekülen interagieren. Die Identifizierung der Region, in der diese Interaktionen stattfinden, spielt insbesondere bei strukturbasierten Verfahren zur frühen Identifizierung möglicher Binder im Wirkstoffentwurf eine zentrale Rolle [59]. Sobald Binderegionen erkannt wurden, können diese näher untersucht werden, z. B. kann die Protein Druggability, also die Wahrscheinlichkeit, dass ein kleines wirkstoffartiges Molekül mit dem Protein interagieren und dessen Funktion modulieren kann, bestimmt werden. Auch kann nach ähnlichen bekannten Bindetaschen gesucht werden, um so z. B. mögliche Liganden in einer ähnlichen Tasche zu identifizieren [60].

Die Vorhersagemethoden von Binderegionen lassen sich in zwei Kategorien einteilen [60]. Zum einen gibt es solche, die Bindetaschen für kleine Moleküle identifizieren. Zum anderen gibt es Methoden zur Bestimmung von Bindestellen, die größere Moleküle wie z. B. andere Proteine binden. Diese Protein-Protein-Bindestellen werden in Kapitel 4.1 behandelt, da diese Methoden meist auf unterschiedlichen Ansätzen zu denen für die Bestimmung von Bindetaschen für kleine Moleküle beruhen. Die Verfahren zur Vorhersage von Bindetaschen kleiner Moleküle lassen sich in folgende Unterkategorien einteilen: strukturbasierte, sequenzbasierte und kombinierte Methoden. Die strukturbasierten Methoden stützen sich auf die Analyse der 3D-Struktur der

Proteine; sequenzbasierte Methoden nutzen evolutionäre Informationen, um Aminosäuren zu identifizieren, die Bindetaschen ausmachen; kombinierte Verfahren verwenden struktur- und sequenzbasierte Ansätze, um von Vorteilen beider Kategorien zu profitieren. Die immer größere Vielfalt an vorhandenen Struktur- und Sequenzinformationen ermöglicht zudem die Entwicklung von Methoden, die auf maschinellem Lernen basieren [60].

In den nachfolgenden Unterkapiteln werden zunächst die verschiedenen Techniken eingehender beschrieben und einige etablierte Methoden vorgestellt, welche zur Vorhersage von Proteinbindetaschen verwendet werden. Im Anschluss folgt eine Zusammenfassung des im Rahmen dieser Arbeit weiterentwickelten Verfahrens zur Vorhersage von Proteinbindetaschen kleiner Moleküle, dessen Evaluation und ein Ausblick.

### 2.1 Verfahren zur Vorhersage von Proteinbindetaschen

Für die Vorhersage von Proteinbindetaschen gibt es verschiedene Ansätze. Im Folgenden werden einige anhand von Beispielen erläutert und diskutiert, um eine Übersicht auf mögliche Lösungswege zu geben.

#### 2.1.1 Strukturbasierte Verfahren

Strukturbasierte Methoden stützen sich auf Strukturinformationen aus Atomkoordinaten. Diese haben gegenüber den sequenzbasierten Methoden den Vorteil, dass die Proteinstruktur im Vergleich zur Aminosäuresequenz in der Evolution häufiger unverändert bleibt [61]. Ein entscheidender Nachteil ist allerdings, dass die Struktur des Zielproteins bekannt sein muss. Nach wie vor sind wesentlich mehr Sequenzen als Proteinstrukturen bekannt, da die Strukturbestimmung sehr viel Zeit und experimentelle Schritte erfordert. Durch Strukturvorhersagemethoden wie AlphaFold [20, 21] und SWISS-MODEL [62] kann dieses Defizit umgangen werden. Diese Methoden sind allerdings in verschiedenem Maße abhängig von bereits bekannten Strukturen, die eine ähnliche Faltung aufweisen. Eine weitere zu beachtende Bedingung für die Nutzung strukturbasierter Methoden ist die Anforderung an die Qualität der Proteinstruktur. Diese muss möglichst hoch sein, damit Vertiefungen auf der Proteinoberfläche möglichst genau erkannt werden können.

Diese strukturbasierten Methoden lassen sich in die drei Unterkategorien der geometriebasierten, energiebasierten und *Template*-basierten Methoden unterteilen.

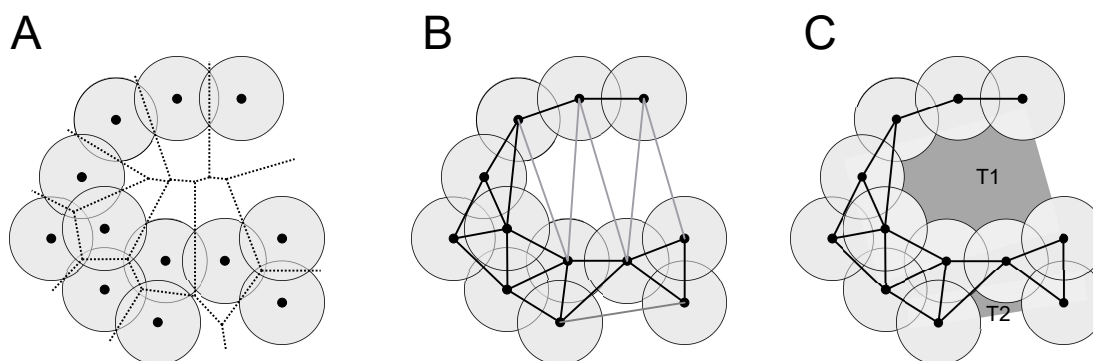
### 2.1.1.1 Geometriebasierte Methoden

Die geometriebasierten Methoden detektieren Oberflächenvertiefungen durch Analyse der Proteinoberfläche. Sie nutzen meistens 1. ein Gitter, in das das Protein eingebettet wird, 2. Simulierte Kugeln, die über die Oberfläche des Proteins gerollt werden oder 3. *Alpha Shapes* [63]. Bei Ersterem werden meist die Gitterpunkte in Protein- und Lösungsmittelpunkte unterteilt. Wenn bestimmte geometrische Bedingungen erfüllt sind, werden Mengen an Lösungsmittelgitterpunkten als Bindetaschenpunkte klassifiziert [64–66]. Diese Methoden sind abhängig von der Auflösung des Gitters bzw. der Anzahl an Gitterpunkten. Je mehr Gitterpunkte es gibt, desto genauer sind die vorhergesagten Taschen, aber desto länger dauert auch die Berechnung. Außerdem ist die Proteinorientierung im Gitter eine potenzielle Fehlerquelle [66]. Ein Verfahren, das diesen gitterbasierten Ansatz verwendet, ist LIGSITE [67]. Zuerst annotiert LIGSITE alle Gitterpunkte anhand der Proteinatome als Protein bzw. Lösungsmittel. Anschließend sucht der Algorithmus entlang der Achsen des Gitters nach Regionen, die auf beiden Seiten von Protein-Gitterpunkten umgeben sind. Beim Auffinden eines solchen Protein-Lösungsmittel-Protein (PSP) Ereignisses wird ein Zähler für den Lösungsmittel-Gitterpunkt erhöht. Damit stellt ein hoher PSP-Wert einen sehr vergrabenen Gitterpunkt und ein niedriger einen lösungsmittlexponierten Gitterpunkt dar. Regionen von Gitterpunkten mit mindestens zwei PSP-Ereignissen werden als Bindetaschenpunkte annotiert. Um kleine Bindetaschen aus den Ergebnissen zu filtern, wird eine Mindestanzahl von Gitterpunkten pro vorhergesagter Bindestelle verwendet.

Bei den kugelbasierten Methoden werden Taschen durch das Rollen von Kugeln über die Proteinoberfläche erkannt. Dies kann allerdings aufgrund der runden Form zu Problemen bei der genauen Beschreibung der Vertiefung führen, da z. B. spitze Bereiche von der Kugel nicht erreicht werden können [66]. Eine solche Methode ist der POCASA Algorithmus [66]. In diesem Ansatz wird das Protein in ein Gitter eingebettet und jedes Proteinatom als Kugel mit einem Radius gleich dem van-der-Waals-Radius beschrieben. Eine weitere, hier als Sonde bezeichnete Kugel, mit einem Radius von 2 Å, wird über die Proteinoberfläche des Gitters gerollt. Die Regionen, die von Proteinoberflächen umgeben sind bzw. zwischen Protein- und Sondenoberfläche liegen, beschreiben anschließend potenzielle Taschen. Zur Verbesserung der Genauigkeit wird für jeden dieser Kandidatenpunkte eine Konfidenz über seine Nachbarn berechnet. Wenn diese Konfidenz einen Grenzwert übersteigt, ist ein Punkt Teil der Tasche. POCASA berechnet außerdem einen Wert basierend auf der Distanz der Taschenpunkte zur Sondenoberfläche, der bewertet, wie wahrscheinlich die Tasche zur Bindung von kleinen Molekülen geeignet

ist. Die Sortierung anhand dieser Werte ermöglicht die Vorhersage biologisch relevanter Bindestellen.

*Alpha Shape* Methoden basieren auf der Delauney Triangulierung [68]. Diese verbindet Punktmengen zu Dreiecken, den sogenannten *Alpha Shapes*, wobei auf dem Umkreis des Dreiecks keine weiteren Punkte enthalten sein dürfen. Im 3D-Raum werden anstelle der Dreiecke und des Umkreises Tetraeder und Umkugeln verwendet. Leere *Alpha Shapes* bzw. solche, die mindestens eine Kante besitzen, die zum Teil oder vollständig außerhalb der Radien der Punkte des *Alpha Shapes* liegen, stellen die Bindestaschen dar. Eine beispielhafte Darstellung hierfür ist in Abb. 2.1 im Zweidimensionalen zu sehen.



**Abbildung 2.1:** Veranschaulichung der *Alpha Shape* Theorie. A: Eine zweidimensionale Darstellung von Bindestaschenatomen mit identischen Radien. Die gestrichelten Linien zeigen das Voronoi-Diagramm der Atome. B: Die konvexe Hülle der Atomzentren mit Delaunay-Triangulation. Dreiecke sind definiert durch Linien. Graue Linien stellen leere *Alpha Shapes* dar. C: Finale Definition der zwei berechneten Taschen *T1* und *T2*.

Ein Tool, das dieses Verfahren verwendet, ist fpocket [69]. Es berechnet Umkugeln (auch *Alpha Spheres* [70] genannt) für je vier Atome und bestimmt auf Basis all dieser Kugeln ein 3D Voronoi Diagramm. Dies hat den Vorteil, dass die Krümmung der lokalen Proteinumgebung durch die *Alpha Shapes* beschrieben wird. Würden die vier Atome am Scheitelpunkt eines Tetraeders liegen, würde dies zu einem Radius nahe dem van-der-Waals-Radius führen. Umgekehrt wäre der Radius unendlich, wenn die Atome auf einer Ebene liegen würden. Daraus lässt sich ableiten, dass kleine Kugeln innerhalb des Proteins liegen müssen, wohingegen große Kugeln auf Hohlräume schließen lassen. Fpocket clustert deshalb *Alpha Spheres* nach ihren Radien. Zuerst werden die *Alpha Spheres* dabei nach Abstandskriterien in Cluster gruppiert. Anschließend erfolgt eine Entfernung aller Cluster mit nur einer *Alpha Sphere* und ein Massenschwerpunkt wird jeweils für die verbleibenden Cluster berechnet. Cluster mit nahe beieinander liegenden Massenschwerpunkten werden zusammengefasst, was insbesondere zur Kombination kleiner Cluster an der Oberfläche führt. Der letzte Schritt des Clusterings basiert auf

dem *Multiple-Linkage-Clustering*-Ansatz, wobei zwei Cluster kombiniert werden, wenn *Alpha Spheres* des einen Clusters nahe an solchen eines anderen Clusters liegen. Eine Rangfolge der vorhergesagten Taschen wird abschließend durch eine Bewertungsfunktion bestimmt, welche auf den Taschendeskriptoren wie der Elektronegativität von Atomen beruht.

### 2.1.1.2 Energiebasierte Methoden

Energiebasierte Methoden identifizieren energetisch günstige Regionen auf der Proteinoberfläche. Dies wird in der Regel durch die Berechnung der Interaktionsenergie zwischen Proteinatomen und einer chemischen Sonde durchgeführt. Die Methoden unterscheiden sich in der chemischen Sonde und dem verwendeten Kraftfeld, das für die Berechnung der Interaktionsenergie benötigt wird. Ein Nachteil dieser Methoden ist die im Vergleich zu anderen Ansätzen höhere *false positive rate* [71].

Ein Verfahren, das diesen Ansatz nutzt, ist AutoSite [72], welches Affinitätskarten basierend auf dem AutoGrid4-Kraftfeld [73] und den AutoDock-Atomtypen [72, 74] für Kohlenstoff (hydrophob), Sauerstoff (Wasserstoffbrückenakzeptor) und Wasserstoff (Wasserstoffbrückendonor) berechnet. Dabei wird für jeden der drei Atomtypen ein äquidistantes 3D-Gitter als Affinitätskarte erstellt. Auf dieser beschreibt jeder Gitterpunkt einen Affinitätswert, der die Summe der paarweisen Wechselwirkungsenergie zwischen diesem Gitterpunkt und allen umliegenden Proteinatomen beschreibt. Anschließend wird ein wissensbasierter Affinitätsschwellenwert verwendet, um Punkte mit niedriger Affinität herauszufiltern. Die verbleibenden Punkte der drei Karten werden dann zu einer kombinierten Karte zusammengeführt. Die Punkte der kombinierten Karte werden mithilfe einer modifizierten Version des DBSCAN-Algorithmus [75] nach einem lokalen Dichtewert geclustert. Jedes der Cluster beschreibt anschließend eine mögliche Bindestelle. Das Ranking der Cluster erfolgt über eine Bewertungsfunktion basierend auf der Anzahl und der Vergrabenheit der Taschengitterpunkte, sowie dem Gyrationradius.

### 2.1.1.3 *Template*-basierte Methoden

*Template*-basierte Methoden versuchen, die Lage von Bindungsstellen aus bekannten Strukturen als Vorlage bzw. Schablone (englisch: *Template*) abzuleiten. Dies baut auf der Annahme auf, dass ähnliche Strukturen auch eine ähnliche Funktion ausüben [76]. Der zugrundeliegende Mechanismus dieser Methoden ist die Berechnung eines Alignment zwischen dem Zielprotein und einem Set an *Templates*. Letztere werden häufig

in Datenbanken gespeichert (z. B. ASSIST [77] oder COFACTOR [78]) oder auch dynamisch generiert (z. B. in LBias und LTsearch [79]). Nach der Suche wird meist eine Bewertung durchgeführt. In dieser werden häufig vorhandene Liganden in den *Templates* überlagert und geclustert, sodass mögliche Taschen aus diesen Clustern abgeleitet werden können. Ein Vorteil dieser Methoden ist die im Vergleich zu Energie- und Geometriebasierten Verfahren größere Menge an korrekten Vorhersagen von Taschen, die kleine Moleküle binden [60]. Einschränkungen liegen allerdings in der zugrundeliegenden oben genannten Annahme. So können beispielsweise gute Alignments zwischen *Templates* und der Zielstruktur existieren, die weit entfernt verwandt sind, aber unterschiedliche Funktionen aufweisen. Einen solchen Fall gibt es z. B. beim TIM-Fass (englisch: TIM barrel), welches in zahlreichen Proteinfamilien mit einer Vielzahl unterschiedlicher Funktionen zu finden ist [80]. Umgekehrt können Proteine, die keine allgemeine strukturelle Ähnlichkeit aufweisen, eine Ähnlichkeit der Bindungsstellen aufweisen [81–83]. Auch kann es vorkommen, dass insbesondere bei nicht gut charakterisierten Typen von Bindetaschen wie kryptischen und allosterischen Taschen keine geeigneten *Templates* mit Liganden bekannt sind.

LBias [79] ist eine *Template*-basierte Methode, die Ligand-bindende Reste des Zielproteins bestimmen soll. Im ersten Schritt werden verwandte Strukturen des Zielproteins mit ihren co-kristallisierten Liganden gesammelt. Anschließend werden die Strukturen der verwandten „Nachbarn“ und des Zielproteins überlagert, sodass ein Score berechnet werden kann. Dieser beschreibt die Wahrscheinlichkeit, dass ein Rest des Proteins an der Ligandenbindung beteiligt ist. Der Score gibt hierbei an, wie gut die ausgebildete Interaktion zwischen der verwandten Struktur und dessen Liganden auch mit dem Zielprotein gebildet werden könnte.

### 2.1.2 Sequenzbasierte Verfahren

Sequenzbasierte Methoden sind in der Regel weniger rechenintensiv als strukturbasierte Methoden, haben aber häufig Schwierigkeiten beim Identifizieren von Bindestellen von weit entfernt verwandten Proteinen. Die zugrundeliegende Annahme der Verfahren ist die Konservierung von Bindetaschen aufgrund ihrer funktionellen Bedeutung für das Protein [84]. Da auch Aminosäurereste konserviert sein können, die nicht an einer Ligandenbindung beteiligt sind [60], kann diese Annahme allerdings auch zu falsch-positiven Ergebnissen führen. In den letzten Jahren entwickelte Methoden nutzen häufig maschinelle Lernverfahren zur Vorhersage möglicher Bindestellen [60].

ROBBY [85] ist eine Methode, die ligandenbindende Reste auf Basis einer trainierten

*Support Vector Machine* [86] (SVM) identifiziert. Als Trainingsdaten werden evolutionäre Informationen verwendet, die durch ein multiples Sequenzalignment mithilfe von PSI-BLAST [87] aus der LigASite [88] Datenbank erfasst wurden.

### 2.1.3 Kombinierte Verfahren

In den kombinierten Verfahren werden struktur- und sequenzbasierte Ansätze zusammen verwendet, um von den Vorteilen beider zu profitieren. Bei der Kombination mehrerer Methoden liegt das Augenmerk meist auf den Bewertungsfunktionen zur Einschätzung und Verknüpfung ihrer Ergebnisse. Meist werden Webservices verwendet, um dem Benutzer die Verwendung dieser Kombination an multiplen Methoden zu erleichtern. Durch das Vermischen komplementärer Ansätze lassen sich teils sehr hohe Genauigkeiten erzielen [89], aus dem gleichen Grund können die Modelle allerdings auch gute Vorhersagen verschlechtern [90].

COACH [91] und die Erweiterung COACH-D [92] akzeptieren sowohl Sequenzen als auch Strukturen zur Taschenvorhersage. Der Unterschied zwischen beiden Verfahren liegt hauptsächlich in der Entfernung von sterischen Kollisionen zwischen Ligand und Bindetasche in COACH-D. Wird eine Sequenz als Eingabe verwendet, wird die Struktur in beiden Ansätzen mithilfe der I-TASSER-Suite vorhergesagt [93]. Sowohl COACH als auch COACH-D verwenden die Methoden TM-SITE [91], S-SITE [91], ConCavity [94], FINDSITE [95] und COFACTOR [96], um Vorhersagen zu treffen. Die besten Vorhersagen jeder Methode werden dann durch eine SVM kombiniert.

### 2.1.4 Maschinelles Lernen

Maschinelles Lernen findet zunehmend Verwendung im strukturbasierten Modeling. Dies liegt an ihrer Fähigkeit, Zusammenhänge in Datenmengen zu erkennen, die anhand einfacher Modelle nur schwer greifbar sind. Meist sind komplexe Beziehungen von Eigenschaften oder eine unzureichende Erforschung für zuverlässige Vorhersagen, Gründe für die Verwendung dieser Methoden. Ein Beispiel für eine große Vielfalt an Informationen sind die hier behandelten Bindetaschen. Diese können mit unterschiedlichsten Detailgraden betrachtet werden, wie z.B. auf Grundlage der Sequenz, Sekundärstruktur oder atomarer Beziehungen. Im Folgenden wird das maschinelle Lernen im Zusammenhang mit Klassifikationsproblemen erläutert. Grob kann die Funktionsweise der Verfahren in die folgenden vier Schritte zusammengefasst werden: 1. das Aufbereiten der Daten, 2. die Auswahl der Informationen bzw. Features, 3. die Modellerstellung und

4. die Evaluierung [97]. Die Modellerstellung kann wiederum grob in die beiden Kategorien des überwachten und unüberwachten Lernens unterteilt werden. Erstere Modelle werden mit bekannten Daten trainiert, bei denen bereits für die Eingabe eine korrekte Ausgabe bzw. ein Ergebnis bekannt ist. Hierdurch soll ein Modell trainiert werden, das anhand der Eingabe das Ergebnis vorhersagen kann. Bei den unüberwachten Lernverfahren werden hingegen Daten übergeben, die ohne Intervention in ein statistisches Modell durch eine Unterteilung (Clustering) umgewandelt werden. Ein bekanntes Ergebnis ist hier nicht notwendig. Der Algorithmus wählt also selbst Zusammenhänge der Features aus, um die Daten unterteilen zu können. Insbesondere beim überwachten Lernen müssen geeignete Features ausgewählt werden, mit denen das Modell trainiert wird. Außerdem muss es möglichst gleich viele positive wie negative Fälle geben, um ein Ungleichgewicht zu vermeiden [98]. Bei den unüberwachten Lernverfahren hingegen ist die Evaluation oft schwierig [99]. Ein dabei typisches Problem ist die Auswahl einer Anzahl von Clustern [100].

Deep-Site [101] ist eine als Webserver verfügbare Methode, die ein *Convolutional Neural Network* [102] verwendet. Proteine werden von der Methode als 3D-Bilder beschrieben und in  $1 \times 1 \times 1 \text{ \AA}^3$  Voxel unterteilt. Jedes Voxel ist durch einige pharmakophore Eigenschaften auf atomarer Basis mithilfe von AutoDock 4-Atomtypen [74] beschrieben. Trainiert wurde das Modell mithilfe von 6860 Strukturen aus der sc-PDB-Version 2013 [35, 103].

## 2.2 Proteinbindetaschenvorhersage mit DoGSite

Die Vorhersage von Proteinbindetaschen wird in der NAOMI-Plattform vom DoGSite-Algorithmus übernommen. Die erste Version [104–106] des DoGSite-Algorithmus, im Folgenden *dogsite.v1* genannt, ist im Dockingprogramm FlexX [107] integriert. Der DoGSite-Algorithmus nutzt einen strukturbasierten Ansatz. Eine Besonderheit des DoGSite-Algorithmus ist die Berechnung von Taschen-Kernen, sogenannten *Subpockets*, die meist kleine Taschen beschreiben und zu größeren Taschen kombiniert werden. Die Unterteilung in *Subpockets* ermöglicht eine genauere strukturelle Beschreibung der Topologie der gesamten Bindetasche. Außerdem kann das Problem der Vorhersage großer Taschen und der Überschätzung des tatsächlichen Ligandenbindungsvolumens durch das *Subpocket*-Konzept gelöst werden [104].

DoGSite wurde zwischen 2010 und 2012 entwickelt und evaluiert. Auf Grundlage der verwendeten Datensätze und Evaluationskriterien konnte gezeigt werden, dass DoGSite in der Vorhersagequalität vergleichbar zu anderen Methoden ist [104, 105]. Neuere



Analysen zeigen allerdings Optimierungsbedarf im Hinblick auf die "Robustheit" und Zuverlässigkeit der Vorhersagen auf [D1].

Um den DoGSite-Algorithmus in der NAOMI-Plattform nutzen zu können, wurde der Algorithmus 2013 in der Plattform reimplementiert (im Folgenden *dogsites.v2* genannt). Die Integration in die NAOMI-Plattform bietet dabei unter anderem den Vorteil, dass Programme, die mit der NAOMI-Plattform entwickelt werden, diesen leicht integrieren können. Die Implementierung des DoGSite-Algorithmus wurde innerhalb NAOMI bis zum Beginn dieses Promotionsprojekts allerdings einzig in der virtuellen Screening-Methode TrixX [108] integriert.

Im Laufe dieses Promotionsprojekts sollte der DoGSite-Algorithmus für die Bereitstellung von vorhergesagten Taschen in Abwesenheit von Liganden in der Weiterentwicklung des Programms PELIKAN verwendet werden. Zu diesem Zweck sollte der DoGSite-Algorithmus verbessert werden. Insbesondere sollte die Optimierung der Vorhersagegenauigkeit und -geschwindigkeit, sowie der Einbezug der Position von Liganden aus Protein-Ligand-Komplexen in die Taschenvorhersage im Fokus stehen. Letzteres ermöglicht es, die Vorhersage von Taschen zu unterstützen, wenn bereits bekannt ist, an welcher Stelle Liganden an das Protein binden und die Gesamtdimensionen der Taschen genauer zu bestimmen. Bei der Analyse der Taschen-Vorhersagen und der Code-Basis von *dogsites.v2* fielen Probleme auf, die neben den vorgesehenen Arbeiten eine umfangreiche Überarbeitung und teilweise Reimplementierung zwingend notwendig machten. So wurden, teils bereits in NAOMI und der Sprache C++ vorhandene, Datenstrukturen für schnellere Zugriffe nicht benutzt und je nach Parameterkombinationen des DoGSite-Algorithmus kam es zu Speicherzugriffsfehlern. Auch wiesen einige essenzielle Funktionalitäten und Algorithmusschritte Fehler auf. So konnten z. B. CCP4-Dateien, die die Ausmaße der Tasche beschreiben, nicht korrekt für vorhergesagte Taschen geschrieben werden und in vielen Schritten des Algorithmus, wie dem Vergrößern von Clustern um benachbarte Lösungsmittelpunkte oder dem Kombinieren von *Subpockets*, wurden nicht alle erforderlichen Gitterpunkte betrachtet. Insbesondere Fehler wie letztere führten zu einer unvollständigen Vorhersage und zu signifikanten Varianzen in den Taschen-Deskriptoren wie der Taschentiefe oder dem -volumen, wenn Taschen für unterschiedliche Transformationen des gleichen Proteins berechnet wurden. Dies ist ein Indikator für eine unzureichende Qualität der Vorhersage, da der Algorithmus ansonsten zu sensitiv und damit abhängig von der Lage der Proteinstruktur im 3D-Raum ist. Die Optimierung der Platzierung des Gitters war deshalb ein weiterer zentraler Bestandteil der Überarbeitung. Da seit der Entwicklung von *dogsites.v2* zehn Jahre vergangen

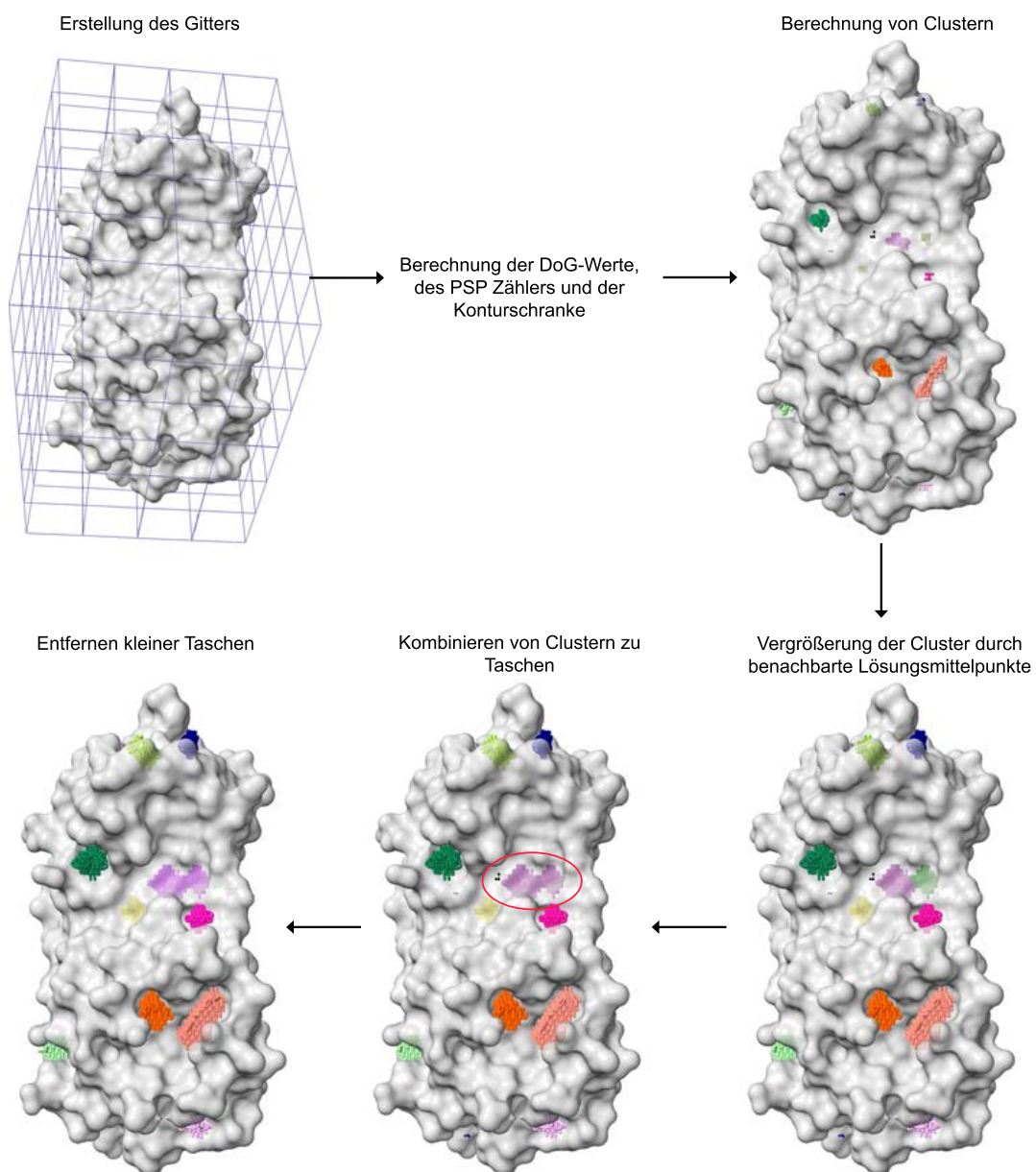
sind, lag ein weiterer Fokus auf der Optimierung der Methodenparameter auf Basis aktueller Datensätze.

Im Folgenden wird das methodische Konzept der neuen Version `dogsites.v3` bzw. DoG-Site3 zusammengefasst, einige Evaluationen dargestellt und ein Ausblick gegeben. Ausführliche Erläuterungen zu Optimierungen und Anpassungen, sowie eine detaillierte Beschreibung des Algorithmus und der Evaluationen sind in [D1] beschrieben.

### 2.2.1 Methodische Zusammenfassung

Der grundlegende Ablauf des DoG-Site-Algorithmus ist in Abbildung 2.2 dargestellt und kann in sechs Schritte unterteilt werden. Zunächst wird ein Gitter für alle makromolekularen Ketten der Eingabestruktur erstellt. Als Ketten zählen im Kontrast zu `dogsites.v1` neben Proteinen auch Nukleinsäuren. Um die erforderliche Gittergröße zu minimieren und eine möglichst orientierungsunabhängige Berechnung zu ermöglichen, wird eine Hauptkomponentenanalyse durchgeführt. Die drei Hauptachsen des Makromoleküls werden anschließend so ausgerichtet, dass sie jeweils mit der x-, y- und z-Achse übereinstimmen. Die Orientierung des Makromoleküls ist damit bis auf mögliche Spiegelungen an den Achsen identisch, sodass die Grenzen des Gitters über die maximalen und minimalen Koordinaten aller Atome berechnet werden können. Die Anzahl der Gitterpunkte ergibt sich aus diesen Koordinaten und einem benutzerdefinierten Gitterpunktabstand  $\delta$ . Falls das Makromolekül zu groß ist und das Gitter mehr als 9 Millionen Gitterpunkte enthält, wird  $\delta$  erhöht und damit ein Gitter mit geringerem Detailgrad erstellt. Im Anschluss wird jeder Gitterpunkt genau dann als Makromolekül-Gitterpunkt beschrieben, wenn dieser einen vordefinierten Abstand zu einem Makromolekül-Atom nicht übersteigt. Dieser Abstand entspricht der Summe des van-der-Waals-Radius des jeweiligen Atoms und einem definierten Abstand ( $rTolerance$ ), der dem Gitterabstand  $\delta$  gleichgesetzt ist.

Im zweiten Schritt werden zwei Eigenschaften für alle Gitterpunkte berechnet, um mit deren Hilfe die Bindetaschen zu bestimmen. Bei diesen handelt sich um die *Difference of Gaussians*-Werte (englisch für Differenz der Gauß-Kurven, DoG) und die PSP Ereignisse. Die DoG-Werte beschreiben hierbei Regionen an der Makromoleküloberfläche, die Platz für ein sphärisches Objekt bzw. ein Atom bieten. Sie werden basierend auf dem Gitter berechnet und verwendet, um eine Konturschranke zu bestimmen. Falls das Einbeziehen der Position eines Liganden aktiviert ist, werden die DoG-Werte von Lösungsmittel-Gitterpunkten an diesen Positionen künstlich reduziert. Dabei wird zuvor der Ligand in rigide Einheiten unterteilt. Eine rigide Einheit wird nur dann zur



**Abbildung 2.2:** Berechnungsschritte des DoGSite-Algorithmus anhand der Proteinstruktur mit dem PDB Code 6Y2F. Zunächst wird das Protein in einem uniformen Gitter platziert. Die Gitterpunkte werden mit dem DoG-Wert und der Anzahl von PSP Ereignissen annotiert. Es wird zudem eine Konturschranke für die DoG-Werte berechnet. Anhand dieser Werte werden anschließend Cluster von Lösungsmittelpunkten gebildet. Anschließend werden die Cluster vergrößert. Cluster, die eine ausreichend große Fläche von benachbarten Punkten zu einem anderen Cluster aufweisen, werden zu einem Cluster kombiniert. Im Beispiel ist dies in der rot umrandeten Region zu sehen. Im letzten Algorithmusschritt werden kleine Taschen anhand ihres Volumens entfernt und die übrig bleibenden als Ergebnismenge ausgegeben. In der Abbildung wird beispielsweise im Bereich der vorherigen Kombination zweier Taschen eine in Schwarz sichtbare Tasche gelöscht. Alle Moleküldarstellungen wurden mithilfe von UCSF ChimeraX [109] erstellt.

Manipulation der DoG-Werte verwendet, wenn die Lösungsmittelzugänglichkeit einen zuvor festgelegten Wert nicht übersteigt. Dieser *Ligand-Bias* genannte Schritt, sorgt dafür, dass eine Vorhersage von Taschen an diesen Ligand-Positionen wahrscheinlicher wird. Die PSP Ereignisse werden analog zu LIGSITE (siehe 2.1.1.1) mithilfe eines Scanline-Verfahrens über alle Achsen des Gitters gezählt.

Die berechneten Werte werden anschließend verwendet, um Cluster von Lösungsmittel-Gitterpunkten zu finden. Dafür wird überprüft, ob der DoG-Wert eines Gitterpunktes unterhalb der berechneten Konturschranke liegt. Ist dies der Fall, werden alle benachbarten Gitterpunkte ebenfalls überprüft. Wenn mindestens fünf Nachbarn ebenfalls einen DoG-Wert unterhalb der Konturschranke aufweisen, wird eine Tiefensuche gestartet. Diese überprüft sukzessive alle Nachbarn und fügt sie zum Cluster hinzu, bis keine weiteren Lösungsmittelgitterpunkte unterhalb der Schranke zu finden sind. Nachdem alle Gitterpunkte überprüft wurden, werden die Gitterpunkte mit einem Cluster-Index annotiert.

Um die Cluster um benachbarte, aber zuvor nicht gefundene Lösungsmittelpunkte zu vergrößern, wird eine zweite Tiefensuche auf jedem Gitterpunkt gestartet, dem ein Cluster bereits zugeordnet wurde. Die Suche durchläuft eine maximale Anzahl an Iterationen und überprüft mehrere Kriterien, wie die Distanz zwischen dem neuen Gitterpunkt und dem Cluster oder die Anzahl der PSP Ereignisse.

Im fünften Schritt wird über alle Cluster iteriert und erneut die Nachbar-Gitterpunkte des jeweiligen Clusters überprüft. Wenn zwei Cluster eine hinreichend große Anzahl an Gitterpunkten in der Nähe des jeweils anderen Clusters aufweisen, werden die beiden Cluster zu einem Cluster kombiniert. Die benötigte Anzahl wird dabei über die Formel  $((1.4 - rTolerance)/delta)^2$  berechnet und beschreibt eine Fläche an Lösungsmittel im Verhältnis zum definierten Gitter. Eventuelle Lösungsmittel-Gitterpunkte zwischen den beiden nun verbundenen Clustern werden außerdem dem neuen Cluster hinzugefügt. Diese neuen Punkte werden dann ebenfalls ihrerseits auf eine mögliche Erweiterung des Übergangs der beiden Cluster zueinander überprüft. Dies ist besonders dann hilfreich, wenn die beiden Cluster Bereiche beinhalten, deren räumliche Distanz gering ist, die aber nicht unmittelbar benachbart sind.

In einem letzten Schritt werden Volumina der Cluster über die Anzahl der Gitterpunkte und den Gitterpunktabstand berechnet. Alle Cluster mit einem Volumen kleiner als  $20 \text{ \AA}^3$  werden entfernt. Abschließend werden die Atome des Makromoleküls identifiziert, die in der unmittelbaren Umgebung des Clustergitters liegen, indem die Distanz zwischen den Gitterpunkten und den Atomen überprüft wird. Außerdem werden

charakteristische Eigenschaften der Bindetaschen berechnet, die die Geometrie und chemischen Eigenschaften der Taschen beschreiben.

### 2.2.2 Evaluation

DoGSite3 wurde in umfangreichen Tests auf Vorhersagegenauigkeit, Laufzeit und Robustheit der berechneten Bindetaschendescriptoren bzw. Stabilität der Deskriptoren gegenüber Änderungen der Rotation der Makromolekülstruktur untersucht. Letztere wurde für die Auswahl einer optimierten Methodenparameterkombination berechnet. Hierfür wurden Proteinstrukturen der ersten beiden Datensätze der ProSPECCTS Benchmarks [110] verwendet. Die Taschenbestimmung wurde ohne Berechnung einer einheitlichen Proteinorientierung durchgeführt. In der Analyse konnte mit den ausgewählten optimierten Parametern insbesondere eine deutliche Erhöhung der Stabilität des Bindetaschenvolumens beobachtet werden. Weitere Analysen zur Stabilität der Deskriptoren werden in Kapitel B.1 beschrieben und sind nicht in der Publikation [D1] enthalten. Diese beinhalten unter anderem einen Vergleich der Deskriptorstabilität von dogsite.v2 und DoGSite3 unter Verwendung gleicher Methodenparameter. Hierdurch zeigt sich der Einfluss der methodischen Überarbeitung von dogsite.v2.

Wie die in [D1] aufgeführten Ergebnisse darlegen, wurde die Vorhersagegenauigkeit durch die beschriebenen Optimierungen ebenfalls verbessert. So konnte im Vergleich zu 40 anderen etablierten geometrischen, energiebasierten, *Template*-basierten und kombinierten Methoden zur Bindetaschenvorhersage demonstriert werden, dass die neue Version DoGSite3 die höchste Vorhersagegenauigkeit erzielt. Diese Analyse wurde anhand von drei Datensätzen mit Bindetaschen aus der PDBbind [111], kryptischen Bindetaschen [112] und allosterischen Bindetaschen [112] durchgeführt. Auch konnte eine signifikante Verbesserung von dogsite.v2 zu DoGSite3 beobachtet werden. Hier erreicht dogsite.v2 auf den drei verwendeten Datensätzen den 18., 25. und 25. Rang, während DoGSite3 jeweils Rang 1 einnimmt.

Auch in den Laufzeitanalysen konnte eine deutliche Verbesserung von dogsite.v2 zu DoGSite3 beobachtet werden. Die Analysen wurden auf dem ersten Datensatz des Bindetaschenvergleichs-Benchmarkset ProSPECCTS [110] durchgeführt. Dieser Datensatz besteht aus mehreren Strukturen für 12 sequentiell diverse Proteine (12 Gruppen, 326 Strukturen). Beide DoGSite-Versionen wurden auf einem PC mit einem Intel i5-8500 (3,0 GHz) Prozessor, 16 GB RAM und einer SSD (256 GB, Modell NVMe) mit xfs Dateisystem getestet. In fünf voneinander unabhängigen Berechnungen benötigte dogsite.v2

im Schnitt 4 158 s und DoGSite3 298 s für alle Strukturen. Die Laufzeit von dogsite.v2 zu DoGSite3 reduziert sich damit um einen Faktor von etwa 14.

### 2.2.3 Ausblick

Die Druggability-Vorhersage stellt eine zentrale Funktion der dogsite.v1 und dogsite.v2 Anwendungen dar. Durch die Reimplementierung des Algorithmus und insbesondere durch die Reoptimierung der Parameter muss das Druggability-Modell neu evaluiert und angepasst werden. Dies ist Teil eines von diesem Promotionsprojekt unabhängigen Projekts.

Die wahrscheinlich größte Herausforderung bei der Identifikation von Bindetaschen ist die Einbeziehung der Proteinflexibilität. Kristallographisch gelöste Proteinstrukturen stellen nur eine gemittelte Momentaufnahme und somit einen rigiden Zustand dar. Da Proteine allerdings im Lösungsmittel dauerhaft in Bewegung sind, verändert sich ebenfalls die Konformation. Die Berechnung von Taschendescriptoren wie dem Volumen beinhaltet dementsprechend eine gewisse Ungenauigkeit. Einen möglichen Ansatz zur Berücksichtigung der Flexibilität wäre die Überlagerung von Proteinstruktur-Ensembles. Anhand eines Ensembles könnte ein Proteingitter berechnet und somit Bindetaschen für mehrere Proteinkonformationen berechnet werden. Unter anderem stellt die Kryoelektronenmikroskopie und die Solution NMR [113] Methode eine vielversprechende Technik zur Bestimmung solcher Ensembles dar. Mit ihnen können Proteine ohne Bildung der starren Kristalle in unterschiedlichen Konformationen aufgenommen werden [113–115].

## Kapitel 3

# Geometrische Mustersuche

Wie bereits in Kapitel 1.2 dargestellt, wächst die Anzahl öffentlich zugänglicher Proteinstrukturen stetig. Strukturen werden in Datenbanken wie der PDB [22] zur Verfügung gestellt. Diese Datenvielfalt ermöglicht eine Vielzahl von Analysen, die die Arzneimittelentwicklung unterstützen und erleichtern können. Zur Nutzung der Strukturdaten müssen allerdings Datenverwaltungssysteme bzw. -suchsysteme entwickelt werden, um solch große Datenmengen effizient analysieren zu können. Insbesondere Informationen über Eigenschaften von Ligandbindetaschen, Ähnlichkeiten von Bindestellen, und Interaktionsmuster sind von Interesse, aber teilweise schwierig zu bestimmen. Sie können verwendet werden, um beispielsweise vorherzusagen, ob ein Ligand an eine bestimmte Tasche binden kann, um ein initiales Fragment zum Fragment-basierten Wirkstoffentwurf zu finden, um die Proteinfunktion vorherzusagen, Selektivitätsunterschiede zu untersuchen, oder um Taschen zu identifizieren, die einen bekannten Wirkstoff binden (*Drug Repurposing*). Insbesondere die Verwendung bekannter Wirkstoffe für neue Indikationen hat zahlreiche Vorteile gegenüber dem Entwurf neuer Wirkstoffe [116]. So ist das Risiko eines Misserfolgs geringer, da die Sicherheit des Wirkstoffes bereits in Studien belegt worden ist. Außerdem kann die Entwicklungszeit reduziert werden, da bereits präklinische Tests durchgeführt wurden und dadurch ebenfalls die benötigten Investitionen in die Entwicklung geringer sind [117].

Die Analyse der 3D-Strukturdaten lässt sich auf das Suchen von geometrischen Mustern reduzieren. Mehrere Methoden wurden deshalb in den letzten zwei Jahrzehnten zu diesem Zweck entwickelt. Unterschiede dieser Methoden liegen allerdings in Aspekten wie der Datenquelle und -verarbeitung, dem Suchalgorithmus oder auf welche Art bzw. mit welchem Detailgrad Muster geometrisch und chemisch beschrieben werden können.

Im Folgenden werden zunächst Herausforderungen der geometrischen Suche in Bindetaschen vorgestellt. Im Anschluss werden einige Verfahren erläutert und bezüglich der Herausforderungen diskutiert. Abschließend wird das im Rahmen des Promotionsprojekts entwickelte Verfahren beschrieben und auf dessen Anwendungen eingegangen.

## 3.1 Herausforderungen der Suche nach ähnlichen Bindetaschen

Eine zentrale Herausforderung der geometrischen Suche in bekannten Bindestellen ist die Laufzeit der Suche, da riesige Datenmengen zu bewältigen sind, welche darüber hinaus stetig zunehmen. Aus diesem Grund muss zuerst entschieden werden, ob nur Ligandenbasierte oder auch vorhergesagte Bindetaschen durchsucht werden sollen. Dies kann bereits einen Faktor von etwa zwei in der Anzahl an Taschen ausmachen [D2]. Auch müssen die 3D-Daten mit ihrem chemischen Kontext gespeichert werden. Hierbei muss dem Benutzer eine Vielfalt an Informationen zur Verfügung gestellt werden, um möglichst genaue Anfragen zu erlauben. Die Menge an Informationen darf allerdings nicht so groß werden, dass die Bedienbarkeit der Anwendung mangels Übersichtlichkeit darunter leidet. Darüber hinaus spielen die Toleranzen der geometrischen Suchkriterien eine entscheidende Rolle. Die Anfragen sollten so flexibel gestaltet werden können, dass nicht nur exakte Übereinstimmungen der Muster gefunden werden können. Da Interaktionen und Distanzen zwischen Atomen und chemischen Gruppen nicht immer gleich sind und somit gewisse Spielräume existieren, sollten diese im geometrischen Muster durch Toleranzen berücksichtigt werden können.

Eine weitere Herausforderung stellt das Präsentieren der Ergebnisse dar. Die teils komplexen und häufig umfangreichen Resultate müssen sortiert und dem Nutzer auf anschauliche Weise präsentiert werden. Auch ist die Möglichkeit der Präzisierung einer bereits durchgeführten Suche häufig von Interesse.

Eine der größten Herausforderungen sind die Validierung, Evaluierung und der Vergleich von Suchverfahren. Die Validierung wird meist methodisch argumentiert oder häufig an nur wenigen Anwendungsbeispielen vorgenommen. Die Evaluierung gestaltet sich noch anspruchsvoller, da die Suchanfragen beliebig restriktiv bzw. tolerant gestellt werden können. So können sich beispielsweise mögliche Toleranzbereiche für Abstände zwischen Atomen oder Atomgruppen unterscheiden und unterschiedlich viele chemische Details definiert werden. Auch sind Methoden nur anhand zuvor festgelegter Kriterien wie der genutzten Datenquelle oder dem Suchalgorithmus vergleichbar. Ein Vergleich



der Ergebnismengen ist häufig ebenfalls nicht möglich, da die Methoden sich teilweise signifikant in der Abdeckung chemischer Eigenschaften, der Wahl der zur Verfügung stehenden Suchpunkte und der Definition der Bindetaschen unterscheiden.

Zusammengefasst gibt es unter anderem die folgenden Herausforderungen bei der Suche in Proteinstrukturdaten:

- Auswahl des Datensatzes bzw. der verfügbaren Strukturen
- Wahl der durchsuchbaren Strukturdaten (Proteine oder Proteinbindetaschen – Ligand-basiert oder vorhergesagt)
- Suchalgorithmus
- Übersichtliche Ergebnispräsentation
- Validierung und Evaluierung

## 3.2 Verfahren zur Suche in Proteindatensammlungen

Die Suche in Proteinstrukturdaten kann konzeptionell in zwei Kategorien eingeteilt werden: 1. Verfahren, die in Proteinen nach strukturellen und meist vorgegebenen Mustern suchen und 2. solche, die Proteinbindetaschen durchsuchen, und somit geometrische und chemische Variabilität der Gestaltung von Suchanfragen ermöglichen. Im Folgenden werden einige Ansätze der beiden Kategorien vorgestellt. Ein detaillierter Vergleich bekannter Methoden ist in [D2] zu finden.

### 3.2.1 Suchverfahren in Proteinstrukturen

Die wahrscheinlich erste Methode zur Suche von Mustern in Proteinstrukturen ist **3DinSight** [118] aus dem Jahr 1998. Dabei handelte es sich um einen Webservice, der mittlerweile allerdings nicht mehr fortgeführt wird. In einer relationalen Datenbank wurden alle Strukturen der PDB gespeichert. Mit 3DinSight konnten Sequenzmuster sowohl in der gesamten Proteinstruktur als auch in Sekundärstrukturelementen gesucht werden. Mit Schlüsselwörtern aus den Strukturdateien wie dem Organismus und Attributen wie der Strukturauflösung ließen sich die Suchen spezifizieren. Anfragen konnten neben einem Webinterface auch über eine *structural query language* (SQL) Schnittstelle gestellt werden. Letztere hatte den Vorteil, dass komplexere Anfragen mithilfe von logischen Verkettungen wie „UND“ und „ODER“ ermöglicht wurden. Über dieses Interface konnten zusätzlich strukturelle Anfragen bezüglich Aminosäuren und Nukleotiden definiert

und Abstandskriterien zwischen diesen definiert werden. Die Ergebnisse einer Suche mithilfe der Webschnittstelle wurden in einer Liste dargestellt. Für jeden Listeneintrag konnten dann Informationen des PDB-Eintrags angezeigt oder auch gefundene strukturelle Muster visualisiert werden.

Die Suchmaschine **PDBeMotif/MSDmotif** [119] bietet eine große Anzahl vordefinierter Objekte zur Strukturanfrage und wird durch die *Protein Data Bank in Europe* (PDBe) bzw. das EMBL-EBI bereitgestellt [120]. Diese vordefinierten Objekte bzw. Struktur motive (englisch: *structural motifs*) stellen z.B. kleine Moleküle, Sekundärstruktursequenzen oder charakteristische Anordnungen von Aminosäureresten dar. Sie können teils durch Interaktions- und Distanzbeschränkungen verbunden werden. Die Verbindung mehrerer Proteinsequenzen mithilfe von Distanzbeschränkungen ist beispielsweise nur möglich, wenn die Sequenzabschnitte der gleichen Proteinkette angehören. Eine Suche nach Interaktionsmustern zwischen Protein- und Ligandatomen wird ebenfalls unterstützt. Hierfür werden verschiedene Arten atomarer Wechselwirkungen und Atomabstände unter  $4,25 \text{ \AA}$  zwischen Protein- und Ligandatomen vorberechnet und in der Datenbank gespeichert. In der Suche können Distanzen von bis zu  $4 \text{ \AA}$  und Interaktionen zwischen einem importierten, gezeichneten oder aus einer Liste ausgewählten Liganden und einem Protein-Anfrageobjekt definiert werden. Der Suchmechanismus und Datenbankaufbau der PDBeMotif Methode wird in der Publikation nur grob skizziert. So ist für die Sequenzsuche nur bekannt, dass eine Implementierung des PSI BLAST [121] Algorithmus verwendet wird. Für 3D-Strukturmotiv-Suchen ist bekannt, dass nur Atomkoordinaten von  $C\alpha$ -Atomen und Seitenketten-Endpunkten gespeichert werden und Suchanfragen auf einen Radius von  $16 \text{ \AA}$  um den jeweiligen Liganden beschränkt sind. Die Ergebnisse werden als Liste dargestellt. In dieser können die gefundenen Reste hervorgehoben werden. Auch ist eine Statistik der Ergebnisse verfügbar.

Ein aktueller Ansatz ist die Methode **strucmotif-search** [122]. Diese ist direkt innerhalb des Struktur-Viewers der PDB nutzbar und zeichnet sich insbesondere durch seine hohe Geschwindigkeit aus. Dabei können drei oder mehr Aminosäurereste und Nukleotide mit einem Abstand von bis zu  $15 \text{ \AA}$  [123] zwischen jedem Paar zu einem Anfragemuster kombiniert werden. Gesucht wird mithilfe eines invertierten Indexes. Hierfür wird eine Nachschlagetabelle über alle in der PDB vorhandenen Anordnungen von Aminosäureresten und Nukleotiden erstellt. Jeder Rest und jeder Nukleotid wird durch die Rückgrat- und Seitenketten-Koordinaten beschrieben. Bei Resten werden die  $C\alpha$ -Atome als Rückgrat und das  $C\beta$ -Atom als Seitenkette ausgewählt. Im Falle von Glycin wird das  $C\beta$ -Atom durch die Überlagerung mit Alanin approximiert. Bei Nukleotiden

wird das C4' als Rückgrat und C1' als Seitenkette betrachtet. Durch diese Strategie wird das Suchen auf das Laden aller Vorkommen aus dem invertierten Index reduziert. Dies wird durchgeführt, indem die Anfrage zuerst in Paare unterteilt und diese in der Liste über einen rotationsinvarianten, symmetrischen Deskriptor gesucht werden. Die Deskriptoren basieren hierbei auf der Bezeichnung der Reste (z. B. Serin, Aspartat und Histidin für die katalytische Triade), dem Abstand der beiden Rückgrat-atome zueinander, dem Abstand der Seitenketten und einem Winkel. Letzterer ist definiert über die beiden Vektoren, die das Rückgrat und die Seitenkette jeder Aminosäure verbinden. Die genauen Werte für die Abstände und Winkel werden in sogenannte *Bins* bzw. Klassen eingeteilt. Ähnliche Vorkommen werden durch den gleichen *Bin* identifiziert und über die Berechnung der *root-mean-square deviation* (deutsch: mittlere quadratische Abweichung, RMSD) bewertet. Die Benutzeroberfläche ist dementsprechend einfach gehalten und erlaubt die Auswahl einzelner Reste in der angezeigten Proteinstruktur. Als Ergebnis wird eine Liste von Treffern mit ihren RMSDs ausgegeben. Außerdem können sowohl die gefundenen Motive als auch die gesamte Struktur überlagert dargestellt werden.

#### 3.2.2 Suchverfahren in Proteinbindetaschen

**Relibase** [124–126] ist ein nicht mehr verfügbares Verfahren. Zur Speicherung der Daten werden alle relevanten Informationen aus der Eingabe PDB-Datei extrahiert und mit zusätzlich berechneten Informationen wie Atom- und Bindungstypen in *Extensible Markup Language* (XML) Dateien gespeichert. Aus diesen wird anschließend ein Relibase-spezifisches Binärformat erstellt. Zu den gespeicherten Informationen zählen drei Molekülklassen: Liganden, Proteinketten und Wassermoleküle. Liganden sind hierbei definitionsgemäß alle Moleküle, die weder Protein noch Wassermoleküle darstellen. Das bedeutet, dass Nukleinsäuren, Metallionen und Peptide mit weniger als 21 Aminosäureresten als Liganden klassifiziert werden. Proteinketten bestehen hingegen aus mindestens 21 der proteinogenen Aminosäuren. Nicht kovalent gebundene Sauerstoffatome werden als Wassermoleküle angesehen. Relibase ermöglicht es mithilfe von Schlüsselwörtern und textbasierten Suchen Informationen der PDB-Dateien wie z. B. eine Autorensuche durchzuführen. Auch ist eine Sequenzsuche basierend auf FASTA [127] und eine Substruktursuche für Strukturen im SMILES [128] Format möglich. Eine geometrische Suche kann erstellt werden, indem Ligand- und Proteinatome mit Distanzbeschränkungen verbunden werden. Der Suchraum ist auf Ligand-basierte Bindetaschen beschränkt, die in einem Radius von 7 Å um den Liganden erstellt werden. Das System ermöglicht die Definition von inter- und intramolekularen Distanzen zwischen Atomen. Außerdem

können Ebenen und Winkelbeschränkungen erstellt und Protein-Protein- und Protein-Wasser-Interaktionen gesucht werden.

Die Suche nach ähnlichen Bindetaschen startet mit einer Ähnlichkeitssuche auf dem Anfrage-Liganden, basierend auf einem topologischen Fingerprint. Die Treffer werden dann in einem Subgraph-Matching mit dem ausgewählten Liganden und seiner Umgebung verglichen. Die Resultate werden in einer Liste dargestellt und können in einem 3D-Viewer visualisiert oder anhand einer Statistik evaluiert werden. Dabei werden die Atome, die mit der Suchanfrage übereinstimmen, hervorgehoben. Auch können Ergebnisse auf Grundlage der Proteinketten überlagert dargestellt werden.

Das Programm **CSD-CrossMiner** [129] ist nur kostenpflichtig verfügbar, vereint allerdings mit der Cambridge Structural Database (CSD) [34] und PDB im Gegensatz zu vielen anderen Programmen zwei Datensätze in der Suche. Die verwendete Datenbank beinhaltet Proteinbindetaschenatome, die durch einen Radius von 6 Å um den Liganden definiert sind. Als Liganden sind hierbei alle Moleküle definiert, die mindestens fünf und maximal 100 Atome haben. Für jedes Atom werden neben der Molekülklasse (Protein oder Ligand) pharmakophore Eigenschaften wie Donor, Akzeptor oder hydrophob bestimmt. Auch werden Richtungsangaben wie Ringnormalen oder projizierte Interaktionspunkte für Ringsysteme, Donoren und Akzeptoren annotiert. All diese Informationen bzw. Features werden für einen schnelleren Suchprozess in Fingerprints umgewandelt und indiziert. Die Suchanfrage kann mithilfe dieser Eigenschaften im 3D-Raum anhand einer Beispielstruktur erstellt werden. Hierbei wird jeder Suchpunkt durch eine sogenannte Toleranzkugel beschrieben. Da eine Eigenschaft durch mehrere Punkte dargestellt werden kann, wie z. B. im Fall eines Wasserstoffbrückendonors, bei dem ein Richtungsvektor vorliegt, beschreibt der Radius dieser Kugel die Toleranz hinsichtlich der Lage der Eigenschaft, um die Spezifität der Anfrage steuern zu können. Die Suche beginnt mit der Übersetzung der 3D-Suchanfrage in Distanzen zwischen den Punkten. Die Summe bzw. die Differenz der Radien der Toleranzkugeln geben dabei die unteren und oberen Schranken für die Distanzabstände an. Anschließend wird für die Anfrage ein Fingerprint berechnet und in der Datenbank gesucht. Durch diesen Prozess werden passende Muster gefunden, die dann durch ein 3D-Suchverfahren in einem weiteren Schritt validiert werden, um die vollständige Übereinstimmung mit der Anfrage zu überprüfen. Das Suchverfahren sortiert hierfür die gefundenen Fingerprint-Treffer und evaluiert mithilfe einer Tiefensuche die Beschränkungen, beginnend beim Suchpunkt mit den meisten Restriktionen. Da für eine Bindetasche mehrere Übereinstimmungen gefunden werden können, wird abschließend der Kabsch-Algorithmus verwendet. Der Treffer mit der besten Überlagerung bzw. dem geringsten RMSD wird für die Bindetasche ausgewählt

und zurückgegeben. Eine Liste aller gefundenen Bindetaschen wird ausgegeben und in einem 3D-Viewer mit der Suchanfrage überlagert dargestellt. Eine 2D-Darstellung der Treffer ist ebenfalls möglich.

**PELIKAN** [52] zeichnet sich durch die Möglichkeit Atom-zentrierter Anfragen und die umfangreichen textuellen und numerischen Filter aus. Für Ersteres werden Bindetaschen in einem Radius von  $6,5 \text{ \AA}$  um den Liganden herum definiert. Liganden sind wie bei CSD-Crossminer alle Moleküle mit mindestens 5 und maximal 100 Schweratomen. Für jede Bindetasche werden alle Atome der kleinen Moleküle, Metall-Ionen, Wassermoleküle und Proteine sowie deren Eigenschaften bestimmt. Hierfür wird zunächst der Algorithmus Protoss [38, 130] verwendet, um fehlende Wasserstoffatome hinzuzufügen und sinnvolle Protonierungszustände und Tautomere zu berechnen. Außerdem wird eine Reihe molekularer Interaktionen wie Wasserstoffbrückenbindungen oder  $\pi$ - $\pi$ -Wechselwirkungen bestimmt. Diese werden ebenfalls in der Datenbank abgespeichert. Aus den PDB-Dateien, die zur Erstellung der Datenbank verwendet werden, werden textuelle und numerische Informationen extrahiert. So können wie in den bereits oben vorgestellten Verfahren Filter z. B. bezüglich der Strukturauflösung oder der EC-Nummer gestellt werden. Weiterhin kann die Suche durch Tascheneigenschaften wie die Tiefe eingeschränkt werden. PELIKAN bestimmt die Tascheneigenschaften, indem Bindetaschen mit dem DoGSite-Algorithmus berechnet und eine von ihnen anhand der Liganden-Coverage (deutsch: Abdeckung) ausgewählt wird. Für die Zuordnung einer Tasche zu einem Liganden müssen mindestens 20 % des Liganden innerhalb der Tasche liegen, ansonsten können keine Eigenschaften zugeordnet werden.

Als Suche können textuelle und numerischen Eigenschaften mit 3D-Anfragen kombiniert oder eigenständig durchgeführt werden. Ebenso sind auch reine 3D-Anfragen möglich. Die Erstellung dieser 3D-Suchen ist anhand einer Benutzer-gewählten Struktur oder auch ohne die Verwendung eines Templates möglich. Sie umfasst 1. Suchpunkte auf Basis von Atomen oder Ringzentren, 2. Distanzen oder Interaktionen mit möglichen Abstandsbeschränkungen, sowie 3. Winkel.

Die Suche von PELIKAN ist in fünf Schritte eingeteilt. Zuerst werden die textuellen und numerischen Filter ausgewertet. Hier wird zudem bereits eine Substruktursuche mithilfe von SMiles ARbitrary Target Specification (SMARTS) [54] Mustern durchgeführt, soweit diese vom Nutzer definiert wurden und entweder seltene chemische Elemente beinhalten oder eine bestimmte Größe übersteigen. Anschließend wird das Anfragemuster in Dreiecke unterteilt, die mithilfe einer im Vorhinein berechneten Indexstruktur gesucht werden, um eine Vorauswahl der Punkte zu treffen. Durch die Dreiecksstruktur wird die Umgebung jeden Punktes beschrieben und die Suchmenge reduziert. Im

dritten Schritt werden alle Suchpunkt-Paare mit Distanz- und Interaktionsbeschränkungen nacheinander überprüft. Die Reihenfolge wird durch die Anzahl an erwarteten Treffern hinsichtlich der Punkteigenschaften sortiert, sodass möglichst restriktive Auswertungen zu Beginn stattfinden und die Resultatmenge schneller reduziert wird. Für die resultierende Liste der übriggebliebenen Suchpunkt-Paare wird im vierten Schritt ein Produktgraph erstellt. In diesem wird ein Knoten für jedes Suchpunkt-Paar definiert. Zwei Knoten werden verbunden, wenn diese den Distanz- und Winkelbeschränkungen entsprechen. Eine Clique der Größe der Anzahl an Suchpunkt-Beschränkungen im Produktgraph beschreibt einen validen Treffer. Im fünften Schritt werden abschließend solche SMARTS Suchen durchgeführt, die nicht bereits zu Beginn ausgeführt wurden.

Die Treffer der Suche werden in der grafischen Benutzerumgebung in einer geschichteten Liste dargestellt. So werden die PDB-Strukturen der gefundenen Treffer in Listen angezeigt und nach ihrer Enzymklasse wie z. B. Transferase oder Lyase sortiert. Nach Auswahl einer Klasse werden die gefundenen PDB-Codes angegeben. Sobald ein PDB-Code ausgewählt wurde, werden alle Treffer sortiert nach ihren Taschen gezeigt. Jeder ausgewählte Treffer wird in einem 3D-Viewer mit der Anfragestruktur überlagert. Hier kann aus mehreren Anzeigemodi ausgewählt werden, um die Übersichtlichkeit zu verbessern. Außerdem ist es möglich, eine Liste der Resultate und eine Statistik anzuzeigen und abzuspeichern. Die Statistik beinhaltet Informationen wie die Distanzen zwischen gefundenen Punkten oder häufig beteiligte Sekundärstrukturelemente gemäß der PDB-Annotation.

#### 3.2.3 Vergleich von Suchverfahren

Alle in Kapitel 3.2.1 und 3.2.2 vorgestellten Verfahren, mit Ausnahme von CSD-Crossminer, arbeiten nur auf Basis der PDB. Dies ist vermutlich dadurch begründet, dass dort eine große Menge an Proteinstrukturen öffentlich verfügbar ist und in einem weitestgehend einheitlichen Format vorliegen. Zudem wird die PDB regelmäßig erweitert und liefert wertvolle Informationen zu bekannten Protein-Ligand-Komplexen, die häufig für die Wirkstoffentwicklung verwendet werden. Einige der Programme, wie PELIKAN, erlauben es allerdings eigene Strukturdateien zu verwenden, die dadurch nicht zwingend in z. B. der PDB oder CSD verfügbar sein müssen.

Bei der Entscheidung, ob ganze Proteine oder nur Proteinbindetaschen durchsucht werden sollen, gilt es abzuwägen, was von größerem Interesse ist. Die Verfahren, die Suchen im gesamten Protein erlauben, reduzieren die durchsuchbaren Features meist auf vordefinierte Objekte, da ansonsten die Laufzeit zu hoch wäre. Demgegenüber eröffnen

Verfahren auf Basis von Proteinbindetaschen häufig mehr Möglichkeiten, bieten aber entsprechend kleinere Suchräume. Die beiden vorgestellten Kategorien haben damit ihre individuellen Vor- und Nachteile. Untereinander unterscheiden sich die Methoden der beiden Kategorien zudem auch noch in den verfügbaren Suchobjekten. Während mit dem Verfahren *strucmotif-search* praktisch jedes mögliche Anfragemuster unter der Bedingung von maximal 15 Å Abstand zwischen den Aminosäuren definiert werden kann, können mit *PDBeMotif* nur vordefinierte Objekte gesucht werden. Auch können z. B. bei *CSD-Crossminer* nur pharmakophore Punkte erstellt werden, bei *Relibase* und *PELIKAN* sind hingegen jegliche atomare Anfragen bezüglich der Bindetaschen möglich.

Entscheidend für die Laufzeit der Suche ist neben einer effizienten Datenspeicherung auch der gewählte Suchalgorithmus. Hier werden verschiedene Ansätze praktiziert, die aber häufig auf der gleichen Grundlage basieren. Während *3DinSight* die benutzerdefinierte Suche in eine SQL-Anfrage umwandelt, nutzen Verfahren wie *strucmotif-search*, *CSD-Crossminer*, *Relibase* und *PELIKAN* einen inkrementellen Ansatz unter Verwendung von Fingerprints oder Deskriptoren, die in einer Indexstruktur abgespeichert sind. Die Indexstrukturen sind das Kernstück letzterer Verfahren und beschreiben die Relation der Punkte zueinander, um die Suchmenge zu reduzieren. Diese werden anschließend weiter prozessiert und validiert. Über den Suchalgorithmus von *PDBeMotif* ist nur wenig bekannt, es ist aber zu vermuten, dass zumindest für einige der Suchfunktionen genau wie bei *3DinSight* eine SQL Anfrage an die Datenbank gestellt wird [131, 132].

Die Darstellung der Ergebnismengen wird bei allen vorgestellten Programmen über eine Liste gehandhabt. Häufig können die gefundenen Treffer als 3D-Überlagerung dargestellt werden. Diese Funktion erleichtert das Untersuchen von Ähnlichkeiten und Unterschieden der Treffer untereinander. Die Verfügbarkeit einer Statistikdatei ist häufig ebenfalls äußerst nützlich, z. B. wenn Geometrien von Interaktionen analysiert werden sollen. Hier können in Statistiken nach beendeter Suche z. B. die Verteilung der Distanzen zwischen interagierenden Atomen oder funktionellen Gruppen untersucht oder Eigenschaften bestimmt werden. Ermöglicht wird dies z. B. in den hier vorgestellten Methoden *PDBeMotif*, *Relibase* und *PELIKAN*.

Wie bereits zuvor beschrieben, sind die Suchverfahren kaum zu validieren, evaluieren oder untereinander zu vergleichen. Für *strucmotif-search*, *PDBeMotif*, *Relibase*, *CSD-Crossminer* und *PELIKAN* sind Fallbeispiele bekannt. Eine Überprüfung von falsch-negativen oder falsch-positiven Ergebnissen wird allerdings nur für *PELIKAN* beschrieben. Zu diesem Zweck wurden 200 PDB Strukturen zufällig ausgewählt. Aus diesen wurden Filter von acht bis zehn Atomen mit zufälligen paarweisen Abstandsbereichen

generiert. Für die Filter wurde anschließend überprüft, ob jedes gefundene Muster den Anfrage-Anforderungen entspricht, um falsch-positive Ergebnisse ausschließen zu können. Außerdem wurden die Taschen durch diejenigen Filter wiedergefunden, aus denen der jeweilige Filter generiert wurde.

### 3.3 Geometrische Mustersuche mit GeoMine

Wie zuvor beschrieben, unterscheiden sich Anwendungen zur geometrischen Mustersuche in mehreren Aspekten. Eines der umfangreichsten dieser Tools ist die vorgestellte Anwendung PELIKAN. Diese wurde bereits hinsichtlich der in Kapitel 3.1 beschriebenen Herausforderungen, also der Auswahl des Datensatzes, der Wahl der durchsuchbaren Strukturdaten, dem Suchalgorithmus, der übersichtlichen Ergebnispräsentation und der Validierung, entwickelt, bietet darüber hinaus allerdings weitere Optimierungsmöglichkeiten. So verwendet PELIKAN zwingend eine SQLite-Datenbank. Bei dieser handelt es sich um ein sogenanntes eingebettetes Datenbanksystem, welches vorwiegend als programminterne Datenbank fungiert und zur lokalen Datenspeicherung dient. Generell verwendet SQLite nur einen Prozessor-Thread bzw. blockt Speicheradressen des Dateisystems, sodass eine Parallelisierung nur eingeschränkt möglich ist [133]. Auch fehlen einige Funktionen im Vergleich zu Server-basierten Datenbanksysteme, wie die Verwaltung von Objektberechtigungen. Eine Verwendung als Webanwendung ist grundsätzlich möglich, hat durch die Limitierung auf einen Prozessor-Thread allerdings einige Nachteile bezüglich dieses Anwendungsbereichs. Insbesondere Webanwendungen werden allerdings immer populärer, da auf eine Installation verzichtet werden kann und Nutzer nicht auf das Vorhandensein nötiger Hardware angewiesen sind.

Die Datenbank von PELIKAN unterstützt nur Ligand-basierte Bindetaschen. Dies reduziert die Dauer der Suche im Vergleich zu Suchen im gesamten Protein. Aufgrund des gleichzeitig kleinerem Suchraums verringert sich allerdings ebenfalls die Anzahl durchsuchbarer Taschen und möglicher Treffer. Die Berücksichtigung vorhergesagter Bindetaschen eröffnet einen erheblich größeren Suchraum, da generell die Anzahl an cokrystallisierten Molekülen gering ist und somit nicht alle Bindetaschen von Interesse übergebundene Liganden identifiziert werden können. Auch muss die PELIKAN-Datenbank bei geänderten oder obsoleten Strukturen neu aufgebaut werden, da keine Löschk Funktionalität existiert.

Ausgehend von diesen Limitationen wurde im Rahmen des Promotionsprojekts die geometrische Mustersuche GeoMine basierend auf PELIKAN entwickelt [D2–D5]. Die Hauptunterschiede bzw. Features von GeoMine werden im Folgenden aufgelistet:



- Neben Ligand-basierten Taschen wird die Suche in vorhergesagten Bindetaschen ermöglicht. Taschen werden hierbei mithilfe des DoGSite3-Algorithmus (siehe Kapitel 2.2) berechnet.
- Anstelle einer SQLite-Datenbank kann eine PostgreSQL-Datenbank verwendet werden. Bei dieser handelt es sich um ein Server-basiertes Datenbanksystem, das für eine neu entwickelte Webapplikation von GeoMine verwendet wird. Die Datenbank kann mithilfe einer Kommandozeilen-Anwendung erstellt und durchsucht werden. Außerdem können Einträge nachträglich hinzugefügt oder gelöscht werden. PostgreSQL ermöglicht darüber hinaus als Server-basierte Datenbank eine umfangreichere Nutzung der Hardware. Der eigens für PELIKAN entwickelte Dreiecksindex wurde mangels Laufzeitvorteilen entfernt. Durch Optimierungen der SQL Abfragen werden zudem bessere Laufzeiten erzielt.
- Eine Parallelisierung der Datenbankerstellung ermöglicht den schnelleren Aufbau neuer Datenbanken. Auch können existierende Datenbanken schneller durch neue Strukturen erweitert werden.
- Die Erstellung der Datenbank und die geometrische Suche wurden durch verschiedene Techniken beschleunigt. So wurden Schlüsselfunktionen im C++-Code neu implementiert, SQL-Datenbankabfragen beispielsweise durch die Verwendung von JOINS für kleinere Abschnitte von Tabellen angepasst und Datenbanktransaktionen stärker gebündelt. Auch wurden Datenbankabfragen analysiert und auf das PostgreSQL-Datenbanksystems optimiert.
- Mit GeoMine ist es möglich, neben den Suchpunkten aus PELIKAN, auch Mittel- und Endpunkte von Sekundärstrukturelementen sowie Nukleinsäurepunkte zu definieren. Sind Sekundärstrukturen in den Eingabedateien nicht angegeben, werden diese berechnet und nicht mehr, wie in PELIKAN, ausgelassen. Ebenso ist die Lösungsmittelzugänglichkeit aller Protein- und Nukleinsäure-Atome als weitere Einschränkung an Suchpunkten annotiert. Dies stellt eine der nützlichsten Erweiterung des Suchraums dar: Mit ihr kann zwischen vergrabenen und zugänglichen Proteinatomen unterschieden und somit eine erhebliche Reduzierung des Suchraums und eine zuverlässigere Überlagerung der Taschen erzielt werden. Neue textuelle und numerische Filter wie die Suche nach Taschen, die aus mehreren Proteinketten bestehen, oder das Ausschließen symmetrischer Matches, falls für einen geometrischen Treffer mehrere symmetrische Treffer gefunden wurden, erweitern die Suchmöglichkeiten und die Übersichtlichkeit der Ergebnisse. Eine neue Ähnlichkeitssuche für Liganden stellt zudem eine einfache und schnelle

Möglichkeit zur Reduzierung des Suchraums dar. In [D2] werden die verfügbaren Eigenschaften in GeoMine zu PELIKAN und ähnlichen Suchverfahren abgegrenzt. Eine Übersicht aller Suchmöglichkeiten von GeoMine wird in Kapitel C gegeben.

Neben diesen Punkten wurde ebenfalls eine neue grafische Benutzeroberfläche in Form einer Webapplikation entwickelt [D3]. Diese ist als Teil des Proteins*Plus*-Servers [D4, 134, 135] zur strukturbasierten molekularen Modellierung unter <https://proteins.plus> frei verfügbar und wurde während eines parallel laufenden Promotionsprojektes durch Konrad Diedrich entworfen und entwickelt.

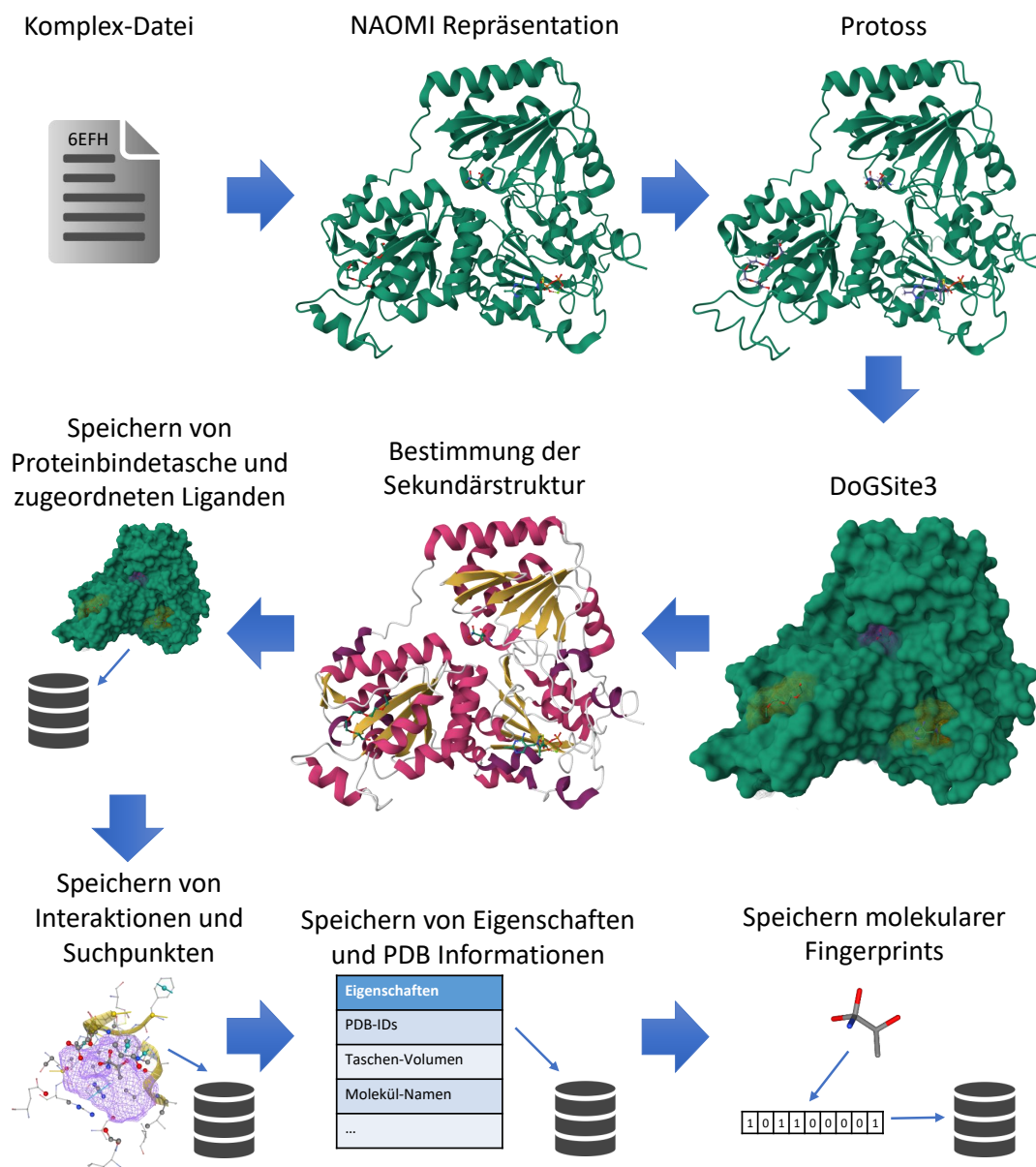
#### 3.3.1 Methodische Zusammenfassung

Der methodische Ablauf von GeoMine lässt sich in zwei Einheiten unterteilen: die Datenbankerstellung und den Suchalgorithmus. Eine grafische Übersicht ist in Abbildung 3.1 bzw. Abbildung 3.2 dargestellt. Im Folgenden ist der Ablauf zusammengefasst beschrieben. Eine detaillierte Methodenbeschreibung ist in der Veröffentlichung des Verfahrens [D2] nachzulesen.

Die **Datenbankerstellung** ist in acht Schritte unterteilt (siehe Abbildung 3.1). Zu Beginn werden eine oder mehrere Dateien im PDB oder mmCIF Format eingelesen und mit der NAOMI-Bibliothek in eine interne Datenstruktur der gelesenen biologischen Makromoleküle konvertiert. Neben kleinen Molekülen, Wassermolekülen, Ionen, Nukleinsäuren und dem Protein werden in diesem Prozess auch Informationen wie der PDB-Code, der Organismus und die Strukturauflösung extrahiert.

Im zweiten Schritt werden mithilfe der Protoss-Bibliothek analog zur PELIKAN-Anwendung fehlende Wasserstoffatome hinzugefügt und sinnvolle Protonierungszustände und Tautomere berechnet. Diese Vorprozessierung bildet die Grundlage für die spätere Berechnung der molekularen Interaktionen.

Für die durch Protoss angepasste Proteinstruktur erfolgt anschließend die Vorhersage der Bindetaschen und deren Eigenschaften, wie die lösungsmittelzugängliche Oberfläche (englisch: *solvent accessible surface area*, SASA), mit dem DoGSite-Algorithmus (siehe Kapitel 2.2). Wie zuvor beschrieben, haben die vorhergesagten Taschen im Vergleich zu den in PELIKAN verwendeten radiusbasierten Taschen unter anderem den Vorteil, dass große Taschen vollständig gefunden und nicht auf Bereiche in unmittelbarer Umgebung des Liganden beschränkt sind. Ein Nachteil ist, dass die Liganden in einem separaten Schritt den Taschen zugeordnet werden müssen. Zu diesem Zweck wird nach Taschen gesucht, die mindestens 20 % aller Schweratome des Liganden beinhalten. Wenn



**Abbildung 3.1:** Programmablauf der Datenbankerstellung von GeoMine dargestellt anhand der Proteinkette A der Proteinstruktur mit dem PDB Code 6EFH. Proteingrafiken wurden mit Mol\* [136] erstellt. Die Grafiken der Schritte des Speicherns von Interaktionen und Suchpunkten sowie der molekularen Fingerprints wurden mit dem *ProteinsPlus* Server [D4, 134, 135] und dem NGL Viewer [137, 138] erzeugt.

für einen Liganden keine solche vorhergesagte Tasche gefunden werden kann, wird zur Berücksichtigung dieser Bindestellen auf die radiusbasierte Bestimmungsmethode mit

einem Radius von  $6,5 \text{ \AA}$  zurückgegriffen (analog zu Kapitel 3.2.2, PELIKAN). Wenn stattdessen mehrere Taschen identifiziert werden, wird der Ligand mit jeder einzelnen Tasche assoziiert, sodass Anfragen mit diesen möglich sind. Zu einer Tasche kann zudem mehr als ein Ligand zugeordnet werden. Zur Limitierung der Datenbankgröße und Sicherstellung der Übersichtlichkeit von Ergebnissen, des Speicherplatzbedarfs und der Suchgeschwindigkeit werden alle Taschen, denen kein Ligand zugeordnet wurde, nach Volumen sortiert. Anschließend werden die größten  $k$  Taschen ausgewählt, wobei  $k$  auf das Doppelte der Anzahl aller Proteinketten in der asymmetrischen Einheit des Proteinkomplexes begrenzt ist. Die asymmetrische Einheit ist als der kleinste Teil einer Kristallstruktur definiert, die durch Symmetrieoperationen die vollständige Einheitszelle beschreibt [139].

Im vierten Schritt werden die Sekundärstrukturelemente aus der Eingabedatei bestimmt. Fehlt diese Angabe, werden die Helices und Faltblätter mit Funktionen der NAOMI-Bibliothek berechnet. Diese Funktionalität beruht auf dem DSSP-Algorithmus [140], wurde in einer Bachelor-Projektarbeit am Zentrum für Bioinformatik im Jahr 2016 entwickelt [141] und im Rahmen dieses Promotionsprojekts in die NAOMI-Bibliothek überführt.

Die letzten vier Schritte beinhalten das Speichern berechneter und extrahierter Informationen in die Datenbank. Dabei werden zunächst die Komplexe mit den kleinen Molekülen, den Wassermolekülen und den Ionen sowie allen vorhergesagten Taschen in die Datenbank geschrieben. Im Gegensatz zu PELIKAN wird dabei jede Tasche jeweils nur einmal in die Datenbank aufgenommen. Da in PELIKAN pro Ligand eine Tasche erstellt wird, kann es zu signifikanten Überschneidungen von Taschen kommen, wenn mehrere Liganden räumlich nah beieinander liegen. Deshalb ist die PELIKAN-Datenbank im Vergleich zur GeoMine-Datenbank größer. Zudem kann es zu Duplikaten in der Ergebnismenge führen, da Proteinatome mehrfach vertreten sind.

Um in den eingefügten Proteinkomplexen geometrische Muster suchen zu können, werden die Protein-Ligand-Interaktionen berechnet. Zu diesen Interaktionen gehören Wasserstoffbrückenbindungen auf Grundlage der vorhergesagten Protonierung und Tautomerie, Wechselwirkungen zwischen aromatischen Ringen, aromatischen Ringen und Kationen, ionische Wechselwirkungen und Metallkoordination. Auch wird jedes Atom und aromatisches Ringzentrum jeder Tasche mit multiplen Eigenschaften wie dem Molekültyp (Protein, Nukleinsäure, Ligand, Metallion oder Wassermolekül), dem chemischen Element, der Lösungsmittelzugänglichkeit und dem Sekundärstrukturtyp in die Datenbank aufgenommen.

Weitere Informationen wie Bindetaschendesriptoren, die PDB-Codes oder Molekül-namen werden für die textuellen und numerischen Filter zusätzlich gespeichert. Diese Informationen wurden bereits beim Einlesen der Eingabedatei oder während der einzelnen Prozessierungsschritte bestimmt.

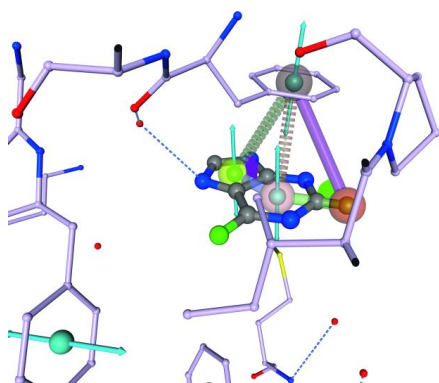
Abschließend werden alle Liganden der Komplexe für Ähnlichkeitssuchen in Fingerprints umgewandelt und der Datenbank hinzugefügt.

Der **Suchalgorithmus** ist in sechs Schritte unterteilt (siehe Abbildung 3.2). Die vom Benutzer erstellte Anfrage kann geometrische, textuelle und numerische Suchkriterien und -muster enthalten. Abhängig von dieser Eingabe können Schritte des Algorithmus übersprungen werden. Die Suche ist dabei so konzipiert, dass frühe Schritte die Suchmenge möglichst signifikant verringern, um zeitaufwendige Schritte wie die geometrische Suche zu beschleunigen.

Zu Beginn werden vorhandene textuelle und numerische Filter auf der kompletten Datenbank angewandt. Mit Eigenschaften wie einer minimalen oder maximalen Tiefe der zu durchsuchenden Taschen, einer Eingrenzung der Proteine auf bestimmte Klassen wie Kinasen oder einer Mindest- bzw. Maximalauflösung der Kristallstruktur kann die Suchmenge enorm reduziert werden. Auch können Moleküle im SMILES-Format verwendet werden, um mithilfe einer Fingerprint-basierten Ähnlichkeitssuche Taschen zu identifizieren, die ähnliche Liganden binden. Dies kann beispielsweise zur Annotation und Funktionsvorhersage für vorhergesagte Bindetaschen verwendet werden. So kann etwa bekannt sein, dass ein Molekül an ein Protein bindet, aber nicht in welcher Bindetasche. Für die Bindetaschen können geometrische Muster definiert und in den Taschen der Datenbank gesucht werden, die ähnliche Moleküle binden. Somit können die Taschen auf eine oder mehrere mögliche ligandbindende Taschen eingeschränkt werden. Der Benutzer kann mit der Ähnlichkeitssuche daher nicht nur die Laufzeit der Suche, sondern auch die Ergebnismenge beeinflussen, um so für seine jeweilige Anwendung zügig relevante Ergebnisse zu erhalten.

Im zweiten Schritt folgt die Auswertung einiger chemischer Substrukturmuster. Wie bei PELIKAN werden hier SMARTS-Muster, die mindestens fünf Atome beschreiben und solche, die seltene Elemente (nicht Kohlenstoff, Sauerstoff oder Stickstoff) enthalten, prozessiert. Diese Vorgehensweise basiert auf der Beobachtung, dass Muster, die diese Randbedingungen erfüllen, den Suchraum stark reduzieren. SMARTS-Muster, die diese Kriterien nicht erfüllen, werden am Ende des Suchalgorithmus ausgewertet, da sie

Geometrisches Anfragemuster



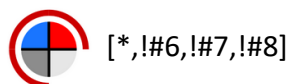
mit textuellen/numerischen Filter

Anwendung von textuellen und numerischen Filtern:

Protein-Filter	Taschen-Filter
PDB-IDs	Tiefe
EC-Nummer	Volumen
Organismus	#Akzeptoren
...	...

Molekül-Filter
#Funktionelle Gruppen
#Chem. Elemente
Ähnlichkeit basierend auf CSFP-Fingerprints
...

Diskriminative SMARTS Analyse von Mustern mit mindestens 5 Atomen oder mindestens einem Atom für welches gilt:



Geometrische Anfrage wird nach SQL konvertiert und ausgeführt:

```
SELECT p1.point_id, p2.point_id, ...
WHERE p1.pocket_key = p2.pocket_key ...
AND ...
AND (p1.x-p2.x) * (p1.y-p2.y) * (p1.z-p2.z) BETWEEN ...
```

Winkel der resultierenden Treffer werden überprüft

Übrige SMARTS Filter werden angewendet

Resultate werden nach RMSD sortiert und ausgegeben

Abbildung 3.2: Programmablauf der GeoMine Suche. Die Molekülgrafik wurde mit der GeoMine Benutzeroberfläche des ProteinsPlus Servers [D4, 134, 135] erstellt.

wahrscheinlich zu zahlreichen Übereinstimmungen führen, was eine längere Laufzeit zur Folge hat.

Nach den ersten beiden Schritten ist neben den potenziell vorhandenen wenig eingrenzenden SMARTS-Mustern die geometrische Anfrage zu verarbeiten. Hierbei wird die geometrische Anfrage zunächst in Paare von Suchpunkten mit Distanz- und Interaktionskriterien zerlegt. Jeder Suchpunkt wird außerdem mit allen Eigenschaften annotiert, die im Suchmuster definiert sind. Anschließend werden alle Kriterien zu einer gebündelten SQL-Datenbankabfrage umgewandelt. Resultat der Abfrage ist eine Zuordnung der Suchpunkte des Anfragemusters zu den Atomen und Ringzentren der gefundenen Bindetaschen der Datenbank, für welche alle Suchpunkt-Eigenschaften und Distanz- und Interaktionskriterien mit denen des Anfragemusters übereinstimmen. Dieser Schritt unterscheidet sich wesentlich vom Algorithmus von PELIKAN. So werden in PELIKAN zuerst alle Punkte gesucht und anschließend alle Distanz- und Interaktionskriterien nacheinander überprüft. Auch wird eine eigens für PELIKAN entworfene Indexstruktur zur schnellen Detektion passender Atome und Ringzentren verwendet (siehe [52]). Die Indexstruktur wird aufgrund ihrer Größe in komprimierter Form abgespeichert und ein Indexzugriff kann nur mit einer teilweisen Dekomprimierung der Indexstruktur durchgeführt werden. Im Laufe dieses Promotionsprojektes wurde festgestellt, dass bei steigender Anzahl an Proteinstrukturen in der Datenbank keine Laufzeitverbesserungen mehr mit der Indexstruktur erzielt werden können. Aus diesem Grund verwendet GeoMine keine eigens entwickelte Indexstruktur. Zur Beschleunigung des Datenbankzugriffs werden stattdessen einzig Indexstrukturen auf Datenbanktabellen mithilfe der Standard-Indexierung von PostgreSQL erstellt und genutzt.

Mit der aus Schritt vier erhaltenen Zuordnung von Anfrage- und Ergebnispunkten sind alle zur Anfrage passenden Atome bekannt. Um die geometrische Suche abzuschließen, müssen benutzerdefinierte Winkelkriterien zwischen Distanzen, Interaktionen und Vektoren wie den Ringnormalen in einem weiteren Schritt ausgewertet werden. Da die Punkte in der Zuordnung bereits auf korrekte Distanz- und Interaktionskriterien überprüft wurden, ist dies ein rein mathematisches Problem und erfordert keine weitere Suche in der Datenbank.

SMARTS-Muster, die im zweiten Schritt des Suchalgorithmus nicht verarbeitet wurden, werden im fünften Schritt auf die Ergebnisse angewandt. Dazu gehören z. B. SMARTS-Muster, die Teile einer Atomumgebung oder chemische Beziehungen zwischen Suchpunkten beschreiben. All diese Muster werden für jedes der übereinstimmenden Atome überprüft und bei Nichterfüllung des SMARTS verworfen.

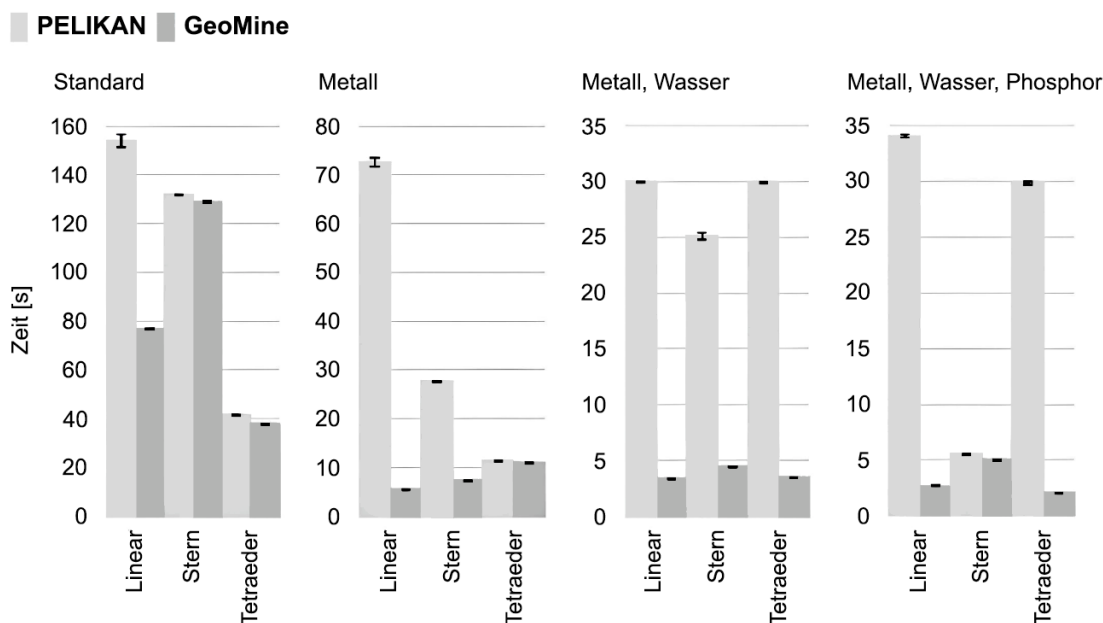
Abschließend werden alle Resultate nach dem RMSD sortiert. Die Sortierung ermöglicht eine Priorisierung der Treffer anhand ihrer Ähnlichkeit zur Suchanfrage. Auch können große Mengen an Ergebnissen auf die ähnlichsten reduziert und in der Benutzeroberfläche dargestellt werden.

#### 3.3.2 Evaluation und Performanz

Zur Evaluation wurde GeoMine während der Entwicklungsphase kontinuierlich mit PELIKAN verglichen. Dies sollte gewährleisten, dass die gleichen Ergebnisse bezüglich der radiusbasierten Taschen gefunden werden. Ebenfalls wurden für alle Schritte in der Datenbankerstellung und dem Suchalgorithmus neue Softwaretests für die Implementierung erstellt und bereits vorhandene bei Bedarf korrigiert und erweitert.

Die geringe Suchzeit der geometrischen Suchen mit GeoMine ist insbesondere für die Nutzung als Webservice von Vorteil. Am einfachsten lässt sie sich mit der von PELIKAN vergleichen, da die gleichen Ausgangspunkte wie Filter und Taschen verwendet werden können. Dennoch ist zu beachten, dass beide Anwendungen unterschiedliche Datenbanktechnologien verwenden. So liest SQLite direkt vom Laufwerk, wohingegen PostgreSQL zuerst eine Netzwerkverbindung aufbauen muss. Auch führt der PostgreSQL-Server eigenständig Abfragen auf multiplen Threads des Prozessors aus, während in SQLite jede Datenbankverbindung nur jeweils einen Thread erlaubt [142]. Dies ist insbesondere für einen Webservice mit vielen Nutzern ineffizient. In 2021 durchgeführten Laufzeitanalysen ist GeoMine bis zu zehnmal schneller als PELIKAN (siehe Abbildung 3.3), was neben der Nutzung von mehr Threads durch PostgreSQL auch durch Optimierungen der Datenbankabfragen begründet werden kann. Für die Analysen wurde dabei ein handelsüblicher Computer mit einem Intel i5-9500 (3,0 GHz) Prozessor, 16 GB Arbeitsspeicher (6 GB durch PostgreSQL nutzbar) und eine Toshiba BG4 PCIe SSD (512 GB, Modell NVMe) verwendet. Als Datensatz wurde die sc-PDB [35, 103] aus dem Jahr 2017 ausgewählt. Die Anfragemuster bzw. GeoMine-Filter wurden aus [143] entnommen. Diese wurden von Therese Inhester für Analysen von PELIKAN entworfen. Drei geometrische Abfragetypen (Linear, Stern und Tetraeder) werden hierbei mit unterschiedlichen Atomeigenschaften als Abfragepunkte verwendet. Insbesondere geometrische Muster des Typs „Linear“ sind mit GeoMine deutlich schneller zu finden. Dies ist auf die im Vergleich zu anderen Mustern geringe Anzahl an Distanzkriterien zurückzuführen. Die Optimierungen der Datenbankabfragen in GeoMine beschleunigt diesen Schritt signifikant, allerdings bleibt er weiterhin zeitkritisch, da hierbei die Atomkoordinaten aller Punkte in Beziehung zueinander gesetzt werden.





**Abbildung 3.3:** Durchschnittliche Laufzeit der geometrischen Mustersuche mit Testanfragen auf PostgreSQL und SQLite Datenbanken des sc-PDB-Datensatz [35, 103] von 2017. Jeder Balken zeigt den Mittelwert von fünf unabhängigen Experimenten an. Für jeden geometrischen Abfragetyp gibt es eine „Standard“-Abfrage, die aus Sauerstoff-, Stickstoff- und Kohlenstoffatomen besteht, eine „Metall“-Abfrage, bei der einer der Abfragepunkte in ein Magnesiumion geändert wird, eine „Metall, Wasser“-Abfrage mit einem Magnesiumion und einem Wasser-Abfragepunkt und eine „Metall, Wasser, Phosphor“-Abfrage, bei der ein dritter Punkt in ein Phosphoratom geändert wird. Entnommen aus [D2] und ins Deutsche übersetzt.

Aufbauend auf diesen Laufzeitvergleichen wurden von Poppinga et al. in [D5] weitere Optimierungen der SQL-Abfragen implementiert. So konnte die Laufzeit im Schnitt um das Dreizehnfache reduziert werden. Bestandteil der Optimierungen sind: 1. Ersetzen des sequentiellen Suchalgorithmus durch ein parallel laufendes Verfahren, welches alle geometrischen Anfragen in einer einzigen SQL-Abfrage durchführt, 2. Hinzufügen eines neuen Index auf der Tabelle, die zur Suche der geometrischen Muster verwendet wird, 3. Entfernen von *Wildcards* (deutsch: Platzhalter) in PDB-Code Abfragen, und 4. Vereinheitlichung der PDB-Codes durch Großschreibung, sodass eine Beachtung der Groß- und Kleinschreibung in Abfragen vermieden wird. Speziell die erste Optimierung erzielt deutliche Laufzeitverbesserungen von etwa 35 % ohne Berücksichtigung der anderen Optimierungen. Die Verringerung der Laufzeit beruht darauf, dass Punkte, Distanzen und Interaktionen nicht mehr voneinander getrennt abgefragt werden müssen, sondern ohne ein wiederholtes Aufbauen von Verbindungen zur Datenbank in einer Abfrage gesucht werden können. Durch die optimierte SQL-Abfrage übernimmt der PostgreSQL-Server das Planen und Durchführen einer möglichst effizienten Suche. Die

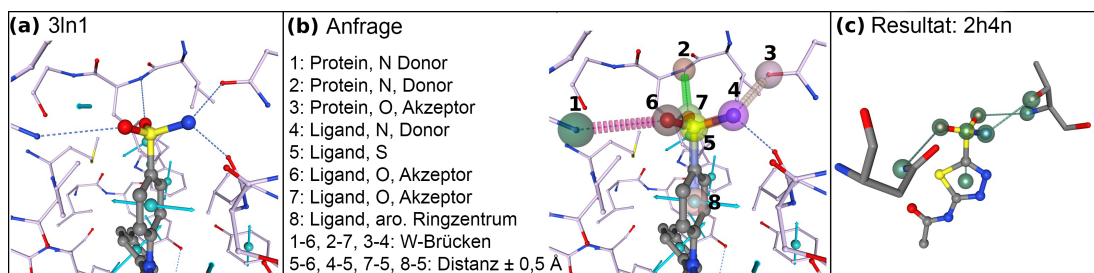
vierte Optimierung reduziert die Laufzeit ebenfalls signifikant mit etwa 30 %, da eine Zeichenkette nun mit einer Abfrage direkt verglichen werden kann, anstatt jedes Zeichen auf Groß- bzw. Kleinschreibung überprüfen zu müssen. In Kombination führen alle vier Optimierungen dazu, dass die Dauer aller getesteten SQL-Abfragen aus [D5] von durchschnittlich über 200 Sekunden auf unter 10 Sekunden reduziert werden kann. Die Laufzeiten wurden hierbei auf einem Server mit zwei Intel Xeon Gold 6248 (2,50 GHz) Prozessoren, 376 GB Arbeitsspeicher und einer Dell PM1725b SSD (1,6 TB, Modell NV-Me) durchgeführt.

Neben den Laufzeitverbesserungen der Suche wurde auch die Erstellung der Datenbank überarbeitet. Zu diesem Zweck durchgeführte Laufzeitanalysen zeigen einen deutlichen Nutzen der Optimierungen dieses Prozesses in GeoMine gegenüber PELIKAN. Insbesondere die Optimierung von Oberflächenberechnungen im Rahmen dieser Arbeit (Beschreibung siehe Kapitel D.2) und die Parallelisierung der Datenbankerstellung reduzieren den Zeitaufwand. Die Analysen sind nicht Teil einer der veröffentlichten Publikationen, können aber in Kapitel B.2 nachgelesen werden. Sie zeigen neben der Laufzeitreduktion ebenfalls eine Verringerung des benötigten Speicherplatzes der Datenbank.

#### 3.3.3 Anwendungen

Wie bereits zu Beginn dieses Kapitels beschrieben, gibt es eine Reihe von Anwendungsgebieten für Methoden der geometrischen Mustersuche und damit auch für GeoMine. Beispielsweise Interaktionsmuster analysiert werden. So beschreiben Günther et al. [126], wie die Methode Relibase verwendet werden kann, um bevorzugte Interaktionsgeometrien von Wasser-unterstützten zweizähligen Salzbrücken zu analysieren. Als Suchmuster werden Punkte, Abstands- und Winkelkriterien definiert, wie es auch mit GeoMine möglich ist. Anhand dieses Beispiels kann gezeigt werden, wie leicht solche Anfragen mit Methoden wie GeoMine erstellt werden können, um umfangreiche und ansonsten manuelle Suchen in kürzester Zeit durchzuführen.

Die Identifizierung von Protein-Ligand-Komplexen mit ähnlichen Interaktionsmustern stellt eine weitere Anwendung dar. Diese Information kann z. B. genutzt werden, um sogenannte *Off-Targets* zu finden. Bei diesen handelt es sich um eine ungewünschte Interaktion des Liganden mit einem anderen Target, dessen Aktivität nicht durch den Liganden beeinträchtigt werden sollte. In [D3] wird eine Anfrage mit GeoMine beschrieben, um mögliche Off-Targets zu finden (siehe Abbildung 3.4). Dabei wird der Wirkstoff Celecoxib mit einigen Suchpunkten und Distanzeinschränkungen zu Proteinatomen der



**Abbildung 3.4:** Anfragemuster für die Off-Target-Suche des Wirkstoffs Celecoxib (a, b) mit GeoMine. (c) zeigt ein beispielhaftes Ergebnis der Suche in einer Bindetasche der Carboanhydrase II mit dem gebundenen Inhibitor Acetazolamid. Die Strukturen, die der Anfrage entsprechen, sind im Ergebnis grün hervorgehoben. Entnommen aus [D3] und ins Deutsche übersetzt.

Bindetasche verwendet. Die Suche lieferte unter anderem Treffer in der Bindetasche von Carboanhydrase II, welche bereits in Studien als *Off-Target* identifiziert wurde.

Sakalli et al. [144] verwendeten GeoMine, um Interaktionsprofile der wichtigsten Bestandteile des ätherischen Eukalyptusöls mit dem Angiotensin-konvertierenden Enzym 2 (ACE2) und Lipoxygenase Enzym (LOX) vorherzusagen. Die Öle werden insbesondere bei Erkrankungen der oberen Atemwege als pflanzliche Arzneimittel eingesetzt. *In vitro* Studien konnten die Vorhersagen bestätigen. Dank der Analysen konnten somit auch strukturelle Informationen zu möglichen Interaktionsmustern gewonnen werden.

In einer weiteren Studie konnten Faria et al. [145] mithilfe von GeoMine Interaktionsmuster potenzieller Protein-Protein-Bindestellen erforschen. Sie fanden heraus, dass bestimmte Typen von Protein-Tyrosin-Phosphatasen (LMWPTP) mit einer Bindestelle der Acetyl-CoA-Carboxylase interagieren, die ebenfalls zu einer Bindung mit dem Pigment Violacein des Bakteriums *Chromobacterium violaceum* führt. Die Protein-Tyrosin-Phosphatasen stellen dabei Biomarker dar, die verwendet werden, um Darmkrebs zu detektieren. Da Violacein Zelltod in mehreren Arten von Krebszellen einleitet, könnte diese Entdeckung ebenfalls Implikationen für Darmkrebs haben.

Weitere Anwendungsbeispiele wie die Ausnutzung ungewöhnlicher Interaktionen, um selektive Inhibitoren zu entwerfen und die Suche nach initialen Molekülfragmenten für den Fragment-basierten Wirkstoffentwurf mit GeoMine oder ähnlichen Anwendungen sind in [D2], [D3], [52], [125], [126], [146] und [147] beschrieben. Insbesondere werden in [D2] Anwendungsbeispiele gezeigt, die die in GeoMine neu hinzugefügten Suchfeatures wie die Angabe der lösungsmittelzugänglichen Oberfläche verwenden.

#### 3.3.4 Ausblick

Trotz der umfangreichen Möglichkeiten zum Erstellen von Anfragen und Optimierungen des Suchalgorithmus gibt es eine Vielzahl möglicher Erweiterungen von GeoMine. So lag ein Fokus der Entwicklung auf der Integration der PostgreSQL-Datenbanktechnologie. Wie bereits erwähnt, zeigte sich die eigens für PELIKAN entworfene Indexstruktur als nicht mehr profitabel hinsichtlich der Laufzeit. Ausschlaggebend dafür ist zum einen eine hohe Dekomprimierungsdauer des Indexes und zum anderen der Suchalgorithmus selbst. Letzterer greift in PELIKAN zuerst auf die Datenbank zu, um die Anzahl möglicher Treffer zu reduzieren, und verwendet diese im Anschluss in mehreren weiteren Datenbankabfragen. In Laufzeitanalysen mit der PostgreSQL-Datenbanktechnologie zeigte sich keine Verschlechterung bzw. sogar teilweise eine Beschleunigung der Anfragen ohne die Indexstruktur, weshalb diese entfernt wurde. Im späteren Verlauf der Entwicklung von GeoMine wurden die iterativen Datenbankabfragen durch Martin Poppinga überarbeitet und durch eine einzelne gebündelte Datenbankabfrage ersetzt (siehe [D5]). Das neue Suchverhalten wurde bisher nicht mit der Indexstruktur aus PELIKAN getestet. In folgenden Untersuchungen muss daher analysiert werden, ob und in welchem Ausmaß die Verwendung der Indexstruktur von Vorteil wäre. Dies gilt dann mit dem höheren benötigten Speicherplatz abzuwägen.

In PDB-Dateien ist der PDB-Code angegeben und gemäß der Dokumentation (Version 3.30) auf vier Zeichen limitiert [148]. Aus diesem Grund werden in NAOMI ebenfalls vier Zeichen aus der Eingabedatei gelesen und in GeoMine zur Unterscheidung der Strukturen verwendet. Da es beispielsweise in der AlphaFold Datenbank [21] bereits jetzt über 214 Millionen Proteinstrukturmodelle gibt, kann GeoMine diese auf Basis der vier Zeichen nicht voneinander unterscheiden. Eine Lösung für dieses Problem wäre die Unterstützung von Zeichenketten mit mehr als vier Zeichen für den Proteinstruktur-Code im Dateiformat oder die Verwendung des Dateinamens zur Unterscheidung in der Datenbank. Bei der Verlängerung dieses Eintrags in der Datenbank muss beachtet werden, dass die PDB-Codes derzeit nur in Großbuchstaben abgespeichert werden. Eine Datenbanksuche unabhängig der Groß- und Kleinschreibung erhöht die Laufzeit der Suche deutlich [D5].

Die geometrische Mustersuche ist das Kernstück von GeoMine. Mit dieser können zahlreiche Eigenschaften beschrieben und in Suchen integriert werden. Die Flexibilität dieser Muster ist allerdings größtenteils auf die benutzerdefinierten Toleranzen der Distanz- und Winkelkriterien beschränkt. Mehr Flexibilität und eine Erweiterung der bisher möglichen „UND“-Beziehungen von Suchpunkteigenschaften könnte mithilfe von

„ODER“ und „NICHT“-Beziehungen eingeführt werden. Beispielsweise könnte dadurch ein Protein-Suchpunkt definiert werden, der sowohl die Aminosäure Alanin als auch Leucin oder z. B. nicht Isoleucin treffen darf.

Das Bestimmen einer Laufzeiteinschätzung abhängig vom Anfragemuster würde die Anwendbarkeit von GeoMine erheblich steigern. Insbesondere Anfragen, deren Laufzeit wenige Minuten überschreiten, können Nutzer irritieren und ihren Arbeitsablauf hemmen oder unterbrechen, da für den Nutzer nicht ersichtlich ist, wann mit Ergebnissen zu rechnen ist. Dies wurde bisher nicht umgesetzt, da intensive Analysen von Mustieranfragen bzw. Statistiken über die in der Datenbank vorliegenden geometrischen Eigenschaften und ihrer Abruf-Laufzeiten notwendig sind. Durch die umfangreichen Möglichkeiten zur Erstellung der Muster gibt es zahlreiche Eigenschaften, die in Kombination betrachtet werden müssen. Zudem ist das Aufbauen der Statistik nach jedem Aktualisieren der Datenbank erneut durchzuführen, falls sich das Verhältnis der Anzahl an einzelnen Eigenschaften ändert.

Eine weitere Möglichkeit zur Verbesserung der Benutzerfreundlichkeit von GeoMine ist das instantane Anzeigen gefundener Treffer. Dies könnte beispielsweise umgesetzt werden, indem Cluster von Taschen berechnet werden. Ein Repräsentant jedes Clusters könnte anschließend zuerst durchsucht und Ergebnisse angezeigt werden. Dadurch wäre es dem Nutzer zudem möglich einen Fokus auf bestimmte Cluster zu legen oder eine Suchanfrage vor deren endgültiger Fertigstellung abubrechen, falls Anfragen angepasst werden sollen. Eine Bedingung für die Erstellung der Cluster ist eine Evaluation der Wahl der Cluster und wie viele mögliche Ergebnisse durch die Betrachtung einzelner Repräsentanten verloren gehen würden.

Die Integration des SIENA-Verfahrens [149, 150] kann für eine allgemeine sequenzbasierte Suche nach strukturell ähnlichen Bindetaschen verwendet werden. Mit dieser kann eine Vorauswahl von interessanten Proteinen getroffen werden. Ein erster Ansatz zur Integration von SIENA in GeoMine wurde in einer Masterarbeit [S1] entwickelt. Auf dieser Basis könnte eine sequenzbasierte Suche realisiert werden.

Die Nutzung textueller und numerischer Filter kann die Suche deutlich beschleunigen, allerdings ist für Nutzer nicht erkennbar, welcher Filter die Suchmenge wie stark reduziert. Eine Ausgabe dieser Information könnte es Nutzern ermöglichen, z. B. leichter Fehler wie einen falsch geschriebenen Molekülnamen zu erkennen.

Weiter könnten statt Bindetaschen ganze Proteine in GeoMine zur Suche verfügbar gemacht werden. Hier wäre allerdings abzuschätzen, wie hoch der Nutzen sein würde.

### *3 Geometrische Mustersuche*

---

Außerdem wären weitere Optimierungen der Laufzeit unabdingbar, wenn Ergebnisse schnell berechnet werden sollen, da die Anzahl an Atomen erheblich steigen würde.

## Kapitel 4

# Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen

Neben den vergleichsweise kleinen und tiefen Proteinbindetaschen für kleine Moleküle gibt es auch große und meist flache [151] Bindestellen auf der Proteinoberfläche, nachfolgend auch Interfaces genannt. Diese dienen häufig der Interaktion mit anderen Proteinen und sind für viele biologische Funktionen in Organismen essenziell. So sorgen diese Protein-Protein-Interaktionen (PPIs) z. B. für die Übertragung von Signalen und passen die Aktivität von Enzymen im Stoffwechsel an [152]. Phizicky und Fields [153] fassen die wichtigsten fünf Eigenschaften von PPIs wie folgt zusammen: PPIs können 1. kinetische Eigenschaften von Enzymen verändern; 2. die Kanalisierung von Substraten ermöglichen; 3. neue Bindestellen für kleine Moleküle bilden; 4. Proteine hemmen oder unterdrücken; 5. die Spezifität von Proteinen zu ihren Substraten durch Interaktion mit anderen Bindungspartnern ändern. Aufgrund ihrer Wichtigkeit für biologische Prozesse werden PPIs immer relevanter für beispielsweise die Bestimmung der Protein- und Genfunktion [154, 155] und die Wirkstoffentwicklung, in der sie neue Möglichkeiten für die Bekämpfung und Behandlung von Krankheiten bieten [156–158]. Im Bereich der Wirkstoffentwicklung gibt es bisher einige Erfolge [159], aber weniger als für Bindestellen kleiner Moleküle [160]. Dies hat zum Teil mit den Eigenschaften dieser Interaktionsregionen zu tun. Wie bereits erwähnt, sind diese Regionen den Proteinbindetaschen kleiner Moleküle meist unähnlich. Dies hat zur Folge, dass kleine Moleküle oft nur mit geringer Affinität an diese Protein-Protein-Bindestellen binden [161–163]. Die Bindungsenergie von PPIs basiert auf der Summe großer Anzahlen schwacher Interaktionen [151]. Die relevanten Interaktionspartner liegen auf der Proteinoberfläche zwar meist in sogenannten Hotspot-Regionen vor, sind aber über die Oberfläche verteilt, sodass häufig insbesondere

kleine Moleküle nicht gut mit ihnen interagieren können [151]. Es ist allerdings dennoch möglich, Wirkstoffe zu entwickeln. Einige Beispiele für zugelassene Medikamente sind: Venetoclax [164], Cabazitaxel [165], Eptifibatide [166, 167], Tacrolimus [168] und Maraviroc [169, 170]. Auch wächst die Datenlage an experimentell bestätigten PPIs derzeit schneller an [171], sodass computergestützte Methoden auf Basis dieser Daten besser validiert werden können. Dies vereinfacht unter anderem die Entwicklung von *in silico* Verfahren zur Unterstützung der Wirkstoffentwicklung.

Interagierende Proteinketten können als Homomere (Ketten mit identischer Sequenz) oder als Heteromere (Proteinketten mit verschiedenen Sequenzen) vorliegen. Weiter werden sie als obligatorisch bzw. permanent (*obligate*) und vorübergehend (*transient*) klassifiziert [172]. *Obligate* PPIs werden durch Proteine eingegangen, die ihre Funktion nur ausüben können, wenn sie im Komplex mit anderen Proteinen vorliegen [173]. Häufig weisen sie im Vergleich zu den *transient* PPIs einen stärkeren hydrophoben Effekt und eine bessere Komplementarität der Form auf. Gebildet werden sie häufig durch die Interaktion von Homodimeren. Die *transient* PPIs sind hingegen typischerweise kleiner und weisen mehr polare und geladene interagierende Reste auf. Sie werden häufig zwischen Enzymen und Inhibitoren oder Hormonen und Rezeptoren beobachtet [173]. Die Analyse von Proteinen anhand ihrer Interaktionen kann die Entwicklung von computergestützten Verfahren vereinfachen und neue Ansatzpunkte in der Medikamentenforschung liefern [174].

Im Folgenden wird zuerst ein Einblick in verschiedene Methoden zur Vorhersage und Analyse von PPIs und Protein-Protein-Bindestellen gegeben. Anschließend wird die Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen als Methode in der Wirkstoffentwicklung vorgestellt und anhand beispielhafter Verfahren erläutert. Im letzten Teil dieses Kapitels wird schließlich der entwickelte Lösungsansatz für die Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen vorgestellt.

### 4.1 Vorhersage und Analyse von PPIs

Methoden zur Vorhersage und Analyse von PPIs lassen sich in die Kategorien der *in vitro*, *in vivo* und *in silico* einteilen. *In vitro* bezeichnet Methoden außerhalb von Lebewesen in einer kontrollierten Umgebung und *in vivo* Studien an lebenden Organismen. Aufgrund der hohen Kosten und dem Zeit- und Arbeitsaufwand der *in vitro* und *in vivo* Verfahren werden zudem computergestützte *in silico* Methoden durchgeführt. Diese werden sowohl als alternative Methoden als auch zur Ergänzung von experimentellen Methoden verwendet. Grob können die computergestützten Methoden in vier



Kategorien eingeteilt werden: 1. Maschinelle Lernverfahren; 2. Erkennung von Hotspots und Bindungsstellen; 3. Docking und virtuelles Screening; 4. Netzwerkanalysen und sequenzbasierte Methoden (siehe Abbildung 4.1) [174]. Im Folgenden werden diese vier Kategorien erläutert und beispielhaft einige Verfahren vorgestellt.

#### 4.1.1 Maschinelles Lernen

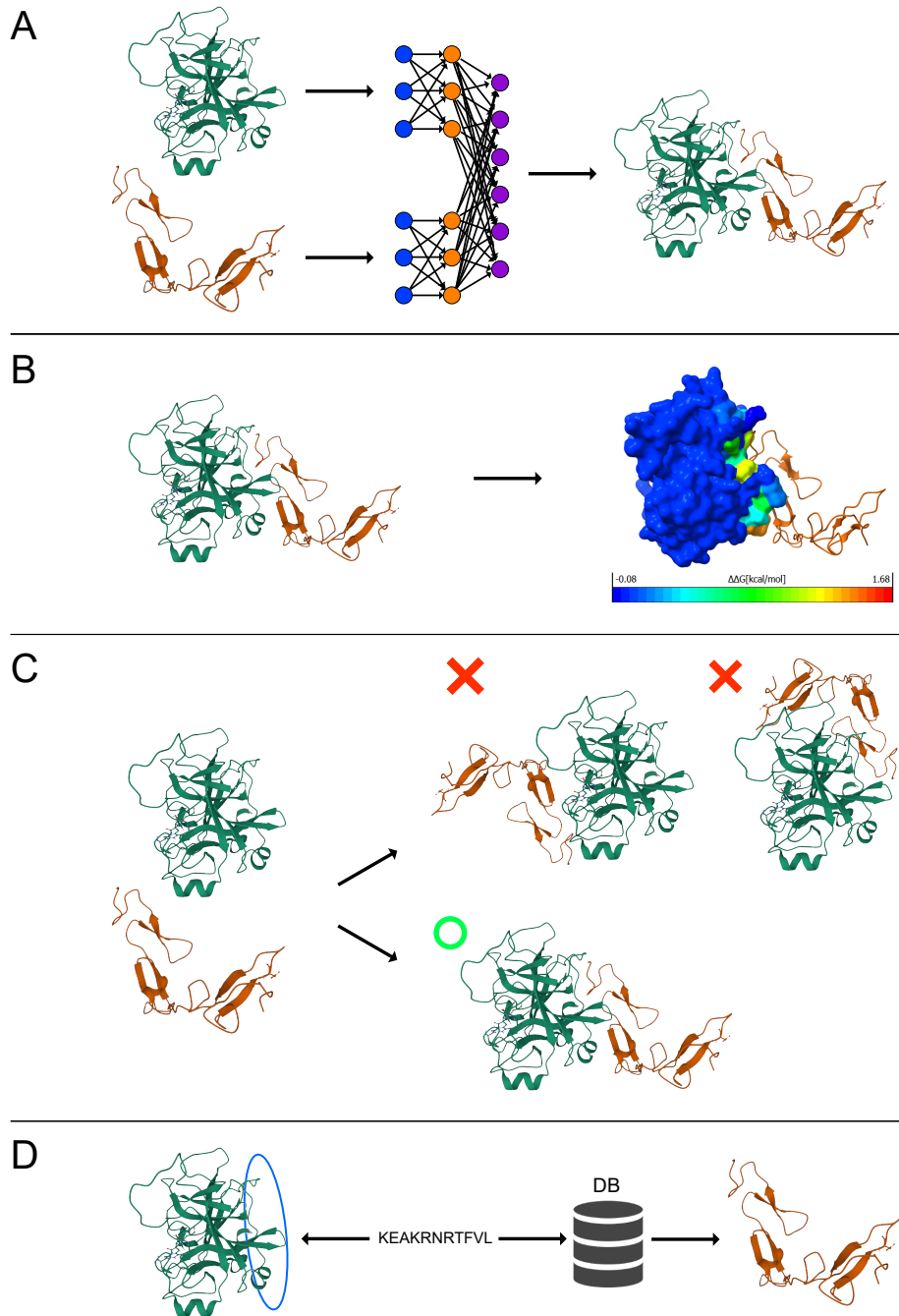
Maschinelle Lernverfahren (ML) werden unter anderem zur Vorhersage von PPIs, Evaluierung von PPI-Vorhersagen und Analyse von PPI-Netzwerken (PPIN) verwendet [176–178]. Die Verfahren funktionieren im Grunde analog zu den bereits vorgestellten Methoden zur Proteinbindetaschenvorhersage (siehe Kapitel 2.1.4). Es werden experimentell bestimmte PPI Daten zum Trainieren eines Modells verwendet, das anschließend Vorhersagen selbstständig treffen kann. In Modellen häufig verwendete Eigenschaften sind z. B. die Aminosäuretypen, die physikochemischen Eigenschaften von Aminosäureresten, evolutionäre Informationen, Sekundär- und Tertiärstrukturen, Proteinoberflächenform und die lösungsmittelzugängliche Oberfläche [179, 180].

Die Methode DeepPPI [181] ist ein solches ML Verfahren und trainiert ein sogenanntes Deep Neural Network, welches vorhersagen soll, ob zwei gegebene Proteine miteinander interagieren. Das Modell basiert hierbei auf Proteinsequenzen und physikochemischen Eigenschaften. So wird beispielsweise die Anzahl der Aminosäuretypen beider Proteine gezählt, die Proteinsequenz verwendet und Attribute wie die Hydrophobizität und das normalisierte van-der-Waals Volumen bestimmt.

#### 4.1.2 Hotspots und Bindungsstellen

Wie bereits in der Einleitung dieses Kapitels erwähnt, findet die Interaktion von Proteinen meist an sogenannten Hotspot-Regionen statt, die zu konformationellen bzw. strukturellen Änderungen der Proteinstruktur führen können [173]. Um einen Wirkstoff zu entwickeln ist es deshalb häufig von Interesse diese Hotspots zu identifizieren, da sie die Bindungsstelle charakterisieren können.

Eine Methode zur Identifizierung von Hotspots ist die *alanine scanning mutagenesis* oder auch Alaninscan-Methode. Bei diesem Verfahren werden einzelne Aminosäuren gegen Alanin ausgetauscht. Die Substitution mit Alanin verursacht den Verlust von Seitenkettenatomen, sodass die Aminosäuren ihre mögliche Funktion im Protein nicht mehr ausüben können. Eine beobachtete Erhöhung der freien Bindungsenthalpie kann dann auf eine Beteiligung dieses Aminosäurerests an der PPI zurückgeführt



**Abbildung 4.1:** Übersicht der vier Kategorien zur Vorhersage und Analyse von PPIs. A: Maschinelles Lernverfahren. B: Erkennung von Hotspots und Bindungsstellen veranschaulicht mithilfe der Alanin-Scanning-Methode  $\text{DrugScore}^{PPI}$  [175]. Die Farben des linken Proteins geben die Änderung der freien Bindungsenergie an. C: Docking und virtuelles Screening. D: Netzwerkanalysen und sequenzbasierte Methoden. Moleküldarstellungen wurden für den Proteinkomplex (PDB-Code: 1AUT) mithilfe von Mol\* [136] und UCSF ChimeraX [109] erstellt.

werden [182]. Das Verfahren kann neben der *in vitro* Anwendung auch mit dem Computer umgesetzt werden. Hierfür ist allerdings die Bestimmung der Änderung der Bindungsenergie zwingend notwendig [183]. Eine Analysemethode für PPIs, die dieses Verfahren umsetzt, ist beispielsweise der öffentlich nutzbare Webservice DrugScore<sup>PPI</sup> [175]. Mit diesem können Hotspots identifiziert werden, die besonders wichtig für eine gegebene Protein-Protein-Interaktion sind. Aus den gewonnenen Daten können dann z. B. mit Docking-Methoden weitere mögliche Interaktionspartner identifiziert oder über einen Abgleich mit Hotspots anderer Protein-Bindestellen auf Komplementarität überprüft werden [174].

### 4.1.3 Docking und virtuelles Screening

Protein-Protein- und Protein-Peptid-Docking hat sich aufgrund der steigenden Anzahl bekannter Proteinstrukturen und verbesserter Einschätzung der Bindungsenergie in den letzten Jahren signifikant weiterentwickelt [184]. Häufig wird es im virtuellen Screening verwendet, um riesige Molekülmengen auf potenzielle Interaktionspartner zu untersuchen. Dabei werden für ein Protein von Interesse eine Vielzahl an Konformeren generiert. Anschließend erfolgt eine Suche für jede Konformation, die mögliche Orientierungen zum Eingehen einer Interaktion mit dem Bindungspartner finden soll. Diese Orientierungen werden dann unter Verwendung der gesamten Strukturinformationen mithilfe von Bewertungsfunktionen überprüft und Komplexe berechnet.

Ein etabliertes Docking-Programm ist HADDOCK (*high ambiguity driven protein-protein docking*) [185]. Es basiert auf einem dreistufigen Verfahren, in dem der erste Schritt eine Randomisierung der Proteinorientierung und anschließende Minimierung der intermolekularen Energie durch Rotation um den Massenschwerpunkt beinhaltet. Anschließend werden die beiden Proteine gedockt, indem die Energie des resultierenden Komplexes minimiert wird. Die Proteinstrukturen sind hierbei starr. Etwa 200 Resultate werden vom ersten in den zweiten Schritt des Algorithmus übernommen. In diesem werden drei *Simulated Annealing*-Approximationsverfahren [186] zur Verbesserung der Ergebnisse durch Anpassung der Torsionswinkel gestartet. Diese optimieren nacheinander die Orientierung, die Seitenketten, sowie zuletzt Proteinrückgrat und Seitenketten zusammen. Im letzten Schritt werden die Atome, die nicht Teil der betrachteten Bindestelle sind, durch ein *Simulated Annealing* mit Lösungsmittel in Form von berechneten Wassermolekülen optimiert. Die resultierenden Strukturen werden abschließend geclustert und anhand ihrer durchschnittlichen Interaktionsenergie und der vergrabenen Interaktionsoberfläche sortiert [185].

### 4.1.4 Netzwerkanalysen und sequenzbasierte Methoden

In den vergangenen Jahren wurde viel an der Erstellung von Datenbanken gearbeitet, die experimentell bestimmte PPIs beinhalten. Dabei liegt ein großer Aufwand in der Validierung und der Sicherstellung der Reproduzierbarkeit von Experimenten bzw. der gewonnenen Informationen über PPIs [174, 187]. Es gibt eine Reihe solcher Datenbanken wie z. B. die BIND [188], BioGRID [36, 189], IntAct [190–192] und HPRD [193], die sich teilweise in der Größe und Vielfalt der PPIs signifikant unterscheiden.

Um herauszufinden wie und welche Proteine miteinander interagieren und wie diese z. B. mit Krankheiten in Verbindung gebracht werden können, werden PPINs aus den Datenbanken erstellt. In den PPIN stellen die Proteine Knoten und die Protein-Protein-Interaktionen die Kanten dar. Proteine mit mehreren Kanten bzw. Interaktionspartnern sind als zentraler Bestandteil einer biologischen Zelle identifizierbar und beeinflussen deren Funktion maßgeblich. Diese Relevanz kann ausgenutzt werden, um Proteine zu identifizieren und deren Aktivität und Interaktionen mit anderen Proteinen gezielt zu modifizieren [194–197]. In einer Studie [198] wurde beispielsweise ein PPIN mit 6 339 menschlichen Proteinen und 34 813 Interaktionen analysiert. Mit diesem konnten die Proteine in die Kategorien „unverzichtbar“, „neutral“ oder „entbehrlich“ unterteilt werden. Die Kategorien beschreiben, wie wichtig ein Protein im Netzwerk ist. Ein „unverzichtbares“ Protein stellt beispielsweise einen zentralen Knoten im Netzwerk dar. Das Entfernen eines solchen Knotens hat somit einen großen Einfluss auf die Funktion von anderen, mit ihm verbundenen Proteinknoten. Mithilfe des Netzwerks wurde festgestellt, dass 21 % der Proteine unverzichtbar sind und häufig primäre Ziele für human-pathogene Viren darstellen, Mutationen unterliegen und/oder durch Wirkstoffe in ihrer Aktivität beeinflusst werden können. Eine weitere Analyse an Krebspatienten zeigte außerdem, dass einige Gene, die bei verschiedenen Krebsarten häufig überexprimiert werden oder deren Expression inhibiert wird, ebenfalls unverzichtbar sind. Dies erlaubt die Identifikation neuer potenzieller Angriffspunkte für Krebsmedikamente [198].

In Fällen, in denen weder eine Proteinstruktur vorliegt noch Interaktionspartner bekannt sind, können sequenzbasierte Methoden eingesetzt werden. Diese sagen die Wahrscheinlichkeit einer Interaktion und die für eine Bindung relevanten Reste vorher. So können mithilfe des multiplen Sequenzalignments über den Abgleich mit Sequenzähnlichen Proteinen aus einer Datenbank Proteinspartner vorhergesagt werden. Die Annahme besteht darin, dass Reste, die für Interaktionen verantwortlich sind, in der Natur konserviert sind, wohingegen sich Reste, die keine Interaktionen eingehen, stärker verändern [173, 174, 199].

## 4.2 Bestimmung der Ähnlichkeit von Protein-Protein-Bindestellen

Die Erforschung der Gemeinsamkeiten von Bindestellen und Interaktionsmustern ist für viele der PPI *in silico*-Bestimmungsmethoden wie der Netzwerkanalyse oder der Identifizierung von Hotspots hilfreich und ermöglicht weitreichende Einblicke. So können durch die Ähnlichkeitsbestimmungen Beziehungen zwischen Proteininteraktionen analysiert, neue Interaktionen, Bindungspartner und -modi vorhergesagt und strukturell charakterisiert, sowie Proteinfunktionen abgeleitet werden [173, 200–202].

In der Vergangenheit wurden strukturbasierte Methoden zur Untersuchung von PPIs entwickelt, die auf verschiedenen Betrachtungsweisen von Ähnlichkeit beruhen [D6]. Eine Methode, die 2010 veröffentlicht wurde und die strukturelle und sequentielle Ähnlichkeit von Protein-Protein-Bindestellen berechnet, ist **iAlign** [203]. iAlign basiert auf der Methode TM-align [204], die eine Ähnlichkeitsbestimmung von Proteinstrukturen ermöglicht. In iAlign werden Protein-Protein-Bindestellen über die Nähe zweier Proteinketten in Dimeren bestimmt. Eine Aminosäure wird als Teil der Bindestelle deklariert, wenn mindestens ein Schweratom der anderen Kette innerhalb eines Abstands von 4,5 Å liegt. Die Bindestelle ist anschließend definiert als die Menge all dieser Aminosäuren.

Die Ähnlichkeit zweier Protein-Protein-Bindestellen wird in zwei Phasen berechnet. In der ersten Phase werden mögliche Überlagerungen bestimmt. In der zweiten Phase werden diese Überlagerungen angewandt und ihre Ähnlichkeit mithilfe von zwei Bewertungsfunktionen abgeschätzt.

Die erste Phase umfasst die Berechnung von vier Überlagerungen: 1. Ein *gapless sequence alignment* (deutsch: lückenlose Sequenzüberlagerung); 2. Eine Sekundärstrukturüberlagerung; 3. Eine Überlagerung basierend auf einer Fragmentierung der Bindestelle und anschließender Überlagerung aller Fragmentpaare, die sich mindestens ein Paar von sequentiell aufeinanderfolgenden Aminosäuren derselben Sekundärstruktur teilen; 4. Basierend auf den drei vorhergehenden Überlagerungen wird eine neue Bewertungsmatrix erstellt, durch die eine weitere Überlagerung generiert wird.

In der zweiten Phase werden die Überlagerungen bewertet und dazu verwendet, iterativ neue Bewertungsmatrizen und Überlagerungen zu generieren. Diese Wiederholung endet nach maximal 30 Iterationen oder sobald die Überlagerung konvergiert (das Konvergenzkriterium wird in der Publikation [203] nicht näher beschrieben).

Als Ähnlichkeitsmaße stehen in iAlign der TM- (*Template Modeling*) und IS-Score

(*Interface Similarity*) zur Verfügung. Beide bewerten die Ähnlichkeit auf Basis der Distanz zwischen den C $\alpha$ -Atomen der überlagerten Reste. Der IS-Score verwendet darüber hinaus einen sogenannten Kontaktüberlapp-Faktor. Dieser beschreibt die Erhaltung der Kontaktmuster, also der Interaktionen zwischen den beiden Proteinen.

Eine weitere Methode namens **I2I-SiteEngine** [205, 206] war sowohl als installierbare Anwendung als auch in Form eines Webservice verfügbar. Der zugrundeliegende Algorithmus wird in drei Phasen eingeteilt: 1. Repräsentation der Protein-Protein-Bindestelle; 2. Berechnung von Transformationen auf Grundlage eines Matching-Verfahrens zur Bestimmung von Überlagerungen; 3. Bewertung der Überlagerungen anhand von Bewertungsfunktionen.

Zur Bestimmung der Protein-Protein-Bindestelle wird das gleiche Verfahren wie bei iAlign verwendet, mit Ausnahme der Distanz zwischen zwei Interfaceresten von 4 Å anstelle von 4,5 Å. Anschließend werden für alle Reste der Bindestelle sogenannte Pseudozentren für ihre lösungsmittelzugänglichen funktionellen Gruppen berechnet. Pseudozentren sollen angeben, auf welche Weise Interaktionen mit den Resten der Bindestelle ausgebildet werden können. Die Typen der Pseudozentren werden analog zu Schmitt et al. [207] berechnet und sind definiert als Donor, Akzeptor, Donor/Akzeptor, aliphatisch oder aromatisch. Mehrere Pseudozentren des gleichen Typs werden als Patch bezeichnet, welcher durch das Pseudozentrum beschrieben wird, das am nächsten am Mittelpunkt des Patches liegt. Am Zentrum des Patches wird außerdem die durchschnittliche Krümmung des Patches mithilfe eines Raumwinkels [208] beschrieben.

Im Matching-Verfahren werden zuerst für die Protein-Protein-Bindestelle alle komplementären Pseudozentren der beiden Proteinketten bestimmt. Proteinbindestellen benötigen mindestens drei Pseudozentren. Es werden anschließend Triplets bzw. Dreiecke der Pseudozentren bestimmt. Hashwerte werden für die Dreiecke anhand der physikochemischen Eigenschaften, Krümmungen und Seitenlängen berechnet und in Hashwert-Tabellen gespeichert. Dieser Vorgang wird ebenfalls für die zu durchsuchenden Protein-Protein-Bindestellen durchgeführt. Über die Suche nach Hashwerten der Anfrage-Bindestelle in der Hashwert-Tabelle der Suchmenge können so ähnliche Dreiecke gefunden werden, deren Matching der jeweiligen Pseudozentren eine Überlagerung/Transformation ermöglicht.

Für die Bewertung der Überlagerungen anhand der berechneten Transformationen werden zwei Bewertungsfunktionen für die Bestimmung der Ähnlichkeit der dreidimensionalen Form und der physikochemischen Eigenschaften verwendet. Der *Low-Resolution*

*Score* ist schnell berechenbar, da er nur für die Patchzentren berechnet wird. Die Ergebnisse werden sortiert und anhand des RMSDs der transformierten Pseudozentren geclustert. Der zweite *Overall Surface Score* bewertet die Überlagerung der gesamten Oberfläche für Repräsentanten der Cluster. Für die Ergebnisse wird ein sogenannter *1:1 Correspondence Score* bestimmt. Abschließend werden alle Scores zu einem *Total Score* aufsummiert und bilden die endgültige Bewertung.

### 4.3 Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen mit PiMine

Aufgrund des steigenden Interesses an der Erforschung von PPIs in der Wirkstoffentwicklung stellte sich bereits zu Beginn dieses Promotionsprojekts die Frage, ob GeoMine um die Suche nach Mustern in Protein-Protein-Bindestellen erweitert werden könnte. In Rücksprache mit anderen Wissenschaftler:innen zeigte sich ein großes Interesse an der Identifizierung ähnlicher Protein-Bindestellen zu einer Anfrage-Bindestelle. Die Formulierung einer GeoMine-Anfrage auf den vergleichsweise meist deutlich größeren Protein-Protein-Bindestellen stellte sich allerdings als problematisch heraus. Stattdessen sollte eine automatische Generierung von Anfragen und anschließende Auswertung der Ergebnisse als benutzerfreundliche Alternative entwickelt werden. Bereits 2020 wurde in einer Masterarbeit [S2] gezeigt, dass GeoMine zur effizienten Bestimmung der Ähnlichkeit von Proteinbindetaschen verwendet werden kann. In dieser Arbeit wurde ein Algorithmus entwickelt, der automatisiert geometrische GeoMine-Anfragemuster in Form von Dreiecken erstellt. Jeder gefundene Treffer stellt eine lokale Ähnlichkeit der Bindetasche dar und wird bewertet. Der Algorithmus zeichnet sich durch eine geringe Laufzeit bei hoher Genauigkeit verglichen mit anderen Programmen zur Ähnlichkeitsbestimmung aus. Im Anschluss an diese Masterarbeit wurde der Algorithmus überarbeitet und in einer neuen Methode namens SiteMine [D7] zur Suche nach Bindetaschenähnlichkeiten integriert. Hierbei wurden die zweidimensionalen Dreiecksfilter durch dreidimensionale Tetraeder ersetzt. Die weitere Dimension ermöglicht eine Hinzunahme der räumlichen Tiefe und damit eine geeignetere Beschreibung der üblicherweise nicht planaren Bindetaschen. Basierend auf diesem sog. TetraScan-Algorithmus wurde anschließend die Methode PiMine im Rahmen dieses Promotionsprojekts entwickelt. Ein Augenmerk lag während der Umsetzung auf den Eingabemöglichkeiten des Programms. Etablierte Methoden wie iAlign und I2I-SiteEngine erlauben einzig die Ähnlichkeitsbestimmung auf Basis zweier Proteinketten, die eine PPI ausbilden. Die Möglichkeit für eine einzige Proteinkette mögliche Interaktionspartner zu bestimmen ist nicht gegeben. Insbesondere wenn kein

interagierendes Protein bekannt ist, kann diese Eingabemöglichkeit allerdings hilfreich sein. Aus diesem Grund ist dies in PiMine neben der herkömmlichen Eingabe zweier Proteinketten eine weitere Eingabevariante. Somit können auch mögliche Protein-Interaktionsbereiche auf Proteinoberflächen mit Programmen wie ISPRED4 [209] vorhergesagt und mithilfe von PiMine anschließend für eine Suche verwendet werden, um PPINs um strukturelle Vorhersagen zu erweitern oder weitere mögliche Interaktionspartner zu identifizieren.

Im Folgenden wird eine methodische Zusammenfassung von PiMine und der an Protein-Protein-Bindestellen angepassten Variante des TetraScan-Algorithmus gegeben. Anschließend folgt eine Evaluierung der Methode im Vergleich zu ähnlichen etablierten Verfahren, sowie ein Ausblick. Weitere Details sind in [D6] zu finden.

### 4.3.1 Methodische Zusammenfassung

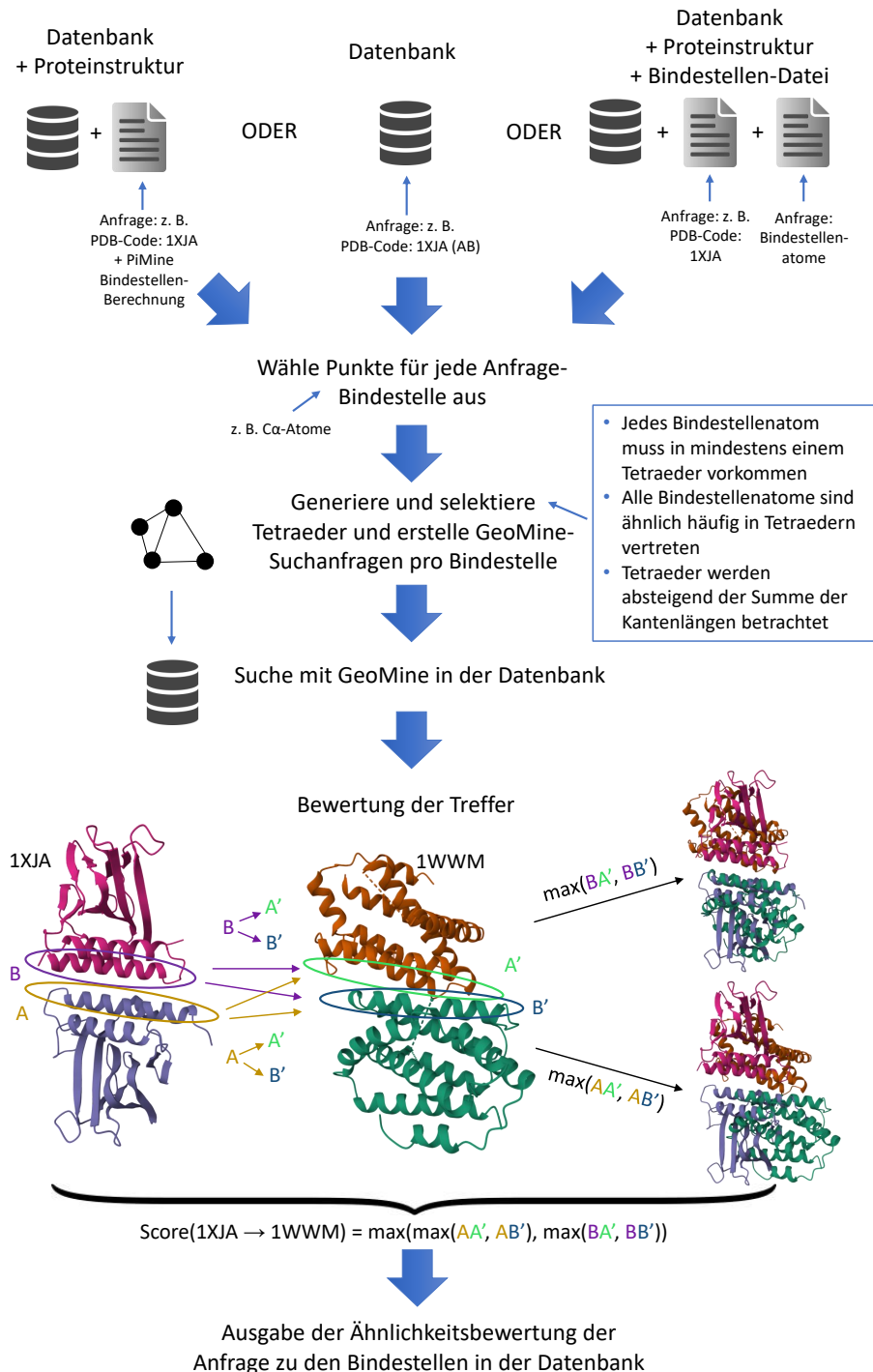
Der allgemeine methodische Ablauf von PiMine ist in Abbildung 4.2 dargestellt. Zu Anfang benötigt PiMine eine Datenbank, die die zu durchsuchenden Protein-Protein-Bindestellen beinhaltet. Diese Datenbank kann mithilfe von PiMine für Proteinstrukturen im PDB- und mmCIF-Format erstellt werden. Anschließend kann die Suche auf drei unterschiedliche Arten gestartet werden: 1. Die Nutzer:innen geben einen in der Datenbank enthaltene PDB-Code und die Bezeichner der Proteinketten der Protein-Protein-Bindestellen an. 2. Die Nutzer:innen übergeben eine Proteinstruktur im PDB- oder mmCIF-Format und die Bindestellen-Proteinketten-Bezeichner. Die Protein-Protein-Bindestellen werden anschließend in der übergebenen Struktur gesucht und prozessiert. 3. Neben einer Proteinstruktur im PDB- oder mmCIF-Format wird eine weitere Struktur im PDB- oder mmCIF-Format übergeben, die eine Protein-Bindestelle enthält.

Das Laden der Anfrage-Bindestelle durch Eingabevarianten 1 und 2 resultiert in jeweils zwei getrennten Protein-Bindestellen, die zusammen die Protein-Protein-Bindestelle darstellen. Die zwei Protein-Bindestellen werden von PiMine separat voneinander betrachtet. Dies basiert auf der Annahme, dass zwei Bindestellen ähnlich sind, wenn die Bindestellen ihrer jeweiligen Interaktionspartner ebenfalls ähnlich sind bzw. die Bindestellen beider Protein-Protein-Komplexe jeweils komplementär zueinander sind. Diese Annahme ist ebenfalls Begründung für die dritte Eingabevariante. Mit dieser wird nur eine einzige Protein-Bindestelle geladen, was beispielsweise zur Identifikation möglicher Interaktionspartner nützlich sein kann.

Sobald die Bindestellen geladen sind, werden Atome und Ringzentren wie in den Programmeinstellungen vorgegeben ausgewählt. Standardmäßig sind dies alle  $C\alpha$ -Atome.



### 4.3 Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen mit PiMine



**Abbildung 4.2:** Programmablauf der Protein-Protein-Bindestellen-Ähnlichkeitssuche mit PiMine, exemplarisch dargestellt anhand der Anfragestruktur mit PDB-Code 1XJA und der zu überprüfenden Struktur mit PDB-Code 1WWM. Der PPI-Score gibt die Ähnlichkeit der ähnlichsten aller möglichen Zuordnungen der Anfrage- und Zielbindestelle an. Die Moleküldarstellung wurde mithilfe von Mol\* [136] erstellt.

Alternativ können ebenfalls z. B. alle lösungsmittelzugänglichen Atome selektiert werden. Die Eigenschaften der ausgewählten Atome und Ringzentren werden abhängig von einem benutzerdefinierten Detailgrad in Suchpunkte übersetzt. So wird ein Suchpunkt beispielsweise durch den Interaktionstyp der zugehörigen Seitenkette oder mit höherem Detailgrad durch das chemische Element des zugrundeliegenden Atoms beschrieben. Für eine PiMine-Suchanfrage kommen anschließend nur solche Suchpunkte infrage, deren Distanz zu einem anderen Suchpunkt innerhalb einer Minimal- und Maximaldistanz von 1 Å bis 14 Å liegt. Auf diese Weise werden jeweils vier tetraedrisch angeordnete Punkte ausgewählt, miteinander verbunden und alle Kombinationen enumeriert. Um die Suche nicht auf die exakte Übereinstimmung dieser Tetraeder einzuschränken, werden für die Distanzen zwischen den Suchpunkten Toleranzen von 1 Å verwendet.

Nachdem alle Filter erstellt wurden, werden diese anhand der Summe der Distanzen absteigend sortiert. 30 Tetraeder werden ausgewählt und mithilfe des GeoMine-Suchalgorithmus in der Datenbank gesucht. Die Auswahl startet mit den größten Tetraedern. Die Tetraeder werden so ausgewählt, dass jedes Bindestellenatom möglichst mit ähnlicher Häufigkeit in den selektierten Tetraedern vertreten ist. Jeder resultierende Treffer der GeoMine-Suchen beschreibt eine tetraedrische Anordnung von Atomen in der Anfrage-Bindestelle, die ebenfalls in einer anderen Bindestelle vorkommt. Hierbei müssen die Suchpunkt-Eigenschaften übereinstimmen, während die Distanzen innerhalb der definierten Bereiche des jeweiligen Filters liegen müssen. Die vier Tetraeder-Punkte der Anfrage und des Treffers können einander zugeordnet werden. Mithilfe dieser Zuordnung kann eine Transformation bzw. Überlagerung der Bindestellen bestimmt und später eine Ähnlichkeit der gesamten Bindestellen bewertet werden. Hier ist zu beachten, dass eine Suchanfrage mehrere Treffer in der gleichen Bindestelle erzielen kann. Diese werden nicht gefiltert, da jeder Treffer potenziell eine andere Transformation beschreibt.

Nachdem die Suche abgeschlossen ist, beginnt eine zweistufige Bewertung der Treffer. Zuerst wird eine schnelle Vorfilterung durchgeführt, um die Treffermenge auf die vielversprechendsten einzugrenzen. Dafür werden die Bindestellen der Anfrage und des Treffers überlagert. Anhand der Übereinstimmung der räumlichen Lage der C $\alpha$ -Atome wird dann die Ähnlichkeit bewertet. Zu diesem Zweck wird innerhalb eines Radius von 6 Å jedes C $\alpha$ -Atoms der Anfrage darauf überprüft, ob auch ein C $\alpha$ -Atom in der Bindestelle des Treffers vorliegt. Die besten  $x$  Treffer pro Bindestelle werden ausgewählt, wobei  $x$  als quadratische Wurzel der Anzahl der Gesamttreffer der jeweiligen Bindestelle definiert ist. Nach dieser Vorfilterung werden alle verbleibenden Treffer verarbeitet und bewertet. Die Bewertung basiert auf einem Radius von 1,5 Å und berechnet neben

der Übereinstimmung der räumlichen Form auch die Ähnlichkeit der pharmakophoren Eigenschaften anhand einer Matrix (siehe [D6]).

Da in PiMine eine Protein-Protein-Interaktion in eine Protein-Bindestelle pro Proteinkette aufgeteilt wird, müssen deren Filter, die Suche und Bewertung zweimal erstellt bzw. durchgeführt werden. Die endgültige Bewertung der Ähnlichkeit zweier Protein-Protein-Bindestellen setzt sich dann aus vier Bewertungen zusammen. Seien beispielsweise A und B die Kettennamen der Anfrage-Interaktion und A' und B' die der Ziel-Interaktion, dann gibt es folgende Kombinationen: 1) A gegen A', 2) A gegen B', 3) B gegen A' und 4) B gegen B'. Die Kombinationen 1) und 2), sowie 3) und 4) werden separat voneinander prozessiert. Abschließend beschreibt das Maximum der vier Kombinationen die Ähnlichkeit der beiden Protein-Protein-Bindestellen zueinander.

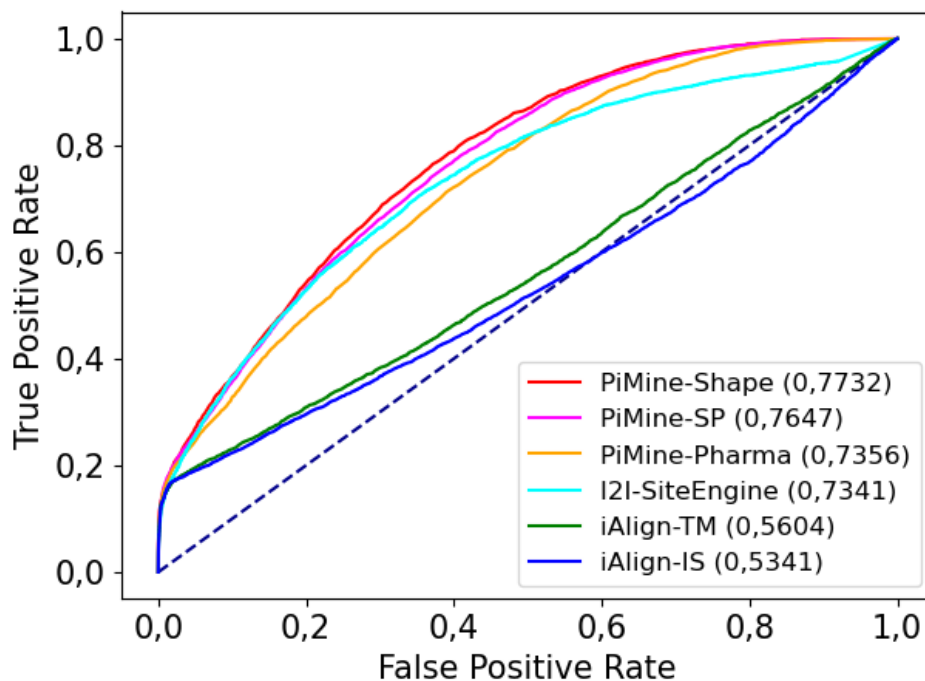
#### 4.3.2 Evaluation

Die Fähigkeit von PiMine ähnliche Protein-Protein-Bindestellen von weniger ähnlichen zu unterscheiden, wurde auf drei unterschiedlichen Datensätzen evaluiert. Bei diesen handelt es sich um zwei etablierte Datensätze von Gao et al. [203] und Keskin et al. [210], sowie einen für die Evaluation von PiMine neu entwickelten Datensatz [D6]. Die Datensätze unterscheiden sich maßgeblich in der Definition von Ähnlichkeit und Unähnlichkeit. Ein Vergleich wird zu zwei häufig zitierten Methoden erstellt. Bei diesen handelt es sich um iAlign [203] und I2I-SiteEngine [205, 206]. Anhand der drei Datensätze konnte mithilfe von Anreicherungsfaktoren (englisch: *enrichment factors*, EFs) und ROC-Kurven (ROC: *receiver operating characteristic*) gezeigt werden, dass PiMine anhand des Scores ähnliche Interfaces überzeugend von unähnlichen unterscheiden kann (siehe Tabelle 4.1). So konnte auf dem für die Evaluation von PiMine entwickelten Datensatz gezeigt werden, dass PiMine insbesondere Ähnlichkeiten für zwei Protein-Protein-Bindestellen A/B und A'/B' verlässlich bestimmen kann, wenn nur A und A' global strukturell sehr ähnlich sind, während B und B' strukturell unähnlich sind. Auf dem Datensatz von Gao et al. erzielt PiMine ebenfalls hohe Genauigkeiten in der Ähnlichkeitsbestimmung, ist allerdings weniger performant als iAlign und I2I-SiteEngine. Dies wird darauf zurückgeführt, dass der Datensatz größtenteils auf der Annahme basiert, dass eine fehlende globale Ähnlichkeit unähnliche Interfaces impliziert. PiMine findet jedoch auch Ähnlichkeiten zwischen global unähnlichen Proteinen, da ihre Interfaces ähnlich sind und gegebenenfalls falsch klassifiziert wurden. Anhand des Datensatzes von Keskin et al. [210] konnte eine hohe Genauigkeit festgestellt werden, wenn Daten ohne eindeutige sequentielle Ähnlichkeiten verwendet werden. Dies

**Tabelle 4.1:** Normalisierte Anreicherungsfaktoren (EF) unter Berücksichtigung der bestbewerteten 5,0 % der Datensatzpaare der drei Methoden iAlign, I2I-SiteEngine, und PiMine mit ihren jeweiligen Bewertungsfunktionen auf den Datensätzen von Keskin et al. [210], Gao et al. [203] und dem für die Evaluation von PiMine entwickeltem Datensatz *PiMineSet* [D6]. Letzterer enthält Protein-Protein-Bindestellen A/B und A'/B', wobei die Proteinketten A und A' ähnlich sind, während B und B' unähnlich sind. Für den *PiMineSet*-Datensatz werden zwei Analysen durchgeführt. Die erste beinhaltet alle Proteinketten (*Unspezifisch*), während in der zweiten Analyse nur die unähnlichen Ketten betrachtet werden (*Spezifisch*). Da die Verwendung von iAlign die Angabe zweier Proteinketten erfordert, kann die zweite Analyse mit dieser Anwendung nicht durchgeführt werden. Die Werte für den *PiMineSet*-Datensatz und den Datensatz von Keskin et al. wurden entnommen aus [D6]. Werte für den Datensatz von Gao et al. wurden berechnet und ergänzt.

Methode	Score	Keskin et al.	Gao et al.	<i>PiMineSet</i> (Unspezifisch)	<i>PiMineSet</i> (Spezifisch)
iAlign	TM	0,19	1,00	0,86	-
	IS	0,19	1,00	0,73	-
I2I-SiteEngine	Total	0,24	0,93	0,52	0,47
PiMine	SP	0,25	0,79	0,87	0,68
	Pharma	0,24	0,80	0,88	0,68
	Shape	0,26	0,73	0,86	0,64

wird ebenfalls in Abbildung 4.3 mithilfe von ROC-Kurven gezeigt. Die ROC-Kurven stellen das Verhältnis von *true positive rate* (deutsch: Richtig-Positiv-Rate) und *false positive rate* (deutsch: Falsch-Positiv-Rate) dar bzw. in welchem Maße eine Methode in der Lage ist, anhand des Scores ähnliche und unähnliche Bindestellen im gesamten Datensatz zu unterscheiden. Die drei Aufflistungen von PiMine in der Legende entsprechen unterschiedlichen Bewertungsmaßen (siehe Kapitel 4.3.1). PiMine-Shape stellt die Übereinstimmung der Form dar, PiMine-Pharma die Übereinstimmung der pharmakophoren Eigenschaften und PiMine-SP die Summe beider. Die Werte in der Legende geben die Fläche unter den Kurven (englisch: *area under the ROC curve*, AUC) an. Eine AUC = 1 beschreibt eine perfekte Unterscheidung von ähnlichen und unähnlichen Interfaces. Auf dem Datensatz von Keskin et al. erzielt PiMine die höchsten AUC-Werte. Die Ergebnisse von I2I-SiteEngine sind auf diesem Datensatz ebenfalls gut und vergleichbar mit denen von PiMine unter der Verwendung des PiMine-Pharma Scores. iAlign erreicht hingegen die geringsten AUC-Werte. Da in der Praxis in der Regel nur die höchsten Bewertungen bzw. ein kurzer Bereich am Anfang der ROC-Kurve angeschaut wird, wurden außerdem die Anreicherungsfaktoren für unterschiedlich lange Bereiche bestimmt. Hier liefert PiMine weiterhin die besten Ergebnisse, während iAlign und I2I-SiteEngine vergleichbar performant sind [D6].



**Abbildung 4.3:** ROC-Kurven für die Vorhersage ähnlicher Protein-Protein-Bindestellen mithilfe der drei Methoden iAlign, I2I-SiteEngine und PiMine auf dem Datensatz von Keskin et al. [210]. Die Abbildung wurde aus [D6] entnommen.

In weiteren Analysen wurde die Laufzeit von PiMine im Vergleich zu I2I-SiteEngine und iAlign evaluiert. Dafür wurde ein Datensatz von 169 944 Protein-Protein-Bindestellen verwendet. Aus dem Datensatz wurde eine Protein-Protein-Bindestelle zufällig ausgewählt und die Ähnlichkeit zu allen anderen ermittelt. Die Berechnungsdauer dieses Prozesses wurde auf einem PC mit einem Intel i5-9500 (3,0 GHz) Prozessor, 32 GB Arbeitsspeicher, sowie eines Toshiba SSD-Laufwerks (512 GB, Modell NVMe) und eines Hitachi HDD-Laufwerks (2 TB) gemessen. Dabei wurden alle drei Programme jeweils getrennt auf der SSD und der HDD ausgeführt. Für PiMine wurden zwei Parametersätze getestet, einer, der für schnelle Berechnungen optimiert wurde und die Standardeinstellung von PiMine darstellt, und einer, der genauere Ähnlichkeitsbewertungen ermöglicht. Die Ergebnisse der Analysen in Tabelle 4.2 zeigen, dass iAlign mit Abstand die schnellste der drei Methoden ist und insgesamt drei bis vier Stunden benötigt. PiMine ist im Vergleich deutlich langsamer. Auf der SSD mit Laufzeit-optimierten Parametern wird eine Laufzeit von etwa 19 Stunden bzw. eine etwa 5,6x längere Laufzeit als bei iAlign bestimmt. Die Laufzeiten von PiMine sind bei Wahl der Laufzeit-optimierten Parameter etwa halb so hoch als unter Verwendung der auf Performanz optimierten

**Tabelle 4.2:** Laufzeiten von PiMine, iAlign, und I2I-SiteEngine. PiMine’s Laufzeit wird mit Laufzeit- und Genauigkeits-optimierten Parametern gemessen. Die verwendete PostgreSQL Datenbank ist sowohl auf einem HDD-Laufwerk als auch einem SSD-Laufwerk initialisiert. Als Datensatz wurden 169 944 Protein-Protein-Komplexe verwendet. Die Laufzeiten wurden für die Berechnung der Ähnlichkeiten einer zufällig ausgewählten Protein-Protein-Bindestelle zwischen den Ketten A und B des PDB Eintrags 3T4M bestimmt. Die Tabelle wurde aus [D6] entnommen und übersetzt.

Methode	Laufwerk	Laufzeit [Stunden]
PiMine - Laufzeitoptimierung	HDD	41,0
PiMine - Genauigkeitsoptimierung	HDD	87,7
iAlign	HDD	3,9
I2I-SiteEngine	HDD	371,6
PiMine - Laufzeitoptimierung	SSD	19,1
PiMine - Genauigkeitsoptimierung	SSD	39,0
iAlign	SSD	3,4
I2I-SiteEngine	SSD	369,6

Parameter. Auch die Wahl der Laufwerktechnologie hat einen entscheidenden Einfluss auf die Laufzeit von PiMine. Die Verwendung einer SSD ermöglicht eine etwa 54 % geringere Laufzeit. I2I-SiteEngine braucht von allen drei Methoden am längsten und benötigt etwa 370 bis 372 Stunden. Die Wahl der Laufwerktechnologie hat mit weniger als einem Prozent kaum einen Einfluss. In einer weiteren Analyse wurde die Genauigkeit der berechneten Überlagerungen ähnlicher Bindestellen bestimmt. Hierfür wurden die ähnlichen Paare des neu entwickelten Datensatzes *PiMineSet* [D6] verwendet. Dieser beinhaltet Protein-Protein-Interaktionen für die folgendes gilt: Seien A und B, sowie A' und B' interagierende Ketten, dann ist Proteinkette A mit Proteinkette A' verwandt, während die Proteinketten B und B' mit keiner Kette verwandt und global unähnlich sind. Es ist dennoch anzunehmen, dass die Bindestellen der beiden Ketten Ähnlichkeiten aufweisen, da sie mit global sehr ähnlichen Ketten in derselben Region interagieren. Für die jeweils verwandten Proteinketten wurden Überlagerungen mithilfe des Programms TM-align [204] berechnet und als „korrekte“ Überlagerungen betrachtet. Zur Einschätzung der Genauigkeit der Überlagerungen von PiMine, iAlign und I2I-SiteEngine wurden anschließend die RMSD Werte zwischen den nicht verwandten Proteinketten berechnet. Die Ergebnisse zeigen, dass iAlign im Median eine Abweichung der berechneten zur korrekten Überlagerung von 1,32 Å bzw. 1,23 Å für den IS-Score bzw. den TM-Score aufweist. PiMine zeigt eine ähnlich gute Leistung mit einem Median von 1,43 Å während I2I-SiteEngine mit 2,57 Å deutlich ungenauer ist.

### 4.3.3 Ausblick

Wir konnten zeigen, dass PiMine gute Ergebnisse für die Ähnlichkeitsanalyse von Protein-Protein-Bindestellen erzielt. Mithilfe von randomisiert generierten kleinen Änderungen der Rotation und/oder Translation könnte dennoch eine Verbesserung der Überlagerung erreicht werden. Derzeit toleriert die Mustersuche durch GeoMine eine Abweichung der Distanzen zwischen den Suchpunkten beim Auffinden ähnlicher Muster und überlagert diese mit der Anfrage. Eine kleine Änderung dieser Überlagerung würde die lokale Ähnlichkeit der Überlagerung des Musters verschlechtern, aber möglicherweise die Gesamtähnlichkeit der Protein-Protein-Bindestellen verbessern. Dies ließe sich voraussichtlich mit einem Clustering der Überlagerungstransformationen kombinieren. Einer der größten Zeitfaktoren des Algorithmus ist das Anwenden der Transformationen auf die Bindestelle während der Bewertungsphase. Durch ein Clustering der Transformationen könnte die Anzahl auf einige Cluster-Repräsentanten reduziert werden und anschließend die Randomisierung stattfinden. Eine Vorbedingung für ein Clustering wäre eine Analyse, wie groß der Einfluss auf den Score ist, wenn anstelle der passendsten Transformation nur noch ein Repräsentant des gleichen Clusters verwendet wird.

Die Erstellung einer Webanwendung für PiMine würde die Nutzung erleichtern, da keine Datenbanken heruntergeladen oder durch Nutzer:innen erstellt werden müssten. Auch wäre eine Installation des Programms nicht notwendig. Mit dem aktuellen Stand wäre eine Webanwendung allerdings nur für kleine Datensätze nutzbar, da die Laufzeiten insbesondere für eine große Anzahl von PPIs hoch sind. Nutzer würden voraussichtlich das Interesse an der Anwendung verlieren, wenn sie nicht innerhalb kurzer Zeit ein Ergebnis erhalten. Eine Optimierung der Laufzeit ist deshalb vonnöten. Hierfür könnte zunächst evaluiert werden, ob es Größenunterschiede zwischen den zu vergleichenden PPIs gibt. Eine Reduzierung der Suche auf beispielsweise solche PPIs, die mindestens halb und maximal doppelt so groß wie die Anfrage sind, könnte die Laufzeit bereits verbessern. Auch könnte eine benutzerdefinierte Untergrenze für den Score eingebaut werden, sodass iterativ während der Suche PPIs nicht mehr weiter auf Ähnlichkeiten durchsucht werden, die bereits einen höheren Score erzielt haben.

Ein Alleinstellungsmerkmal von PiMine ist die Suche von Ähnlichkeiten einer einzelnen Protein-Bindestelle zu einer oder mehreren bekannten Protein-Protein-Bindestellen. Mit dieser Suche können beispielsweise mögliche Interaktionspartner für die Protein-Bindestelle vorhergesagt werden. Die Suche nach einer Protein-Bindestelle wurde entwickelt, da Proteine Bindestellen aufweisen können, für die bisher kein Protein-Protein-Komplex experimentell modelliert wurde. Wie oben in Kapitel 4.1 beschrieben, gibt

es deshalb Methoden zur Vorhersage von Protein-Bindestellen. Bisher wird die PiMine Datenbank für Protein-Protein-Bindestellen aufgebaut. Eine Vorhersage von einzelnen Bindestellen findet nicht statt. Durch die Implementierung oder das Verwenden von existierenden Vorhersagemethoden könnte die PiMine Datenbank deutlich vergrößert werden. Dies hätte den Vorteil, dass Ähnlichkeiten bestimmt werden könnten, auch wenn es kein Protein-Protein-Strukturmodell gibt. Somit könnte z. B. die Funktion eines Proteins anhand einer bekannten PPI bestimmt werden.



## Kapitel 5

# Zusammenfassung

Die in dieser Dissertation präsentierten Verfahren wurden für die geometrische Suche in großen Proteinstruktursammlungen wie der PDB entwickelt bzw. reimplementiert, überarbeitet und erweitert. Sie umfassen die Detektion, Analyse und Durchmusterung von Bindestellen kleiner Moleküle sowie die Ähnlichkeitsbestimmung von Protein-Protein-Bindestellen.

Jede der vorgestellten Methoden bietet neuartige Funktionalitäten für ihren intendierten Anwendungsbereich. Die Überarbeitung und Weiterentwicklung des DoGSite-Algorithmus erlaubt erstmals die Berechnung von schwer zu detektierenden Bindetaschen auf Basis gebundener Liganden in der Proteinstruktur. Mit einer eindeutigen Gitterorientierung werden unabhängig von der Proteinorientierung dieselben Taschen bestimmt. Auch konnte die Laufzeit um etwa das Zehnfache reduziert, die Stabilität der berechneten Taschenskriptoren erhöht und die Vorhersagequalität auf mehreren etablierten Datensätzen deutlich gesteigert werden. Mit GeoMine wurde auf Grundlage der PELIKAN-Applikation eine geometrische Suche in Proteinbindestellen entwickelt. Sie ist speziell auf äußerst diverse Suchmöglichkeiten, kurze Anfragedauer, multiple gleichzeitige Methodenzugriffe und die Verwendung von ligandbasierten und vorhergesagten Bindetaschen ausgerichtet. Letzteres wird durch die Integration von DoGSite3 ermöglicht. Durch die serverbasierte PostgreSQL-Datenbanktechnologie können mehrere Suchen schnell und parallel ausgeführt werden. Eine flexible atombasierte Definition von Suchpunkten und anpassbaren Toleranzen zwischen diesen Punkten erlauben präzise wie auch unspezifischere Formulierungen von Abfragemustern. Durch eine Vielzahl textueller und numerischer Eigenschaften von Liganden, Taschen, Proteinen und Komplexen kann der Suchraum eingeschränkt und die Laufzeit verringert werden. Auf Grundlage

der Suchmöglichkeiten in GeoMine wird mit PiMine eine Ähnlichkeitsbestimmung in Protein-Protein-Bindestellen realisiert. Diese Anwendung sucht pro Protein-Bindestelle mithilfe automatisch generierter geometrischer Muster. Die einzigartige Fähigkeit, eine einzelne Protein-Bindestelle ohne eine weitere, mit ihr interagierende Bindestelle, zu definieren, ermöglicht es beispielsweise, mit PiMine potenzielle Interaktionspartner von Proteinketten vorherzusagen.

In der Gesamtheit wird mit den vorgestellten Verfahren eine Möglichkeit zur schnellen und vielfältigen geometrischen und chemischen Suche in Proteinstruktursammlungen geschaffen. Die Integration des DoGSite-Algorithmus in GeoMine erlaubt eine deutliche Vergrößerung des Suchraums durch die Vorhersage möglicher biologisch relevanter Bindestellen. Mit dieser können z. B. bisher unbekannte potenzielle Bindetaschen für Liganden identifiziert oder die Proteinfunktion ermittelt werden. Anhand ligandengebundener Bindetaschen können vor der eigentlichen Detektion Bereiche des verwendeten Gitters klassifiziert werden, sodass diese Bereiche zur Bindetaschenvorhersage zusätzlich genutzt werden können. Diese Taschen können damit eine realitätsnähere Abgrenzung der Bindestellen von nicht-bindenden Regionen der Proteinoberfläche beschreiben, als es über eine Ligandenradius-basierte Taschendefinition möglich gewesen wäre. Die berechneten Taschendeskriptoren sowie das Taschenvolumen erweitern zudem die Suchmöglichkeiten in GeoMine, um den Suchraum weiter eingrenzen und Taschen mit bestimmten geometrischen und physikochemischen Eigenschaften effizient finden und analysieren zu können.

Durch die Verwendung des Suchverfahrens PiMine können außerdem Protein-Protein-Bindestellen miteinander verglichen und auf Ähnlichkeiten überprüft werden. Durch die automatisierte Generierung und Durchführung von Suchanfragen ist die Suche nutzerfreundlich und nicht auf exakte Treffer aller Suchpunkte beschränkt. Mithilfe der optimierten und entwickelten Methoden ergeben sich unzählige Möglichkeiten zur Erforschung von Interaktionen und Bindestellen im Allgemeinen und ermöglichen Forschenden im Bereich des strukturbasierten Designs zahlreiche Anwendungsmöglichkeiten. Im Hinblick auf die steigende Anzahl verfügbarer Proteinstrukturen und das steigende Interesse an Protein-Protein-Bindestellen stellen die entwickelten Verfahren hilfreiche Werkzeuge für den Wirkstoffentwurf dar.

# Literaturverzeichnis

- [1] Munos, B. Lessons from 60 years of pharmaceutical innovation. *Nat. Rev. Drug Discovery* 8.12 (2009), S. 959–968. DOI: 10.1038/nrd2961.
- [2] Medina-Franco, J. L. Grand Challenges of Computer-Aided Drug Design: The Road Ahead. *Front. Drug Discov.* 1 (2021). DOI: 10.3389/fddsv.2021.728551.
- [3] Rarey, M., Degen, J., Reulecke, I., „Docking and Scoring for Structure-based Drug Design“. In: *Bioinformatics-From Genomes to Therapies*. John Wiley & Sons, Ltd, 2007. Kap. 16, S. 541–599. ISBN: 9783527619368. DOI: 10.1002/9783527619368.ch16.
- [4] Berg, J. M., Tymoczko, J. L., Stryer, L., *Biochemistry*. 7th ed., international ed. New York, N.Y: W.H. Freeman und Company New York, N.Y, 2012. ISBN: 9781429276351.
- [5] Chothia, C., Lesk, A. M., The relation between the divergence of sequence and structure in proteins. *EMBO J.* 5.4 (1986), S. 823–826. DOI: 10.1002/j.1460-2075.1986.tb04288.x.
- [6] Sanger, F., Tuppy, H., The amino-acid sequence in the phenylalanyl chain of insulin. 1. The identification of lower peptides from partial hydrolysates. *Biochem. J.* 49.4 (1951), S. 463–481. DOI: 10.1042/bj0490463.
- [7] Sanger, F., Tuppy, H., The amino-acid sequence in the phenylalanyl chain of insulin. 2. The investigation of peptides from enzymic hydrolysates. *Biochem. J.* 49.4 (1951), S. 481–490. DOI: 10.1042/bj0490481.
- [8] Sanger, F., Thompson, E. O. P., The amino-acid sequence in the glycyl chain of insulin. 1. The identification of lower peptides from partial hydrolysates. *Biochem. J.* 53.3 (1953), S. 353–366. DOI: 10.1042/bj0530353.

- [9] Sanger, F., Thompson, E. O. P., The amino-acid sequence in the glyceryl chain of insulin. 2. The investigation of peptides from enzymic hydrolysates. *Biochem. J.* 53.3 (1953), S. 366–374. DOI: 10.1042/bj0530366.
- [10] Ryle, A. P., Sanger, F., Smith, L. F., Kitai, R., The disulphide bonds of insulin. *Biochem. J.* 60.4 (1955), S. 541–556. DOI: 10.1042/bj0600541.
- [11] Wooley, J., Ye, Y., „A Historical Perspective and Overview of Protein Structure Prediction“. In: 2007, S. 1–43. ISBN: 978-0-387-33319-9. DOI: 10.1007/978-0-387-68372-0\_1.
- [12] Anfinsen, C. B., Redfield, R. R., Choate, W. L., Page, J., Carroll, W. R., Studies on the gross structure, cross-linkages, and terminal sequences in ribonuclease. *J. Biol. Chem.* 207.1 (1954), S. 201–210. DOI: 10.1016/S0021-9258(18)71260-X.
- [13] Kendrew, J. C., Bodo, G., Dintzis, H. M., Parrish, R. G., Wyckoff, H., Phillips, D. C., A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. *Nature* 181.4610 (1958), S. 662–666. DOI: 10.1038/181662a0.
- [14] Smyth, M. S., Martin, J. H., x ray crystallography. *Mol. Pathol.* 53.1 (2000), S. 8–14. DOI: 10.1136/mp.53.1.8.
- [15] Deschamps, J. R. X-ray crystallography of chemical compounds. *Life Sci.* 86.15 (2010), S. 585–589. DOI: 10.1016/j.lfs.2009.02.028.
- [16] Webber, J., Strange, J., Dore, J., An evaluation of NMR cryoporometry, density measurement and neutron scattering methods of pore characterisation. *Magn. Reson. Imaging* 19.3 (2001), S. 395–399. DOI: 10.1016/S0730-725X(01)00255-7.
- [17] Webber, J. B. W., Dore, J. C., Neutron Diffraction Cryoporometry—A measurement technique for studying mesoporous materials and the phases of contained liquids and their crystalline forms. *Nucl. Instrum. Methods Phys. Res., Sect. A* 586.2 (2008), S. 356–366. DOI: 10.1016/j.nima.2007.12.004.
- [18] Kono, F., Kurihara, K., Tamada, T., Current status of neutron crystallography in structural biology. *Biophys. Physicobiol.* 19 (2022), e190009. DOI: 10.2142/biophysico.bppb-v19.0009.
- [19] Callaway, E. The revolution will not be crystallized: a new method sweeps through structural biology. *Nature* 525.7568 (2015), S. 172–174. DOI: 10.1038/525172a.
- [20] Jumper, J. Highly accurate protein structure prediction with AlphaFold. *Nature* 596.7873 (2021), S. 583–589. DOI: 10.1038/s41586-021-03819-2.

- [21] Varadi, M. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res.* 50.D1 (2021), S. D439–D444. DOI: 10.1093/nar/gkab1061.
- [22] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., Bourne, P. E., The Protein Data Bank. *Nucleic Acids Res.* 28.1 (2000), S. 235–242. DOI: 10.1093/nar/28.1.235.
- [23] Laskowski, R. A. Protein Structure Databases. *Mol. Biotechnol.* 48.2 (2011), S. 183–198. DOI: 10.1007/s12033-010-9372-4.
- [24] Suplatov, D. A., Švedas, V. K., Study of Functional and Allosteric Sites in Protein Superfamilies. *Acta Naturae* 7.4 (2015), S. 34–45. DOI: 10.32607/20758251-2015-7-4-34-54.
- [25] Konc, J., Janežič, D., Protein binding sites for drug design. *Biophys. Rev.* 14.6 (2022), S. 1413–1421. DOI: 10.1007/s12551-022-01028-3.
- [26] Thangudu, R. R., Tyagi, M., Shoemaker, B. A., Bryant, S. H., Panchenko, A. R., Madej, T., Knowledge-based annotation of small molecule binding sites in proteins. *BMC Bioinf.* 11.1 (2010), S. 365. DOI: 10.1186/1471-2105-11-365.
- [27] PDB, R. *PDB Statistics*. (Zugriff am 15.02.2022). URL: <https://www.rcsb.org/stats/>.
- [28] Ambrogelly, A., Palioura, S., Söll, D., Natural expansion of the genetic code. *Nat. Chem. Biol.* 3.1 (2007), S. 29–35. DOI: 10.1038/nchembio847.
- [29] Fischer, E. Einfluss der Configuration auf die Wirkung der Enzyme. *Ber. Dtsch. Chem. Ges.* 27.3 (1894), S. 2985–2993. DOI: 10.1002/cber.18940270364.
- [30] Koshland, D. E. Application of a Theory of Enzyme Specificity to Protein Synthesis. *Proc. Natl. Acad. Sci. U. S. A.* 44.2 (1958), S. 98–104. DOI: 10.1073/pnas.44.2.98.
- [31] Pauling, L. *General Chemistry*. New York: Dover, 1988, S. 346. ISBN: 0-486-65622-5.
- [32] Bronowska, A. K., „Thermodynamics of Ligand-Protein Interactions: Implications for Molecular Design“. In: *Thermodynamics*. Rijeka: IntechOpen, 2011. Kap. 1. DOI: 10.5772/19447.
- [33] Bissantz, C., Kuhn, B., Stahl, M., Corrections to A Medicinal Chemist’s Guide to Molecular Interactions. *J. Med. Chem.* 53.16 (2010), S. 6241–6241. DOI: 10.1021/jm100950p.

- [34] Groom, C. R., Bruno, I. J., Lightfoot, M. P., Ward, S. C., The Cambridge Structural Database. *Acta Crystallogr., Sect. B: Struct. Sci., Cryst. Eng. Mater.* 72.2 (2016), S. 171–179. DOI: 10.1107/S2052520616003954.
- [35] Desaphy, J., Bret, G., Rognan, D., Kellenberger, E., sc-PDB: a 3D-database of ligandable binding sites—10 years on. *Nucleic Acids Res.* 43.D1 (2014), S. D399–D404. DOI: 10.1093/nar/gku928.
- [36] Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M., BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.* 34.Database issue (2006), S. D535–9. DOI: 10.1093/nar/gkj109.
- [37] Niedzialkowska, E., Gasiorowska, O., Handing, K. B., Majorek, K. A., Porebski, P. J., Shabalin, I. G., Zasadzinska, E., Cymborowski, M., Minor, W., Protein purification and crystallization artifacts: The tale usually not told. *Protein Sci.* 25.3 (2016), S. 720–733. DOI: 10.1002/pro.2861.
- [38] Bietz, S., Urbaczek, S., Schulz, B., Rarey, M., Protoss: a holistic approach to predict tautomers and protonation states in protein-ligand complexes. *J. Cheminf.* 6.1 (2014), S. 12. DOI: 10.1186/1758-2946-6-12.
- [39] Anandakrishnan, R., Aguilar, B., Onufriev, A. V., H++ 3.0: automating pK prediction and the preparation of biomolecular structures for atomistic molecular modeling and simulations. *Nucleic Acids Res.* 40.W1 (2012), W537–W541. DOI: 10.1093/nar/gks375.
- [40] Jiménez-García, B., Elez, K., Koukos, P. I., Bonvin, A. M., Vangone, A., PRODIGY-crystal: a web-tool for classification of biological interfaces in protein complexes. *Bioinformatics* 35.22 (2019), S. 4821–4823. DOI: 10.1093/bioinformatics/btz437.
- [41] Nittinger, E. „Water Molecules Within the HYDE Scoring Function: Placement, Optimization, and Energetic Contributions“. Diss. Universität Hamburg, 2018, S. 62–63.
- [42] Holm, L. Using Dali for protein structure comparison. *Methods Mol. Biol.* 2112 (2020), S. 29–42. DOI: 10.1007/978-1-0716-0270-6\_3.
- [43] Shindyalov, I. N., Bourne, P. E., Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng., Des. Sel.* 11.9 (1998), S. 739–747. DOI: 10.1093/protein/11.9.739.

- [44] Levitt, M., Gerstein, M., A unified statistical framework for sequence comparison and structure comparison. *Proc. Natl. Acad. Sci. U. S. A.* 95.11 (1998), S. 5913–5920. DOI: 10.1073/pnas.95.11.5913.
- [45] Krissinel, E., Henrick, K., Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr., Sect. D: Struct. Biol.* 60.12 Part 1 (2004), S. 2256–2268. DOI: 10.1107/S0907444904026460.
- [46] Gibrat, J.-F., Madej, T., Bryant, S. H., Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.* 6.3 (1996), S. 377–385. DOI: 10.1016/S0959-440X(96)80058-3.
- [47] Mavridis, L., Ritchie, D. W., „3D-BLAST: 3D Protein Structure Alignment, Comparison, and Classification Using Spherical Polar Fourier Correlations“. In: *Biocomputing 2010*, S. 281–292. DOI: 10.1142/9789814295291\_0030.
- [48] Keskin, O., Nussinov, R., Favorable scaffolds: proteins with different sequence, structure and function may associate in similar ways. *Protein Eng., Des. Sel.* 18.1 (2005), S. 11–24. DOI: 10.1093/protein/gzh095.
- [49] Rajamani, D., Thiel, S., Vajda, S., Camacho, C. J., Anchor residues in protein–protein interactions. *Proc. Natl. Acad. Sci. U. S. A.* 101.31 (2004), S. 11287–11292. DOI: 10.1073/pnas.0401942101.
- [50] Lichtarge, O., Bourne, H. R., Cohen, F. E., An Evolutionary Trace Method Defines Binding Surfaces Common to Protein Families. *J. Mol. Biol.* 257.2 (1996), S. 342–358. DOI: 10.1006/jmbi.1996.0167.
- [51] Choi, Y. S., Yang, J.-S., Choi, Y., Ryu, S. H., Kim, S., Evolutionary conservation in multiple faces of protein interaction. *Proteins: Struct., Funct., Bioinf.* 77.1 (2009), S. 14–25. DOI: 10.1002/prot.22410.
- [52] Inhester, T., Bietz, S., Hilbig, M., Schmidt, R., Rarey, M., Index-Based Searching of Interaction Patterns in Large Collections of Protein–Ligand Interfaces. *J. Chem. Inf. Model.* 57.2 (2017), S. 148–158. DOI: 10.1021/acs.jcim.6b00561.
- [53] Miranda, J. L. Position-dependent interactions between cysteine residues and the helix dipole. *Protein Sci.* 12.1 (2003), S. 73–81. DOI: 10.1110/ps.0224203.
- [54] Daylight Chemical Information Systems, I. *Daylight Theory Manual*. (Zugriff am 27.01.2021). URL: <http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html>.

- [55] Urbaczek, S., Kolodzik, A., Fischer, J. R., Lippert, T., Heuser, S., Groth, I., Schulz-Gasch, T., Rarey, M., NAOMI: On the Almost Trivial Task of Reading Molecules from Different File formats. *J. Chem. Inf. Model.* 51.12 (2011), S. 3199–3207. DOI: 10.1021/ci200324e.
- [56] Urbaczek, S., Kolodzik, A., Groth, I., Heuser, S., Rarey, M., Reading PDB: Perception of Molecules from 3D Atomic Coordinates. *J. Chem. Inf. Model.* 53.1 (2013), S. 76–87. DOI: 10.1021/ci300358c.
- [57] Urbaczek, S., Kolodzik, A., Rarey, M., The Valence State Combination Model: A Generic Framework for Handling Tautomers and Protonation States. *J. Chem. Inf. Model.* 54.3 (2014), S. 756–766. DOI: 10.1021/ci400724v.
- [58] Kolodzik, A., Urbaczek, S., Rarey, M., Unique Ring Families: A Chemically Meaningful Description of Molecular Ring Topologies. *J. Chem. Inf. Model.* 52.8 (2012), S. 2013–2021. DOI: 10.1021/ci200629w.
- [59] Mills, C. L., Beuning, P. J., Ondrechen, M. J., Biochemical functional predictions for protein structures of unknown or uncertain function. *Comput. Struct. Biotechnol. J.* 13 (2015), S. 182–191. DOI: 10.1016/j.csbj.2015.02.003.
- [60] Macari, G., Toti, D., Polticelli, F., Computational methods and tools for binding site recognition between proteins and small molecules: from classical geometrical approaches to modern machine learning strategies. *J. Comput.-Aided Mol. Des.* 33.10 (2019), S. 887–903. DOI: 10.1007/s10822-019-00235-7.
- [61] Illergård, K., Ardell, D. H., Elofsson, A., Structure is three to ten times more conserved than sequence—A study of structural response in protein cores. *Proteins: Struct., Funct., Bioinf.* 77.3 (2009), S. 499–508. DOI: 10.1002/prot.22458.
- [62] Waterhouse, A. SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Res.* 46.W1 (2018), W296–W303. DOI: 10.1093/nar/gky427.
- [63] Edelsbrunner, H., Kirkpatrick, D., Seidel, R., On the shape of a set of points in the plane. *IEEE Trans. Inf. Theory* 29.4 (1983), S. 551–559. DOI: 10.1109/TIT.1983.1056714.
- [64] Yang, L.-W., Bahar, I., Coupling between Catalytic Site and Collective Dynamics: A Requirement for Mechanochemical Activity of Enzymes. *Structure* 13.6 (2005), S. 893–904. DOI: 10.1016/j.str.2005.03.015.



- [65] Weisel, M., Proschak, E., Schneider, G., PocketPicker: analysis of ligand binding-sites with shape descriptors. *Chem. Cent. J.* 1.1 (2007), S. 7. DOI: 10.1186/1752-153X-1-7.
- [66] Yu, J., Zhou, Y., Tanaka, I., Yao, M., Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics* 26.1 (2009), S. 46–52. DOI: 10.1093/bioinformatics/btp599.
- [67] Hendlich, M., Rippmann, F., Barnickel, G., LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *J. Mol. Graphics Modell.* 15.6 (1997), S. 359–363. DOI: 10.1016/S1093-3263(98)00002-3.
- [68] Delaunay, B. Sur la sphère vide. Französisch. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na* 1934.6 (1934), S. 793–800.
- [69] Le Guilloux, V., Schmidtke, P., Tuffery, P., Fpocket: An open source platform for ligand pocket detection. *BMC Bioinf.* 10.1 (2009), S. 168. DOI: 10.1186/1471-2105-10-168.
- [70] Liang, J., Woodward, C., Edelsbrunner, H., Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Protein Sci.* 7.9 (1998), S. 1884–1897. DOI: 10.1002/pro.5560070905.
- [71] Laurie, A. T., Jackson, R. M., Methods for the Prediction of Protein-Ligand Binding Sites for Structure-Based Drug Design and Virtual Ligand Screening. *Curr. Protein Pept. Sci.* 7.5 (2006), S. 395–406. DOI: 10.2174/138920306778559386.
- [72] Ravindranath, P. A., Sanner, M. F., AutoSite: an automated approach for pseudo-ligands prediction—from ligand-binding sites identification to predicting key ligand atoms. *Bioinformatics* 32.20 (2016), S. 3142–3149. DOI: 10.1093/bioinformatics/btw367.
- [73] Huey, R., Morris, G. M., Olson, A. J., Goodsell, D. S., A semiempirical free energy force field with charge-based desolvation. *J. Comput. Chem.* 28.6 (2007), S. 1145–1152. DOI: 10.1002/jcc.20634.
- [74] Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., Olson, A. J., AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *J. Comput. Chem.* 30.16 (2009), S. 2785–2791. DOI: 10.1002/jcc.21256.

- [75] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., „A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise“. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, S. 226–231.
- [76] Dey, F., Cliff Zhang, Q., Petrey, D., Honig, B., Toward a “Structural BLAST”: Using structural relationships to infer function. *Protein Sci.* 22.4 (2013), S. 359–366. DOI: 10.1002/pro.2225.
- [77] Caprari, S., Toti, D., Viet Hung, L., Di Stefano, M., Polticelli, F., ASSIST: a fast versatile local structural comparison tool. *Bioinformatics* 30.7 (2013), S. 1022–1024. DOI: 10.1093/bioinformatics/btt664.
- [78] Roy, A., Yang, J., Zhang, Y., COFACTOR: an accurate comparative algorithm for structure-based protein function annotation. *Nucleic Acids Res.* 40.W1 (2012), W471–W477. DOI: 10.1093/nar/gks372.
- [79] Hwang, H., Dey, F., Petrey, D., Honig, B., Structure-based prediction of ligand–protein interactions on a genome-wide scale. *Proc. Natl. Acad. Sci. U. S. A.* 114.52 (2017), S. 13685–13690. DOI: 10.1073/pnas.1705381114.
- [80] Nagano, N., Orengo, C. A., Thornton, J. M., One Fold with Many Functions: The Evolutionary Relationships between TIM Barrel Families Based on their Sequences, Structures and Functions. *J. Mol. Biol.* 321.5 (2002), S. 741–765. DOI: 10.1016/S0022-2836(02)00649-6.
- [81] Gherardini, P. F., Wass, M. N., Helmer-Citterich, M., Sternberg, M. J., Convergent Evolution of Enzyme Active Sites Is not a Rare Phenomenon. *J. Mol. Biol.* 372.3 (2007), S. 817–845. DOI: 10.1016/j.jmb.2007.06.017.
- [82] Totrov, M. Ligand binding site superposition and comparison based on Atomic Property Fields: identification of distant homologues, convergent evolution and PDB-wide clustering of binding sites. *BMC Bioinf.* 12.1 (2011), S35. DOI: 10.1186/1471-2105-12-S1-S35.
- [83] Barelier, S., Sterling, T., O’Meara, M. J., Shoichet, B. K., The Recognition of Identical Ligands by Unrelated Proteins. *ACS Chem. Biol.* 10.12 (2015), S. 2772–2784. DOI: 10.1021/acscchembio.5b00683.
- [84] Capra, J. A., Singh, M., Predicting functionally important residues from sequence conservation. *Bioinformatics* 23.15 (2007), S. 1875–1882. DOI: 10.1093/bioinformatics/btm270.

- [85] Pai, P. P., Dattatreya, R. K., Mondal, S., Ensemble Architecture for Prediction of Enzyme-Ligand Binding Residues Using Evolutionary Information. *Mol. Inf.* 36.11 (2017), S. 1700021. DOI: 10.1002/minf.201700021.
- [86] Cortes, C., Vapnik, V., Support-vector networks. *Mach. Learn.* 20.3 (1995), S. 273–297. DOI: 10.1007/BF00994018.
- [87] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J., Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. *Nucleic Acids Res.* 25.17 (1997), S. 3389–3402. DOI: 10.1093/nar/25.17.3389.
- [88] Dessailly, B. H., Lensink, M. F., Orengo, C. A., Wodak, S. J., LigASite—A Database of Biologically Relevant Binding Sites in Proteins With Known Apo-Structures. *Nucleic Acids Res.* 36.suppl\_1 (2007), S. D667–D673. DOI: 10.1093/nar/gkm839.
- [89] Gallo Cassarino, T., Bordoli, L., Schwede, T., Assessment of ligand binding site predictions in CASP10. *Proteins: Struct., Funct., Bioinf.* 82.S2 (2014), S. 154–163. DOI: 10.1002/prot.24495.
- [90] Roche, D. B., Brackenridge, D. A., McGuffin, L. J., Proteins and Their Interacting Partners: An Introduction to Protein–Ligand Binding Site Prediction Methods. *Int. J. Mol. Sci.* 16.12 (2015), S. 29829–29842. DOI: 10.3390/ijms161226202.
- [91] Yang, J., Roy, A., Zhang, Y., Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics* 29.20 (2013), S. 2588–2595. DOI: 10.1093/bioinformatics/btt447.
- [92] Wu, Q., Peng, Z., Zhang, Y., Yang, J., COACH-D: improved protein–ligand binding sites prediction with refined ligand-binding poses through molecular docking. *Nucleic Acids Res.* 46.W1 (2018), W438–W442. DOI: 10.1093/nar/gky439.
- [93] Yang, J., Yan, R., Roy, A., Xu, D., Poisson, J., Zhang, Y., The I-TASSER Suite: protein structure and function prediction. *Nat. Methods* 12.1 (2015), S. 7–8. DOI: 10.1038/nmeth.3213.
- [94] Capra, J. A., Laskowski, R. A., Thornton, J. M., Singh, M., Funkhouser, T. A., Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure. *PLoS Comput. Biol.* 5.12 (2009), S. 1–18. DOI: 10.1371/journal.pcbi.1000585.

- [95] Brylinski, M., Skolnick, J., A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. *Proc. Natl. Acad. Sci. U. S. A.* 105.1 (2008), S. 129–134. DOI: 10.1073/pnas.0707684105.
- [96] Zhang, C., Freddolino, P. L., Zhang, Y., COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Res.* 45.W1 (2017), W291–W299. DOI: 10.1093/nar/gkx366.
- [97] Angermueller, C., Pärnamaa, T., Parts, L., Stegle, O., Deep learning for computational biology. *Mol. Syst. Biol.* 12.7 (2016), S. 878. DOI: 10.15252/msb.20156651.
- [98] Longadge, R., Dongre, S., Class Imbalance Problem in Data Mining Review. *Int. J. Comput. Sci. Netw.* 2.1 (2013), S. 83–87. DOI: 10.48550/arXiv.1305.1707.
- [99] Garg, V., Kalai, A. T., „Supervising Unsupervised Learning“. In: *Advances in Neural Information Processing Systems*. Hrsg. von S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi und R. Garnett. Bd. 31. Curran Associates, Inc., 2018, S. 4996–5006.
- [100] Kleinberg, J. „An Impossibility Theorem for Clustering“. In: *Advances in Neural Information Processing Systems*. Hrsg. von S. Becker, S. Thrun und K. Obermayer. Bd. 15. MIT Press, 2003, S. 463–470.
- [101] Jiménez, J., Doerr, S., Martínez-Rosell, G., Rose, A. S., De Fabritiis, G., DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics* 33.19 (2017), S. 3036–3042. DOI: 10.1093/bioinformatics/btx350.
- [102] LeCun, Y., Bengio, Y., Hinton, G., Deep Learning. *Nature* 521 (2015), S. 436–44. DOI: 10.1038/nature14539.
- [103] Rognan, D. *sc-PDB*. (Zugriff am 10.07.2020). URL: <http://bioinfo-pharma.u-strasbg.fr/scPDB/>.
- [104] Volkamer, A., Griewel, A., Grombacher, T., Rarey, M., Analyzing the Topology of Active Sites: On the Prediction of Pockets and Subpockets. *J. Chem. Inf. Model.* 50.11 (2010), S. 2041–2052. DOI: 10.1021/ci100241y.
- [105] Volkamer, A., Kuhn, D., Grombacher, T., Rippmann, F., Rarey, M., Combining Global and Local Measures for Structure-Based Druggability Predictions. *J. Chem. Inf. Model.* 52.2 (2012), S. 360–372. DOI: 10.1021/ci200454v.
- [106] Volkamer, A. „COMPASITES - Computer-aided active site analysis of protein structures“. Diss. Universität Hamburg, 2012.

- [107] Rarey, M., Wefing, S., Lengauer, T., Placement of medium-sized molecular fragments into active sites of proteins. *J. Comput.-Aided Mol. Des.* 10.1 (1996), S. 41–54. DOI: 10.1007/BF00124464.
- [108] Schellhammer, I., Rarey, M., TriX: structure-based molecule indexing for large-scale virtual screening in sublinear time. *J. Comput.-Aided Mol. Des.* 21.5 (2007), S. 223–238. DOI: 10.1007/s10822-007-9103-5.
- [109] Pettersen, E. F., Goddard, T. D., Huang, C. C., Meng, E. C., Couch, G. S., Croll, T. I., Morris, J. H., Ferrin, T. E., UCSF ChimeraX: Structure visualization for researchers, educators, and developers. *Protein Sci.* 30.1 (2021), S. 70–82. DOI: 10.1002/pro.3943.
- [110] Ehrt, C., Brinkjost, T., Koch, O., A Benchmark Driven Guide to Binding Site Comparison: An Exhaustive Evaluation Using Tailor-Made Data Sets (PROSPECCTs). *PLoS Comput. Biol.* 14.11 (Nov. 2018), S. 1–50. DOI: 10.1371/journal.pcbi.1006483.
- [111] Li, Y., Liu, Z., Li, J., Han, L., Liu, J., Zhao, Z., Wang, R., Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. *J. Chem. Inf. Model.* 54.6 (2014), S. 1700–1716. DOI: 10.1021/ci500080q.
- [112] Huang, W., Wang, G., Shen, Q., Liu, X., Lu, S., Geng, L., Huang, Z., Zhang, J., ASBench: Benchmarking Sets for Allosteric Discovery. *Bioinformatics* 31.15 (2015), S. 2598–2600. DOI: 10.1093/bioinformatics/btv169.
- [113] Puthenveetil, R., Vinogradova, O., Solution NMR: A powerful tool for structural and functional studies of membrane proteins in reconstituted environments. *J. Biol. Chem.* 294.44 (2019), S. 15914–15931. DOI: 10.1074/jbc.REV119.009178.
- [114] Wang, H.-W., Wang, J.-W., How cryo-electron microscopy and X-ray crystallography complement each other. *Protein Sci.* 26.1 (2017), S. 32–39. DOI: 10.1002/pro.3022.
- [115] Tivol, W. F., Briegel, A., Jensen, G. J., An Improved Cryogen for Plunge Freezing. *Microsc. Microanal.* 14.5 (2008), S. 375–379. DOI: 10.1017/S1431927608080781.
- [116] Pushpakom, S. Drug repurposing: progress, challenges and recommendations. *Nat. Rev. Drug Discovery* 18.1 (2019), S. 41–58. DOI: 10.1038/nrd.2018.168.
- [117] Breckenridge, A., Jacob, R., Overcoming the legal and regulatory barriers to drug repurposing. *Nat. Rev. Drug Discovery* 18.1 (2019), S. 1–2. DOI: 10.1038/nrd.2018.92.

- [118] An, J., Nakama, T., Kubota, Y., Sarai, A., 3DinSight: an integrated relational database and search tool for the structure, function and properties of biomolecules. *Bioinformatics* 14.2 (1998), S. 188–195. DOI: 10.1093/bioinformatics/14.2.188.
- [119] Golovin, A., Henrick, K., MSDmotif: exploring protein sites and motifs. *BMC Bioinf.* 9.1 (2008), S. 312. DOI: 10.1186/1471-2105-9-312.
- [120] EMBL-EBI, *Motifs and Sites*. (Zugriff am 26.01.2022). URL: <https://www.ebi.ac.uk/pdbe-site/pdbemotif/>.
- [121] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J., Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25.17 (1997), S. 3389–3402. DOI: 10.1093/nar/25.17.3389.
- [122] Bittrich, S., Burley, S. K., Rose, A. S., Real-time structural motif searching in proteins using an inverted index strategy. *PLoS Comput. Biol.* 16.12 (2020), S. 1–17. DOI: 10.1371/journal.pcbi.1008502.
- [123] PDB, R. *Structure Motif Search*. (Zugriff am 26.01.2022). URL: <https://www.rcsb.org/docs/search-and-browse/advanced-search/structure-motif-search>.
- [124] Hendlich, M., Bergner, A., Günther, J., Klebe, G., Relibase: Design and Development of a Database for Comprehensive Analysis of Protein–Ligand Interactions. *J. Mol. Biol.* 326.2 (2003), S. 607–620. DOI: 10.1016/S0022-2836(02)01408-0.
- [125] Bergner, A., Günther, J., Hendlich, M., Klebe, G., Verdonk, M., Use of Relibase for retrieving complex three-dimensional interaction patterns including crystallographic packing effects. *Biopolymers* 61.2 (2001), S. 99–110. DOI: 10.1002/1097-0282(2001/2002)61:2<99::AID-BIP10075>3.0.CO;2-8.
- [126] Günther, J., Bergner, A., Hendlich, M., Klebe, G., Utilising Structural Knowledge in Drug Design Strategies: Applications Using Relibase. *J. Mol. Biol.* 326.2 (2003), S. 621–636. DOI: 10.1016/S0022-2836(02)01409-2.
- [127] Pearson, W. R., Lipman, D. J., Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. U. S. A.* 85.8 (1988), S. 2444–2448. DOI: 10.1073/pnas.85.8.2444.
- [128] Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* 28.1 (1988), S. 31–36. DOI: 10.1021/ci00057a005.

- [129] Korb, O., Kuhn, B., Hert, J., Taylor, N., Cole, J., Groom, C., Stahl, M., Interactive and Versatile Navigation of Structural Databases. *J. Med. Chem.* 59.9 (2016), S. 4257–4266. DOI: 10.1021/acs.jmedchem.5b01756.
- [130] Lippert, T., Rarey, M., Fast automated placement of polar hydrogen atoms in protein-ligand complexes. *J. Cheminf.* 1.1 (2009), S. 13. DOI: 10.1186/1758-2946-1-13.
- [131] Golovin, A., Dimitropoulos, D., Oldfield, T., Rachedi, A., Henrick, K., MSDsite: A database search and retrieval system for the analysis and viewing of bound ligands and active sites. *Proteins: Struct., Funct., Bioinf.* 58.1 (2005), S. 190–199. DOI: 10.1002/prot.20288.
- [132] Golovin, A., Henrick, K., Chemical Substructure Search in SQL. *J. Chem. Inf. Model.* 49.1 (2009), S. 22–27. DOI: 10.1021/ci8003013.
- [133] Gaffney, K. P., Prammer, M., Brasfield, L., Hipp, D. R., Kennedy, D., Patel, J. M., SQLite: Past, Present, and Future. *Proc. VLDB Endow.* 15.12 (2022), S. 3535–3547. DOI: 10.14778/3554821.3554842.
- [134] Fährrolfes, R., Bietz, S., Flachsenberg, F., Meyder, A., Nittinger, E., Otto, T., Volkamer, A., Rarey, M., ProteinsPlus: a web portal for structure analysis of macromolecules. *Nucleic Acids Res.* 45.W1 (2017), W337–W343. DOI: 10.1093/nar/gkx333.
- [135] Schöning-Stierand, K., Diedrich, K., Fährrolfes, R., Flachsenberg, F., Meyder, A., Nittinger, E., Steinegger, R., Rarey, M., ProteinsPlus: interactive analysis of protein–ligand binding interfaces. *Nucleic Acids Res.* 48.W1 (2020), W48–W53. DOI: 10.1093/nar/gkaa235.
- [136] Sehnal, D., Bittrich, S., Deshpande, M., Svobodová, R., Berka, K., Bazgier, V., Velankar, S., Burley, S. K., Koča, J., Rose, A. S., Mol\* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic Acids Res.* 49.W1 (Mai 2021), W431–W437. DOI: 10.1093/nar/gkab314.
- [137] Rose, A. S., Hildebrand, P. W., NGL Viewer: a web application for molecular visualization. *Nucleic Acids Res.* 43.W1 (2015), W576–W579. DOI: 10.1093/nar/gkv402.
- [138] Rose, A. S., Bradley, A. R., Valasatava, Y., Duarte, J. M., Prlić, A., Rose, P. W., NGL viewer: web-based molecular graphics for large complexes. *Bioinformatics* 34.21 (2018), S. 3755–3758. DOI: 10.1093/bioinformatics/bty419.

- [139] Rondeau, J.-M., Schreuder, H., „Chapter 22 - Protein Crystallography and Drug Discovery“. In: *The Practice of Medicinal Chemistry (Fourth Edition)*. Hrsg. von C. G. Wermuth, D. Aldous, P. Raboisson und D. Rognan. Fourth Edition. San Diego: Academic Press, 2015, S. 511–537. ISBN: 978-0-12-417205-0. DOI: 10.1016/B978-0-12-417205-0.00022-5.
- [140] Kabsch, W., Sander, C., Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22.12 (1983), S. 2577–2637. DOI: 10.1002/bip.360221211.
- [141] Geiser, J., Schüttpelz, L., Dolfus, u., Schulze, T., *Sekundärstrukturbestimmung aus Geometriedaten*. Projektausarbeitung CiS Biochemie. Zentrum für Bioinformatik, Universität Hamburg, 2016.
- [142] SQLite.org, *Using SQLite In Multi-Threaded Applications*. (Zugriff am 11.05.2023). URL: <https://sqlite.org/threadsafe.html>.
- [143] Inhester, T. „Mining of Interaction Geometries in Collections of Protein Structures“. Diss. Universität Hamburg, 2017.
- [144] Sakallı, E. A., Teralı, K., Karadağ, A. E., Biltekin, S. N., Koşar, M., Demirci, B., Başer, K. H. C., Demirci, F., In vitro and in silico Evaluation of ACE2 and LOX Inhibitory Activity of Eucalyptus Essential Oils, 1,8-Cineole, and Citronellal. *Nat. Prod. Commun.* 17.6 (2022). DOI: 10.1177/1934578X221109409.
- [145] Faria, A. V., Fonseca, E. M., S. Fernandes-Oliveira, P., de Lima, T. I., Clerici, S. P., Justo, G. Z., Silveira, L. R., Durán, N., Ferreira-Halder, C. V., Violacein switches off low molecular weight tyrosine phosphatase and rewires mitochondria in colorectal cancer cells. *Bioorg. Chem.* 127 (2022), S. 106000. DOI: 10.1016/j.bioorg.2022.106000.
- [146] Nadzirin, N., Willett, P., Artymiuk, P. J., Firdaus-Raih, M., IMAAAGINE: a webserver for searching hypothetical 3D amino acid side chain arrangements in the Protein Data Bank. *Nucleic Acids Res.* 41.W1 (2013), W432–W440. DOI: 10.1093/nar/gkt431.
- [147] Verdonk, M. L., Boks, G. J., Kooijman, H., Kanters, J. A., Kroon, J., Stereochemistry of charged nitrogen-aromatic interactions and its involvement in ligand-receptor binding. *J. Comput. Aided Mol. Des.* 7.2 (1993), S. 173–182. DOI: 10.1007/BF00126443.
- [148] PDB, R. *File Format Documentation*. (Zugriff am 09.11.2023). URL: <https://www.wwpdb.org/documentation/file-format>.



- [149] Bietz, S., Rarey, M., ASCONA: Rapid Detection and Alignment of Protein Binding Site Conformations. *J. Chem. Inf. Model.* 55.8 (2015), S. 1747–1756. DOI: 10.1021/acs.jcim.5b00210.
- [150] Bietz, S., Rarey, M., SIENA: Efficient Compilation of Selective Protein Binding Site Ensembles. *J. Chem. Inf. Model.* 56.1 (2016), S. 248–259. DOI: 10.1021/acs.jcim.5b00588.
- [151] Fry, D. C. Targeting protein-protein interactions for drug discovery. *Methods Mol. Biol.* 1278 (2015), S. 93–106. DOI: 10.1007/978-1-4939-2425-7\_6.
- [152] Braun, P., Gingras, A.-C., History of protein-protein interactions: From egg-white to complex networks. *Proteomics* 12.10 (2012), S. 1478–1498. DOI: 10.1002/pmic.201100563.
- [153] Phizicky, E. M., Fields, S., Protein-protein interactions: methods for detection and analysis. *Microbiol. Rev.* 59.1 (1995), S. 94–123. DOI: 10.1128/mr.59.1.94-123.1995.
- [154] Wang, P. I., Marcotte, E. M., It’s the machine that matters: Predicting gene function and phenotype from protein networks. *J. Proteomics* 73.11 (2010), S. 2277–2289. DOI: 10.1016/j.jprot.2010.07.005.
- [155] Lage, K. A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nat. Biotechnol.* 25.3 (2007), S. 309–316. DOI: 10.1038/nbt1295.
- [156] Pujol, A., Mosca, R., Farrés, J., Aloy, P., Unveiling the role of network and systems biology in drug discovery. *Trends Pharmacol. Sci.* 31.3 (2010), S. 115–123. DOI: 10.1016/j.tips.2009.11.006.
- [157] Klipp, E., Wade, R. C., Kummer, U., Biochemical network-based drug-target prediction. *Curr. Opin. Biotechnol.* 21.4 (2010), S. 511–516. DOI: 10.1016/j.copbio.2010.05.004.
- [158] Scott, D. E., Bayly, A. R., Abell, C., Skidmore, J., Small molecules, big targets: drug discovery faces the protein-protein interaction challenge. *Nat. Rev. Drug Discov.* 15.8 (2016), S. 533–550. DOI: 10.1038/nrd.2016.29.
- [159] Lu, H., Zhou, Q., He, J., Jiang, Z., Peng, C., Tong, R., Shi, J., Recent advances in the development of protein-protein interactions modulators: mechanisms and clinical trials. *Signal Transduction Targeted Ther.* 5.1 (2020), S. 213. DOI: 10.1038/s41392-020-00315-3.

- [160] Gurevich, E. V., Gurevich, V. V., „Therapeutic Potential of Small Molecules and Engineered Proteins“. In: *Arrestins - Pharmacology and Therapeutic Potential*. Hrsg. von V. V. Gurevich. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 1–12. ISBN: 978-3-642-41199-1. DOI: 10.1007/978-3-642-41199-1\_1.
- [161] Fry, D. C. Protein-protein interactions as targets for small molecule drug discovery. *Biopolymers* 84.6 (2006), S. 535–552. DOI: 10.1002/bip.20608.
- [162] Wells, J. A., McClendon, C. L., Reaching for high-hanging fruit in drug discovery at protein-protein interfaces. *Nature* 450.7172 (2007), S. 1001–1009. DOI: 10.1038/nature06526.
- [163] Dömling, A. Small molecular weight protein-protein interaction antagonists: an insurmountable challenge? *Curr. Opin. Chem. Biol.* 12.3 (2008), S. 281–291. DOI: 10.1016/j.cbpa.2008.04.60.
- [164] Green, D. R. A BH3 mimetic for killing cancer cells. *Cell* 165.7 (2016), S. 1560. DOI: 10.1016/j.cell.2016.05.080.
- [165] Paller, C. J., Antonarakis, E. S., Cabazitaxel: a novel second-line treatment for metastatic castration-resistant prostate cancer. *Drug Des. Devel. Ther.* 5 (2011), S. 117–124. DOI: 10.2147/DDDT.S13029.
- [166] Goodman, S. L., Picard, M., Integrins as therapeutic targets. *Trends Pharmacol. Sci.* 33.7 (2012), S. 405–412. DOI: 10.1016/j.tips.2012.04.002.
- [167] Tcheng, J. E. Clinical pharmacology of higher dose eptifibatid in percutaneous coronary intervention (the PRIDE study). *Am. J. Cardiol.* 88.10 (2001), S. 1097–1102. DOI: 10.1016/s0002-9149(01)02041-0.
- [168] Fung, J. J. Tacrolimus and transplantation: a decade in review. *Transplantation* 77.9 Suppl (2004), S41–3. DOI: 10.1097/01.tp.0000126926.61434.a5.
- [169] Lieberman-Blum, S. S., Fung, H. B., Bandres, J. C., Maraviroc: a CCR5-receptor antagonist for the treatment of HIV-1 infection. *Clin. Ther.* 30.7 (2008), S. 1228–1250. DOI: 10.1016/s0149-2918(08)80048-3.
- [170] Van Der Ryst, E. Maraviroc - A CCR5 antagonist for the treatment of HIV-1 infection. *Front. Immunol.* 6 (2015), S. 277. DOI: 10.3389/fimmu.2015.00277.
- [171] Szklarczyk, D. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res.* 39.Database issue (2011), S. D561–8. DOI: 10.1093/nar/gkq973.
- [172] Jones, S., Thornton, J. M., Principles of protein-protein interactions. *Proc. Natl. Acad. Sci. U. S. A.* 93.1 (1996), S. 13–20. DOI: 10.1073/pnas.93.1.13.

- [173] Keskin, O., Gursoy, A., Ma, B., Nussinov, R., Principles of Protein-Protein Interactions: What are the Preferred Ways For Proteins To Interact? *Chem. Rev.* 108.4 (2008), S. 1225–1244. DOI: 10.1021/cr040409x.
- [174] Macalino, S. J. Y., Basith, S., Clavio, N. A. B., Chang, H., Kang, S., Choi, S., Evolution of In Silico Strategies for Protein-Protein Interaction Drug Discovery. *Molecules (Basel, Switzerland)* 23.8 (2018). DOI: 10.3390/molecules23081963.
- [175] Krüger, D. M., Gohlke, H., DrugScorePPI webserver: fast and accurate in silico alanine scanning for scoring protein–protein interactions. *Nucleic Acids Res.* 38.suppl\_2 (Mai 2010), W480–W486. DOI: 10.1093/nar/gkq471.
- [176] Lin, N., Wu, B., Jansen, R., Gerstein, M., Zhao, H., Information assessment on predicting protein-protein interactions. *BMC Bioinf.* 5 (2004), S. 154. DOI: 10.1186/1471-2105-5-154.
- [177] Lu, L. J., Xia, Y., Paccanaro, A., Yu, H., Gerstein, M., Assessing the limits of genomic data integration for predicting protein networks. *Genome Res.* 15.7 (2005), S. 945–953. DOI: 10.1101/gr.3610305.
- [178] Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., Chung, S., Emili, A., Snyder, M., Greenblatt, J. F., Gerstein, M., A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* 302.5644 (2003), S. 449–453. DOI: 10.1126/science.1087361.
- [179] Wang, B., Chen, P., Huang, D.-S., Li, J.-J., Lok, T.-M., Lyu, M. R., Predicting protein interaction sites from residue spatial sequence profile and evolution rate. *FEBS Lett.* 580.2 (2006), S. 380–384. DOI: 10.1016/j.febslet.2005.11.081.
- [180] Sikić, M., Tomić, S., Vlahovicek, K., Prediction of protein-protein interaction sites in sequences and 3D structures by random forests. *PLoS Comput. Biol.* 5.1 (2009). DOI: 10.1371/journal.pcbi.1000278.
- [181] Du, X., Sun, S., Hu, C., Yao, Y., Yan, Y., Zhang, Y., DeepPPI: Boosting Prediction of Protein–Protein Interactions with Deep Neural Networks. *J. Chem. Inf. Model.* 57.6 (2017), S. 1499–1510. DOI: 10.1021/acs.jcim.7b00028.
- [182] Bradshaw, R. T., Patel, B. H., Tate, E. W., Leatherbarrow, R. J., Gould, I. R., Comparing experimental and computational alanine scanning techniques for probing a prototypical protein-protein interaction. *Protein Eng. Des. Sel.* 24.1-2 (2011), S. 197–207. DOI: 10.1093/protein/gzq047.
- [183] Morrow, J. K., Zhang, S., Computational prediction of protein hot spot residues. *Curr. Pharm. Des.* 18.9 (2012), S. 1255–1265. DOI: 10.2174/138161212799436412.

- [184] Vakser, I. A. Protein-Protein Docking: From Interaction to Interactome. *Biophys. J.* 107.8 (2014), S. 1785–1793. DOI: 10.1016/j.bpj.2014.08.033.
- [185] Dominguez, C., Boelens, R., Bonvin, A. M. J. J., HADDOCK: A Protein-Protein Docking Approach Based on Biochemical or Biophysical Information. *J. Am. Chem. Soc.* 125.7 (2003), S. 1731–1737. DOI: 10.1021/ja026939x.
- [186] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., Optimization by Simulated Annealing. *Science* 220.4598 (1983), S. 671–680. DOI: 10.1126/science.220.4598.671.
- [187] De Las Rivas, J., Fontanillo, C., Protein-protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS Comput. Biol.* 6.6 (2010), e1000807. DOI: 10.1371/journal.pcbi.1000807.
- [188] Bader, G. D., Betel, D., Hogue, C. W. V., BIND: The Biomolecular Interaction Network Database. *Nucleic Acids Res.* 31.1 (2003), S. 248–250. DOI: 10.1093/nar/29.1.242.
- [189] Chatr-Aryamontri, A. The BioGRID interaction database: 2017 update. *Nucleic Acids Res.* 45.D1 (2017), S. D369–D379. DOI: 10.1093/nar/gkw1102.
- [190] Hermjakob, H. IntAct: an open source molecular interaction database. *Nucleic Acids Res.* 32.Database issue (2004), S. D452–5. DOI: 10.1093/nar/gkh052.
- [191] Aranda, B. The IntAct molecular interaction database in 2010. *Nucleic Acids Res.* 38.Database issue (2010), S. D525–31. DOI: 10.1093/nar/gkp878.
- [192] Kerrien, S. The IntAct molecular interaction database in 2012. *Nucleic Acids Res.* 40.Database issue (2012), S. D841–6. DOI: 10.1093/nar/gkr1088.
- [193] Keshava Prasad, T. S. Human Protein Reference Database–2009 update. *Nucleic Acids Res.* 37.Database issue (2009), S. D767–72. DOI: 10.1093/nar/gkn892.
- [194] Aragues, R., Sali, A., Bonet, J., Marti-Renom, M. A., Oliva, B., Characterization of protein hubs by inferring interacting motifs from protein interactions. *PLoS Comput. Biol.* 3.9 (2007), S. 1761–1771. DOI: 10.1371/journal.pcbi.0030178.
- [195] Kim, P. M., Lu, L. J., Xia, Y., Gerstein, M. B., Relating three-dimensional structures to protein networks provides evolutionary insights. *Science* 314.5807 (2006), S. 1938–1941. DOI: 10.1126/science.1136174.
- [196] Jeong, H., Mason, S. P., Barabási, A. L., Oltvai, Z. N., Lethality and centrality in protein networks. *Nature* 411.6833 (2001), S. 41–42. DOI: 10.1038/35075138.

- [197] Peng, X., Wang, J., Wang, J., Wu, F.-X., Pan, Y., Rechecking the centrality-lethality rule in the scope of Protein Subcellular Localization Interaction Networks. *PLoS One* 10.6 (2015). DOI: 10.1371/journal.pone.0130743.
- [198] Vinayagam, A. Controllability analysis of the directed human protein interaction network identifies disease genes and drug targets. *Proc. Natl. Acad. Sci. U. S. A.* 113.18 (2016), S. 4976–4981. DOI: 10.1073/pnas.1603992113.
- [199] Pazos, F., Valencia, A., In silico two-hybrid system for the selection of physically interacting protein pairs. *Proteins* 47.2 (2002), S. 219–227. DOI: 10.1002/prot.10074.
- [200] Redfern, O. C., Dessailly, B., Orengo, C. A., Exploring the structure and function paradigm. *Curr. Opin. Struct. Biol.* 18.3 (2008), S. 394–402. DOI: 10.1016/j.sbi.2008.05.007.
- [201] Orengo, C. A., Todd, A. E., Thornton, J. M., From protein structure to function. *Curr. Opin. Struct. Biol.* 9.3 (1999), S. 374–382. DOI: 10.1016/S0959-440X(99)80051-7.
- [202] Aloy, P., Ceulemans, H., Stark, A., Russell, R. B., The relationship between sequence and interaction divergence in proteins. *J. Mol. Biol.* 332.5 (2003), S. 989–998. DOI: 10.1016/j.jmb.2003.07.006.
- [203] Gao, M., Skolnick, J., iAlign: a method for the structural comparison of protein–protein interfaces. *Bioinformatics* 26.18 (2010), S. 2259–2265. DOI: 10.1093/bioinformatics/btq404.
- [204] Zhang, Y., Skolnick, J., TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* 33.7 (2005), S. 2302–2309. DOI: 10.1093/nar/gki524.
- [205] Shulman-Peleg, A., Mintz, S., Nussinov, R., Wolfson, H. J., „Protein-Protein Interfaces: Recognition of Similar Spatial and Chemical Organizations“. In: *Algorithms in Bioinformatics*. Hrsg. von I. Jonassen und J. Kim. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, S. 194–205. ISBN: 978-3-540-30219-3.
- [206] Shulman-Peleg, A., Nussinov, R., Wolfson, H. J., SiteEngines: recognition and comparison of binding sites and protein-protein interfaces. *Nucleic Acids Res.* 33.Web Server issue (2005), W337–W341. DOI: 10.1093/nar/gki482.
- [207] Schmitt, S., Kuhn, D., Klebe, G., A New Method to Detect Related Function Among Proteins Independent of Sequence and Fold Homology. *J. Mol. Biol.* 323.2 (2002), S. 387–406. DOI: 10.1016/S0022-2836(02)00811-2.

- [208] L Connolly, M. Measurement of protein surface shape by solid angles. *J. Mol. Graphics* 4.1 (1986), S. 3–6. DOI: 10.1016/0263-7855(86)80086-8.
- [209] Savojardo, C., Fariselli, P., Martelli, P. L., Casadio, R., ISPRED4: interaction sites PREDiction in protein structures with a refining grammar model. *Bioinformatics* 33.11 (2017), S. 1656–1663. DOI: 10.1093/bioinformatics/btx044.
- [210] Keskin, O., Tsai, C.-J., Wolfson, H., Nussinov, R., A new, structurally nonredundant, diverse data set of protein–protein interfaces and its implications. *Protein Sci.* 13.4 (2004), S. 1043–1055. DOI: 10.1110/ps.03484604.
- [211] BioSolveIT GmbH, *Academic Drug Discovery*. (Zugriff am 04.10.2023). URL: <https://www.biosolveit.de/academic-drug-discovery/>.
- [212] Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* 5.2 (1965), S. 107–113. DOI: 10.1021/c160017a018.
- [213] Bellmann, L., Penner, P., Rarey, M., Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration. *J. Chem. Inf. Model.* 59.11 (2019), S. 4625–4635. DOI: 10.1021/acs.jcim.9b00571.
- [214] Tanimoto, T. T. Elementary mathematical theory of classification and prediction. *IBM Internal Report* (1958).
- [215] Schrödinger, LLC, „The PyMOL Molecular Graphics System, Version 2.3“. 2019.
- [216] Virtanen, P. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* 17.3 (2020), S. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [217] Winn, M. D. Overview of the CCP4 suite and current developments. *Acta Crystallogr., Sect. D: Struct. Biol.* 67.4 (2011), S. 235–242. DOI: 10.1107/S09074444910045749.

# Literaturverzeichnis der kumulativen Dissertation

- [D1] **Graef, J.**, Ehrt, C., Rarey, M., Binding Site Detection Remastered: Enabling Fast, Robust, and Reliable Binding Site Detection and Descriptor Calculation with DoGSite3. *J. Chem. Inf. Model.* 63.10 (2023), S. 3128–3137. DOI: 10.1021/acs.jcim.3c00336.
- [D2] **Graef, J.**, Ehrt, C., Diedrich, K., Poppinga, M., Ritter, N., Rarey, M., Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures. *J. Med. Chem.* 65.2 (2022), S. 1384–1395. DOI: 10.1021/acs.jmedchem.1c01046.
- [D3] Diedrich, K., **Graef, J.**, Schöning-Stierand, K., Rarey, M., GeoMine: interactive pattern mining of protein–ligand interfaces in the Protein Data Bank. *Bioinformatics* 37.3 (2020), S. 424–425. DOI: 10.1093/bioinformatics/btaa693.
- [D4] Schöning-Stierand, K., Diedrich, K., Ehrt, C., Flachsenberg, F., **Graef, J.**, Sieg, J., Penner, P., Poppinga, M., Ungethüm, A., Rarey, M., ProteinsPlus: a comprehensive collection of web-based molecular modeling tools. *Nucleic Acids Res.* 50.W1 (2022), W611–W615. DOI: 10.1093/nar/gkac305.
- [D5] Poppinga, M., **Graef, J.**, Diedrich, K., Rarey, M., Ritter, N., „Database and Workflow Optimizations for Spatial-Geometric Queries in GeoMine“. In: *Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings*. Bd. 3630. CEUR Workshop Proceedings. Maarburg, Deutschland: CEUR-WS.org, 2023, S. 86–97. URL: <https://ceur-ws.org/Vol-3630/LWDA2023-paper8.pdf>.
- [D6] **Graef, J.**, Ehrt, C., Reim, T., Rarey, M., Database-Driven Identification of Structurally Similar Protein-Protein Interfaces. *J. Chem. Inf. Model.* (2024). Akzeptiert.

- [D7] Reim, T., Ehrt, C., **Graef, J.**, Günther, S., Meents, A., Rarey, M., SiteMine: Large Scale Binding Site Similarity Searching in Protein Structure Databases. *Arch. Pharm.* (2024). DOI: 10.1002/ardp.202300661.



## Betreute studentische Arbeiten

- [S1] Harm, J. *Analyse und Optimierung von Anfragen an geometrische Protein-Ligand-Datenbanken*. Masterarbeit. Universität Hamburg, 2023.
- [S2] Reim, T. *Development of a database-based approach to search for similar protein binding sites*. Masterarbeit. Universität Hamburg, 2020.
- [S3] Hahn, M. *Adaptive Bewertungsfunktionen für die flexible Überlagerung von Molekülen*. Masterarbeit. Universität Hamburg, 2020.
- [S4] Fender, I. *Methods to search for ligand-specific binding sites based on geometric patterns*. Masterarbeit. Universität Hamburg, 2022.



# Anhang A

## Wissenschaftliche Beiträge

In diesem Abschnitt werden die wissenschaftlichen Beiträge des Autors aufgeführt, die Themen dieses Promotionsprojekts behandeln.

M. Rarey betreute alle im Folgenden genannten Projekte und war an der Konzipierung der Methoden und Evaluierungen beteiligt.

### A.1 Publikationen

- [D1] **Graef, J.**, Ehrt, C., Rarey, M., Binding Site Detection Remastered: Enabling Fast, Robust, and Reliable Binding Site Detection and Descriptor Calculation with DoGSite3. *J. Chem. Inf. Model.* 63.10 (2023), S. 3128–3137. DOI: 10.1021/acs.jcim.3c00336.

Die Publikation erläutert die Methode DoGSite3, stellt einen Vergleich zum Vorgänger DoGSiteScorer2 her und zeigt Optimierungen auf. Anhand mehrerer Datensätze werden die Parameter der Methode evaluiert und optimiert, sowie ein Vergleich zu gängigen Methoden zur Proteinbindetaschenvorhersage erstellt. Die Implementierung von DoGSite3 und die neuen Funktionalitäten wurde von J. Graef durchgeführt. C. Ehrt evaluierte DoGSite3 und optimierte die Parameter. J. Graef, C. Ehrt und M. Rarey verfassten das Manuskript. J. Graef und C. Ehrt teilen sich die Erstautorschaft.

- [D2] **Graef, J.**, Ehrt, C., Diedrich, K., Poppinga, M., Ritter, N., Rarey, M., Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures. *J. Med. Chem.* 65.2 (2022), S. 1384–1395. DOI: 10.1021/acs.jmedchem.1c01046.

Diese Publikation umfasst eine umfangreiche Beschreibung der Datenbankerstellung, Datenbankinhalte, Suchalgorithmus und Suchmöglichkeiten von GeoMine. Zur Verdeutlichung des Nutzens der neu implementierten Funktionalitäten werden drei Anwendungsbeispiele behandelt. Die Anwendungsbeispiele wurden von C. Ehrt konzipiert. Die Implementierung neuer Funktionalitäten wurde von J. Graef durchgeführt. K. Diedrich implementierte die grafische Darstellung der neuen Funktionalitäten. M. Poppinga und N. Ritter waren am Datenbankdesign und der Datenbankanfrage beteiligt. J. Graef, C. Ehrt und M. Rarey sind die Verfasser des Manuskripts.

- [D3] Diedrich, K., **Graef, J.**, Schöning-Stierand, K., Rarey, M., GeoMine: interactive pattern mining of protein–ligand interfaces in the Protein Data Bank. *Bioinformatics* 37.3 (2020), S. 424–425. DOI: 10.1093/bioinformatics/btaa693.

Diese Publikation stellt die erste Veröffentlichung der Anwendung GeoMine dar. Sie beinhaltet eine kurze Einführung in die Methode, die Beschreibung der Benutzeroberfläche und zwei Anwendungsbeispiele. Die grafische Benutzeroberfläche von GeoMine wurde von K. Diedrich während seiner Masterarbeit entwickelt und in seinem anschließendem Promotionsprojekt überarbeitet und erweitert. Die textuelle, numerische und geometrische Suche, sowie die Datenbankerstellung wurde von J. Graef auf Basis der Anwendung PELIKAN [52] entwickelt. Die Anwendungsbeispiele wurden von K. Diedrich entworfen. Alle Autoren waren am Schreiben der Publikation beteiligt.

- [D4] Schöning-Stierand, K., Diedrich, K., Ehrt, C., Flachsenberg, F., **Graef, J.**, Sieg, J., Penner, P., Poppinga, M., Ungethüm, A., Rarey, M., ProteinsPlus: a comprehensive collection of web-based molecular modeling tools. *Nucleic Acids Res.* 50.W1 (2022), W611–W615. DOI: 10.1093/nar/gkac305.

Die Publikation beinhaltet neue Features des Modellierungsservers ProteinsPlus. Es werden neue Funktionalitäten von GeoMine, hinzugefügte Anwendungen wie JAMDA und MicroMiner, sowie eine Integration von AlphaFold Strukturen vorgestellt. Alle Autoren trugen zur Publikation bei und schrieben oder überarbeiteten Beiträge. Im Zusammenhang mit dem hier behandelten Promotionsprojekt des Autors schrieben

K. Diedrich und J. Graef unter anderem den Beitrag zu GeoMine.

- [D5] Poppinga, M., **Graef, J.**, Diedrich, K., Rarey, M., Ritter, N., „Database and Workflow Optimizations for Spatial-Geometric Queries in GeoMine“. In: *Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings*. Bd. 3630. CEUR Workshop Proceedings. Maarburg, Deutschland: CEUR-WS.org, 2023, S. 86–97. URL: <https://ceur-ws.org/Vol-3630/LWDA2023-paper8.pdf>.

Die Publikation erläutert Optimierungen der Datenbank und der Suche von GeoMine. Anhand mehrerer Experimente wird der Einfluss von PostgreSQL Anfrageoptimierungen auf die Suchlaufzeit dargestellt. M. Poppinga konzipierte und implementierte eine neue dreidimensionale Suchstrategie. Optimierungen textueller und numerischer GeoMine Anfragen wurden durch M. Poppinga und J. Graef implementiert. M. Poppinga führte die Experimente durch. Alle Autoren waren am Schreiben der Publikation beteiligt.

- [D6] **Graef, J.**, Ehrt, C., Reim, T., Rarey, M., Database-Driven Identification of Structurally Similar Protein-Protein Interfaces. *J. Chem. Inf. Model.* (2024). Akzeptiert.

Diese Publikation umfasst eine umfangreiche Beschreibung des Suchalgorithmus und der Suchmöglichkeiten von PiMine. Anhand des Vergleichs mit etablierten Methoden auf mehreren Datensätzen wird die Leistung gezeigt. Anwendungsbeispiele verdeutlichen den Nutzen der Methode. Die Implementierung der Suche nach Protein-Protein-Bindestellen Ähnlichkeiten auf Basis des TetraScan Algorithmus wurde durch J. Graef durchgeführt. Anwendungsbeispiele und neue Datensätze wurden von C. Ehrt konzipiert und erstellt. J. Graef und T. Reim entwickelten den TetraScan Algorithmus für Ähnlichkeitssuchen. Alle Autoren waren am Schreiben der Publikation beteiligt.

- [D7] Reim, T., Ehrt, C., **Graef, J.**, Günther, S., Meents, A., Rarey, M., SiteMine: Large Scale Binding Site Similarity Searching in Protein Structure Databases. *Arch. Pharm.* (2024). DOI: 10.1002/ardp.202300661.

In dieser Publikation wird die Bindetaschenähnlichkeitssuche SiteMine beschrieben. Die Methode wird auf dem ProSPECCTs Benchmark mit anderen Methoden verglichen. Ein Anwendungsbeispiel zeigt den Nutzen der Methode auf. Die Implementierung der Ähnlichkeitssuche von Bindetaschen auf Basis des TetraScan Algorithmus und die

Konzipierung des Anwendungsbeispiels wurde durch T. Reim durchgeführt. T. Reim und J. Graef entwickelten den TetraScan Algorithmus für Ähnlichkeitssuchen. Alle Autoren waren am Schreiben der Publikation beteiligt.

## A.2 (Inter-)nationale Konferenzbeiträge

Im Folgenden werden alle vom Autor auf Konferenzen vorgestellten Beiträge aufgeführt.

### A.2.1 Vorträge

- [V1] **Graef, J.**, Ehrt, C., Diedrich, K., Poppinga, M., Ritter, N., Rarey, M., *GeoMine: On-The-Fly Geometric Pattern Mining in Binding Sites*. International Conference on Chemical Structures (ICCS). Noordwijkerhout, Niederlande, 2022. DOI: 10.5281/zenodo.6837572.

### A.2.2 Poster

- [P1] Diedrich, K., **Graef, J.**, Nittinger, E., Rarey, M., *GeoMine: A Web-Based Tool for chemical 3D Searching of the PDB*. German Conference on Cheminformatics (GCC). Mainz, Deutschland, 2019.
- [P2] **Graef, J.**, Diedrich, K., Schöning-Stierand, K., Rarey, M., *GeoMine: A Web-Based Tool for Chemical Three-Dimensional Searching of the PDB*. International Society For Computational Biology (ISMB). Virtuelle Konferenz, 2020.

## Anhang B

# Weitere Analysen

In diesem Kapitel werden Analysen behandelt, die im Rahmen des Promotionsprojekts durchgeführt wurden, allerdings kein Bestandteil einer der veröffentlichten Publikationen sind.

### B.1 dogsite.v3

Zur Diskretisierung des Proteins berechnet der DoGSite-Algorithmus abhängig von der Proteinorientierung ein Gitter. Änderungen dieser Orientierung beeinflussten die berechneten Taschendescriptoren wie die Größe und Tiefe der vorhergesagten Taschen in dogsite.v2 (verfügbar als DoGSiteScorer 2.0.0 unter [211]) erheblich. Aus diesem Grund wurde in dogsite.v3, wie in Kapitel 2.2.1 beschrieben, eine einheitliche orientierungsunabhängige Gitterberechnung implementiert. Diese stellt sicher, dass die gleichen Taschen und somit ebenfalls gleiche Taschendescriptoren bestimmt werden. Dennoch ist es wichtig, dass die Stabilität der Deskriptoren auch ohne diese hinzugefügte Funktionalität möglichst hoch ausfällt, da durch sie eine Proteinorientierung festgelegt wird. Um dies zu überprüfen, wurde in einem Experiment die Stabilität bzw. die Standardabweichung der folgenden Deskriptoren für die Versionen dogsite.v2, dogsite.v3 mit Parametern von dogsite.v2 und dogsite.v3 mit den neu optimierten Parametern bestimmt: Anzahl der Atome, Anzahl der Wasserstoffbrückenakzeptoren, Anzahl der Wasserstoffbrückendonoren, Volumen und Tiefe (siehe Abbildung B.1). Die Analyse wurde auf dem zweiten Datensatz der ProSPECCTS Benchmarks [110] durchgeführt, welcher verschiedene Proteinstrukturen von NMR Ensembles (17 Gruppen, 329 Strukturen) enthält. Für alle Strukturen wurden je 200 zufällige Proteinausrichtungen mithilfe des Rotationsmoduls

der Python-Bibliothek SciPy [216] berechnet. Die Matrizen dieser gleichmäßig verteilten Rotationen wurden als Eingabe für das CCP4-Tool pdbset [217] verwendet, um die jeweiligen Strukturen bzw. deren Koordinaten zu transformieren.

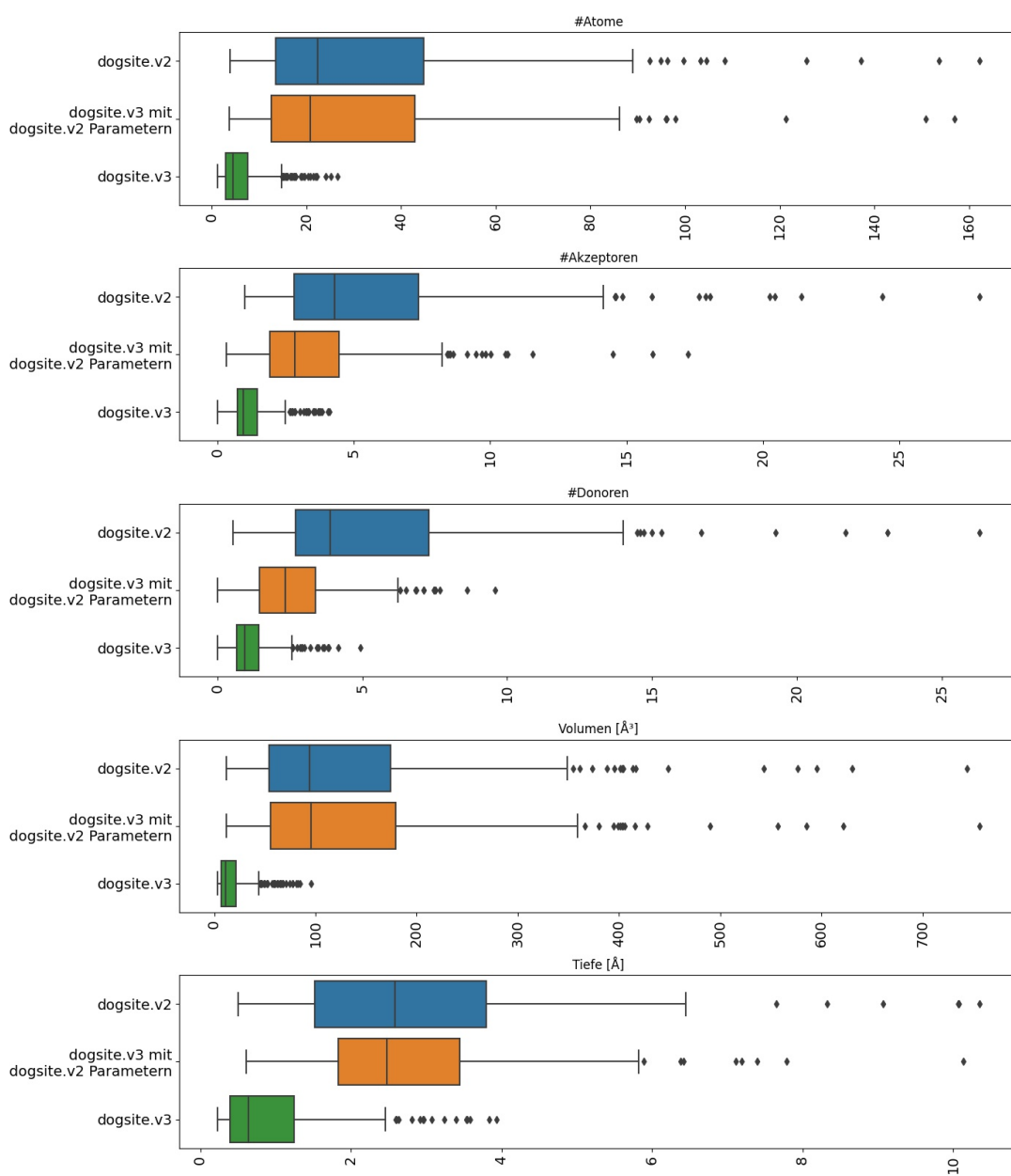
Die Analysen zeigen, dass dogsite.v2 (blau) ausgenommen im Fall des Volumens eine höhere Varianz der Standardabweichungen aufweist als dogsite.v3 mit denselben Parametern. Im Allgemeinen zeigen die Ergebnisse im Hinblick auf dogsite.v2 und dogsite.v3 mit alten Standardparametern, dass die Optimierungen am Algorithmus einen positiven Effekt ausüben. Für dogsite.v3 mit den neuen optimierten Parametern (grün) ist darüber hinaus der Vorteil gegenüber dogsite.v2 in allen Fällen deutlich erkennbar und belegt eine signifikante Reduzierung der mittleren Standardabweichung und damit eine erhöhte Stabilität der berechneten Taschenskriptoren.

Wie in Abbildung B.1 ersichtlich, ist die durchschnittliche Standardabweichung des Volumens und deren Varianz mit dogsite.v3 erheblich geringer als mit dogsite.v2. Während die vorhergesagten Taschen von dogsite.v3 im Durchschnitt eine Abweichung von etwa  $20 \text{ \AA}^3$  aufweisen, sind es bei dogsite.v2 ca.  $95 \text{ \AA}^3$ . Dies ist hauptsächlich auf die deutlich höheren Taschenvolumina und verwendeten Parameter zurückzuführen. Ein Vergleich einer Tasche in dogsite.v2 und dogsite.v3 ist in Abbildung B.2 dargestellt. Hier ist das größere Taschenvolumen mit dogsite.v2 (Abb. B.2a) gegenüber dogsite.v3 (Abb. B.2b) offensichtlich erkennbar. Ein Grund für kleinere Taschen in dogsite.v3 ist die Korrektur eines Konzeptfehlers beim Kombinieren von *Subpockets*. In dogsite.v2 wurden alle *Subpockets* miteinander kombiniert, sobald mindestens ein Gitterpunkt der einen *Subpocket* neben einem der anderen lag. Dies kann aufgrund der Ungenauigkeiten der Gitterdarstellung des Proteins zu dem in Abbildung B.3 dargestellten Problem führen. dogsite.v2 bestimmt in diesem Beispiel zwei *Subpockets*, die in dem orange umrandeten Bereich kombiniert werden, da jeweils ein Gitterpunkt beider *Subpockets* nebeneinander liegt. In der gezeigten Vergrößerung lässt sich unter Hinzunahme der Proteinoberfläche erkennen, dass die beiden Taschen über einen Bereich miteinander kombiniert werden, der innerhalb des Proteins liegt. In dogsite.v3 wurde dieses Problem gelöst, indem eine Formel erstellt wurde, mit der eine Mindestanzahl an Gitterpunkten berechnet wird, die die Voraussetzung für ein Kombinieren von Taschen darstellt.

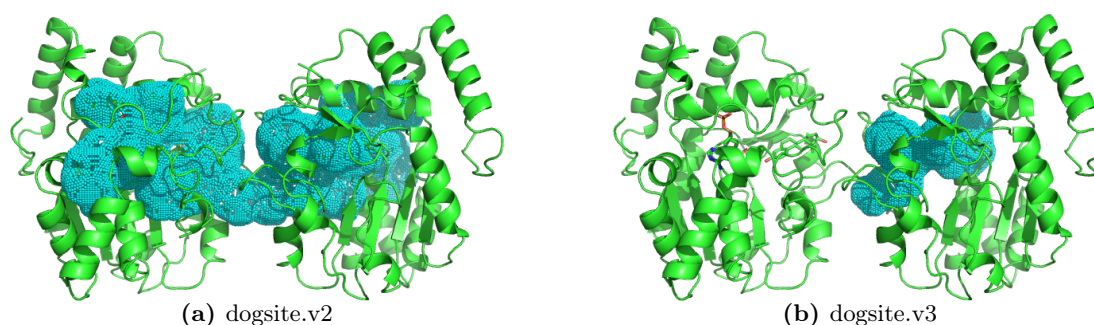
## B.2 GeoMine

In Laufzeitanalysen wurde evaluiert, wie schnell mit PELIKAN und GeoMine jeweils eine Datenbank für die sc-PDB [35, 103] (Version 2017) erstellt werden kann. Da GeoMine

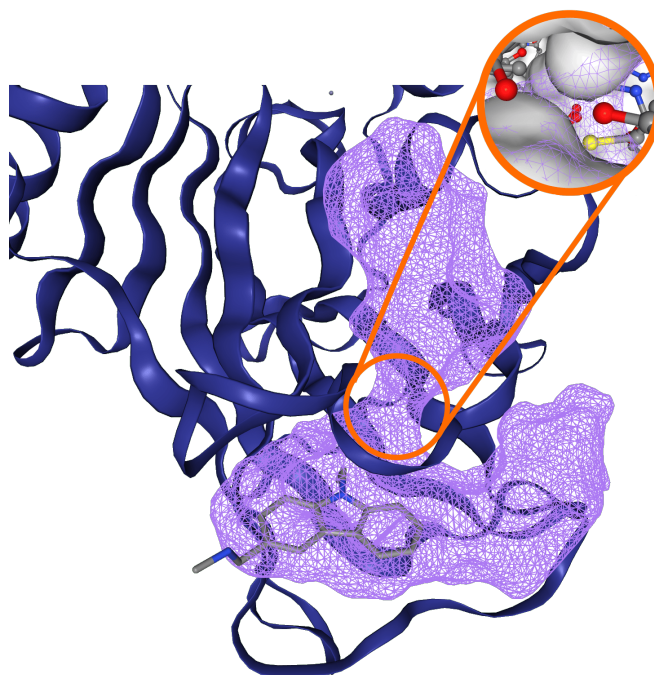




**Abbildung B.1:** Analyse der Stabilität bzw. der Standardabweichung von Taschendescriptoren. Die Anzahl der Atome, Anzahl der Wasserstoffbrückenakzeptoren, Anzahl der Wasserstoffbrückendonoren, Volumen und Tiefe für dogsite.v2 (blau), dogsite.v3 mit Parametern von dogsite.v2 (orange) und dogsite.v3 mit neuen und optimierten Parametern (grün) sind dargestellt. Die Durchführung fand auf den Gruppen des zweiten Datensatzes der ProSPECCTS [110] Benchmarks statt. Für alle Strukturen wurden je 200 zufällige Proteinausrichtungen berechnet.



**Abbildung B.2:** Darstellung der vorhergesagten Taschen mit dogsite.v2 und dogsite.v3 für die Proteinstruktur mit dem PDB-Code 1AQU. Protein- und Gitterabbildungen, erstellt mit PyMol [215].



**Abbildung B.3:** Fehlerhafte Kombination von *Subpockets* an der orange umrandeten Stelle im Proteinkomplex mit PDB-Code 2VUK. Eine Vergrößerung des Bereichs, in der die Kombination stattfindet, mit Darstellung von Proteinatomen und der Proteinoberfläche (grau), zeigt ein aufgrund der Gitterungenauigkeit in das Protein hineinreichendes Gitter. Molekül- und Gitterdarstellung, erstellt mithilfe des NGL Viewers [137, 138].

neben den radiusbasierten Taschen auch DoGSite-basierte Taschen berechnen kann, werden beide Experimente separat durchgeführt. Zu beachten ist bei der Berechnung von DoGSite-basierten Taschen, dass eine radiusbasierte Tasche für jeden Liganden berechnet wird, wenn für diesen Liganden keine Tasche unter den DoGSite-basierten Taschen gefunden wird. Im Vergleich zu PELIKAN überprüft GeoMine das Verhältnis der Lösungsmittelzugänglichkeit von Liganden im gebundenen und ungebundenen Zustand.

Hierbei wird der Ligand wie in `dogsites.v3` (siehe Kapitel 2.2.1) in rigide Einheiten unterteilt. Wenn das Verhältnis keiner rigiden Einheit unter einer Benutzer-definierten Schranke  $S$  mit  $\text{gebundeneSAS}/\text{ungebundeneSAS} \leq S$  liegt, wird dieses Molekül bei der Datenbankerstellung nicht länger berücksichtigt. Um feststellen zu können, wie lange die Datenbankerstellung mit GeoMine benötigt, wenn die Liganden nicht auf diese Weise gefiltert werden, wird ein weiteres Experiment durchgeführt. Da im Laufe dieses Promotionsprojekts die Oberflächenberechnung in NAOMI optimiert wurde (Beschreibung siehe Kapitel D.2), wird ein weiteres Experiment ohne diese Optimierung dargestellt. Dabei ist zu beachten, dass diese Analyse nur zu den anderen GeoMine-Experimenten in Beziehung gesetzt werden kann, da in PELIKAN keine Oberflächenannotation vorhanden ist. Auch kann die Erstellung der GeoMine Datenbank durch eine Parallelisierung beschleunigt werden, weshalb ihr Einfluss ebenfalls analysiert wird. Jedes dieser Experimente wurde fünfmal ausgeführt und der Mittelwert dieser Laufzeiten wurde zum Vergleich herangezogen. Die Resultate sind in Tabelle B.1 gelistet. Durchgeführt wurden sie auf einem Rechner mit einem Intel Xeon E5640 (2,67 GHz) Prozessor, 64 GB Arbeitsspeicher und einer WD RE4 WD2003FYYS HDD (2 TB).

**Tabelle B.1:** Mittlere Laufzeiten von jeweils fünf unabhängigen Datenbankerstellung für die sc-PDB [35, 103] (Version 2017) mit PELIKAN und GeoMine. Zwei Experimente mit PELIKAN und sechs mit GeoMine werden dargestellt. BOB = Beschleunigte Oberflächenberechnung, EML = Entfernen von Molekülen mit einer hohen Lösungsmittelzugänglichkeit

Datenbankerstellungsexperiment	Laufzeit [min]
PELIKAN	2 480 ± 58
PELIKAN mit Dreiecksindexstruktur	3 065 ± 21
GeoMine – Radius-Taschen mit BOB	1 564 ± 44
GeoMine – Radius-Taschen mit BOB und ohne EML	5 206 ± 6
GeoMine – DoGSite-Taschen mit BOB	2 971 ± 58
GeoMine – DoGSite-Taschen ohne BOB	10 095 ± 208
GeoMine – DoGSite-Taschen mit BOB und 2 Threads	1 647 ± 9
GeoMine – DoGSite-Taschen mit BOB und 4 Threads	1 056 ± 6

PELIKAN benötigt ohne bzw. mit Berechnung der Dreiecksindexstruktur eine Laufzeit zur Datenbankerstellung von 2 480 Minuten bzw. 3 065 Minuten. Mit Berechnung der Radius-Taschen und allen Optimierungen erstellt GeoMine die Datenbank in 1 564 Minuten und ist damit etwa 900 bzw. 1 500 Minuten schneller als PELIKAN ohne bzw. mit der Berechnung der Dreiecksindexstruktur. Wenn hingegen im Erstellungsprozess keine Moleküle mit hoher Lösungsmittelzugänglichkeit entfernt werden, erhöht sich die

Laufzeit auf 5 206 Minuten und damit auf mehr als das Doppelte der PELIKAN Laufzeit ohne Berechnung der Dreiecksindexstruktur. Dies liegt darin begründet, dass in GeoMine mehr Eigenschaften als in PELIKAN berechnet werden. Besonders die Berechnung der Datenstruktur für schnelle Ähnlichkeitssuchen von Molekülen, die Proteinoberfläche und hinzugefügte Indizes beeinflussen die Dauer der Datenbankerstellung. Da einer der Hauptunterschiede zwischen GeoMine und PELIKAN die Erweiterung des Suchraums durch vorhergesagte Bindetaschen ist, ist insbesondere die Laufzeit mit DoGSite-Taschen von Interesse. Mit einer Dauer von 2 971 Minuten dauert die Erstellung dieser Datenbank knapp doppelt so lang wie die der Erstellung, wenn nur Taschen mit Liganden berücksichtigt werden. Im Vergleich zur Berechnung der PELIKAN-Datenbank mit Dreiecksindexstruktur ist die Laufzeit mit einem Unterschied von durchschnittlich 94 Minuten vergleichsweise ähnlich. Der DoGSite-Algorithmus und die Berechnung der Taschen-Eigenschaften nehmen damit einen signifikanten Anteil der Laufzeit ein. Eine weitere Analyse zeigt hierbei die Auswirkung einer Optimierung der Oberflächenberechnung während des Promotionsprojekts. So dauert die Erstellung der Datenbank ohne die Optimierung 10 095 Minuten und dauert damit etwa 3,4x länger als mit der Optimierung. Der Nutzen der Optimierungen ist somit offensichtlich. Im Allgemeinen zeigen die Analysen von GeoMine im Vergleich zu PELIKAN, dass die Erstellung der Datenbank trotz neuer Features nicht wesentlich mehr Zeit benötigt. Besonders die Reduzierung der zu berücksichtigenden Moleküle durch die Berechnung der Lösungsmittelzugänglichkeit sorgt für eine Reduktion der Laufzeit und damit einen Ausgleich zur vergleichsweise zeitaufwendigen Berechnung der DoGSite-Taschen und Oberflächen. Unter Verwendung der Parallelisierung in GeoMine ist die Erstellung von Datenbanken mit 1 647 bzw. 1 056 Minuten für die Verwendung mit DoGSite-Taschen, beschleunigter Oberflächenberechnung und 2 bzw. 4 Threads deutlich schneller als in PELIKAN. Die Laufzeit mit 2 bzw. 4 Threads kann im Vergleich zur Nutzung mit einem Thread um ca. 44 % bzw. 64 % weiter reduziert werden. Jede Eingabe-Struktur wird hierbei auf einem eigenen Thread prozessiert, sodass eine maximale Beschleunigung der Datenbankerstellung bei  $Threads = n$ , mit  $n$  als Anzahl der Eingabe-Strukturen, erzielt wird. Zudem zeigen separate Analysen, dass trotz der Berechnung weiterer Bindetaschendescriptoren auch die Datenbankgröße verringert werden konnte. Durch Optimierungen in der Speicherung der Protein- und Moleküleigenschaften werden für die sc-PDB mit ligandbasierten Taschen und ohne Entfernen von Molekülen 9,7 GB benötigt, während es in PELIKAN 11 GB sind. Die größte Reduzierung beläuft sich darauf, dass in PELIKAN eine Tasche pro Ligand abgespeichert wird, während GeoMine für mehrere Liganden nur einmal eine Tasche speichert, insofern die Taschen identisch sind. Dies verhindert

eine Abspeicherung von Duplikaten der Suchpunkte und verringert den Speicherbedarf (siehe Analysen in [S1]).



## Anhang C

# Suchfilter in GeoMine

In diesem Kapitel werden alle Suchfilter aufgelistet, die zum Zeitpunkt der Abgabe dieser Dissertationsschrift in GeoMine implementiert sind. In Tabelle C.1 werden die Eigenschaften und Auswahlmöglichkeiten von Suchpunkten in der geometrischen Anfrage gezeigt. Tabelle C.2 und Tabelle C.3 beinhalten die verfügbaren textuellen und numerischen Suchfilter. Alle Angaben sind in Englisch, da das Programm und die Suchfilter nur in Englisch verfügbar sind. Kursive Einträge zeigen neue Filter von GeoMine im Vergleich zu PELIKAN an. Neben den dargestellten Filtern kann außerdem eine Liste an PDB-Codes übergeben werden, auf die der Suchraum reduziert werden soll.

Die Suche wird mithilfe von XML-Abfragedateien durchgeführt. Die Struktur dieser Dateien wird im Folgenden beschrieben.

Alle Filter beginnen mit der gleichen Kopfzeile (außer dem beliebigen Feld "name"):

```
<!DOCTYPE GeoMineFilterPresets>
<GeoMineFilter xmlns:i='urn:naomi:InteractionDB'
xmlns:ip='urn:naomi:GeoMine' xmlversion='6'
xmlns:propertydb='urn:naomi:PropertyDB' name='7AQI'
xmlns:m='urn:naomi:MoleculeDB'>
```

Nach der Kopfzeile folgen die textuellen und numerischen Filter. Diese beginnen mit einer Liste von Strukturen auf der Grundlage ihrer PDB-Codes, um den Suchraum einzugrenzen. Jeder der PDB-Codes ist wie folgt angegeben:

```
<propertydb:SubstringFilter_OR rule='including'
id='intPatterns.protein_pdb_id' subsettype='2'>
<substring_element substring='1A00' />
```

**Tabelle C.1:** Eigenschaften und Auswahlmöglichkeiten für Suchpunkte in geometrischen Anfragen. Kursive Einträge stellen GeoMine Filter dar, die im Laufe dieses Promotionsprojekts implementiert wurden und nicht aus PELIKAN stammen. Alle Angaben sind zur Einhaltung der Konsistenz zum Programm und den Suchfiltern in englischer Sprache.

Property		Possible Choices
Original molecule		Ligand, Metal, Protein, <i>Nucleic Acid</i> , Water
Element		Boron, Bromine, Calcium, Carbon, Chlorine, Cobalt, Copper, Fluorine, Iodine, Iron, Magnesium, Manganese, Nickel, Nitrogen, Oxygen, Phosphorus, Sulfur, Zinc
Interaction type		Acceptor, Anion, AromaticRingCenter, Cation, Donor, Hydrophobic, Metal
If Original molecule = Protein	Amino acid	Ala, Arg, Asn, Asp, Cso, Cys, Glu, Gln, Gly, His, Ile, Leu, Lys, Met, Phe, Pro, Ser, Thr, Trp, Tyr, Val, Hydrophobic, Polar, Aromatic, Acidic, Basic, Neutral
	Secondary Structure	Helix, Sheet, <i>Helix End</i> , <i>Helix C Terminus</i> , <i>Helix N Terminus</i> , <i>Helix Mid</i> , <i>Strand End</i> , <i>Strand Mid</i> , no secondary structure
If Original molecule = Nucleic Acid	<i>Residue</i>	<i>Adenosine</i> , <i>Cytidine</i> , <i>Guanosine</i> , <i>Inosine</i> , <i>Nucleoside</i> , <i>Uridine</i> , <i>Deoxyadenosine</i> , <i>Deoxycytidine</i> , <i>Deoxyguanosine</i> , <i>Deoxynukleoside</i> , <i>Deoxythymidine</i>
If Original molecule = Protein or Nucleic Acid	Location in amino acid	Backbone, Sidechain
	<i>Minimal surface</i>	<i>All floating-point numbers <math>\geq 0 \text{ \AA}^2</math></i>
If Original molecule = Ligand	Functional Group	Alcohol, Aldehyde, Amide, Amidine, Amine, Azide, Ester, Ether, Furane, Guanidine, Ketone, Nitrile, Phenyl, Pyridine, Pyrrole, Thiophene
If Orig molecule = Ligand or Protein	Atom description	SMARTS



**Tabelle C.2:** Textuelle und numerische Filter in GeoMine. Kursive Einträge stellen GeoMine Filter dar, die im Laufe dieses Promotionsprojekts implementiert wurden und nicht aus PELIKAN stammen. Alle Angaben sind zur Einhaltung der Konsistenz zum Programm und den Suchfiltern in englischer Sprache.

Category	Property	Possible choices
Ligand filter	Element	Boron, Bromine, Carbon, Chlorine, Fluorine, Iodine, Nitrogen, Oxygen, Phosphorus, Sulfur, and a count (min, max)
	Functional group	Alcohol, Aldehyde, Amide, Amidine, Amine, Azide, Ester, Ether, Furane, Guanidine, Ketone, Nitrile, Phenyl, Pyridine, Pyrrole, Thiophene, and a count (min, max)
	Molecule property	Acceptors, aromatic atoms, aromatic rings, aromatic ring systems, charge, cyclomatic number, donors, halogens, heavy atoms, hetero atoms, inorganic atoms, Lipinski acceptors, logP, molecular weight, max continuous path of rotatable bonds, max cyclomatic number, max ring size, max ring system size, rings, ring systems, rotatable bonds, stereo bonds (E/Z), stereo centers (R/S), topological polar surface area, unique ring families (URFs), volume, and a count (min, max)
	<i>Similarity</i>	<i>CSFP, tCSFP, ECFPlike, a similarity percentage the pocket name to which other ligands should be similar to and a min and max for the CSFP and tCSFP or a radius for the ECFPlike Morgan Fingerprint</i>
	SMARTS	SMARTS String
Protein filter	Uniprot ID	Free text
	EC number	All four number can be set individually

**Tabelle C.3:** Textuelle und numerische Filter in GeoMine. Kursive Einträge stellen GeoMine Filter dar, die im Laufe dieses Promotionsprojekts implementiert wurden und nicht aus PELIKAN stammen. Alle Angaben sind zur Einhaltung der Konsistenz zum Programm und den Suchfiltern in englischer Sprache.

Category	Property	Possible choices
Pocket filter	Ligand name	Free text
	Amino acid	Ala, Arg, Asn, Asp, Cso, Cys, Glu, Gln, Gly, His, Ile, Leu, Lys, Met, Phe, Pro, Ser, Thr, Trp, Tyr, Val, and a count (min, max)
	Property	Acceptors, depth, donors, protein heavy atoms, hydrophobicity, metal, surface, surface-volume-ratio, volume, and a range (min, max)
	<i>Has ligand?</i>	<i>If filter is set only pockets containing a ligand are searched.</i>
	<i>Is multichain?</i>	<i>If filter is set only pockets consisting of at least two chains are searched.</i>
P-L-Complex filter	PDB title entry	Free text
	Resolution	Range (min, max)
	Experimental source	Unknown, electron crystallography, electron microscopy, fiber diffraction, NMR solid state, NMR solution, neutron diffraction, solution scattering, X-ray
	Organism	Free text
	Excluded PDB codes	List of PDB codes that are to be excluded from the search space.
<i>Result filter</i>	<i>Max number of pockets</i>	<i>Value bigger than zero</i>
	<i>No symmetrical matches</i>	<i>If filter is set and multiple matches consisting of the same atoms are found, only one match is returned.</i>
	<i>Query RMSD</i>	<i>Range (min, max)</i>

```
<substring_element substring='12GS' />
</propertydb:SubstringFilter_OR>
```

Alle Liganden in der Datenbank können nach einem oder mehreren SMARTS-Mustern durchsucht werden:

```
<m:SMARTSFilter>
<SMARTS method='include'> nCCON </SMARTS>
</m:SMARTSFilter>
```

EC-Nummern können auf folgende Weise angegeben werden. Alle vier Ziffern können angepasst werden:

```
<ip:ECFilter_Chain>
<ECFilter rule='including' ecnumber='2.1.0.0' />
</ip:ECFilter_Chain>
```

Ähnlichkeitssuchen können mit einem eindeutigen SMILES, der minimalen Ähnlichkeit und dem zu verwendenden Min/Max/Durchmesser durchgeführt werden:

```
<ip:SimilarityFilter_Chain>
<SimilarityFilter simPercentage='70' simMaxOrDiameter='10'
simMin='1' simUSmiles='[O-]C(=O)' variant='CSFP' rule='including' />
</ip:SimilarityFilter_Chain>
```

Mit dem folgenden Filter kann der Suchraum auf Taschen beschränkt werden, die (keine) Liganden enthalten:

```
<ip:PocketHasLigandFilter_Chain>
<PocketHasLigandFilter pocketHasLigand='true' />
</ip:PocketHasLigandFilter_Chain>
```

Um alle Matches, die die gleichen Atome enthalten, bis auf eine einzige zu entfernen, kann folgende Zeile verwendet werden:

```
<ip:filterSymmetricMatchesFilter filterSymmetricMatches='true' />
```

Matches können auch durch ihre RMSD zum definierten geometrischen Abfragemuster begrenzt werden:

```
<ip:RMSDfilter min='0' max='4.2' />
```

Um nur maximal  $x$  Matches pro Tasche zurückzugeben, wird folgende Zeile verwendet:

```
<ip:MaxMatchesPerPocketFilter max='100' />
```

Alle folgenden Suchfilter sind in Liganden- (0), Protein- (2) und Taschenfilter (3) unterteilt. Dies wird durch den *SubsetType* angegeben. Das Schlüsselwort "rule" kann

entweder "including" oder "excluding" sein, sodass der angegebene Filter alle (nicht) passenden Taschen entfernt. Im Folgenden werden einige der verfügbaren Filter gezeigt.

Filterzahl bestimmter chemischer Elemente im gebundenen Liganden einer Tasche:

```
<propertydb:PropertyFilter subsettype='0' rule='including'  
id='intPatterns.Boron' min='0' max='100' />
```

Filterung der Anzahl einer bestimmten funktionellen Gruppe im gebundenen Liganden:

```
<propertydb:PropertyFilter subsettype='0' rule='including'  
id='intPatterns.Alcohol' min='0' max='100' />
```

Filterung der Anzahl der Wechselwirkungspunkte der gebundenen Liganden, z. B. Wasserstoffakzeptoren:

```
<propertydb:PropertyFilter subsettype='0' rule='including'  
id='naomi.Acceptors' min='3' max='6' />
```

Filterung anhand der experimentellen Methode, die zur Erzeugung der Proteinstruktur verwendet wurde:

```
<propertydb:PropertyFilter subsettype='2' rule='including'  
id='intPatterns.protein_expType0' min='11' max='11' />
```

Filtern der Proteine auf Basis der Strukturauflösung in Ångström:

```
<propertydb:PropertyFilter subsettype='2' rule='including'  
id='intPatterns.protein_resolution' min='0.0' max='3.0' />
```

Filtern der Taschen anhand der Anzahl bestimmter Aminosäurentypen:

```
<propertydb:PropertyFilter subsettype='3' rule='including'  
id='intPatterns.ala_Pocket' min='0' max='100' />
```

Reduzieren des Suchraums auf Taschen, die eine Anzahl von Wechselwirkungspunkten beinhalten:

```
<propertydb:PropertyFilter subsettype='3' rule='including'  
id='intPatterns.Acceptors' min='0' max='100' />
```

Wenn der folgende Filter angegeben wird, werden alle Taschen aus der Suche entfernt bzw. die Suche auf diese reduziert, die zwischen mehreren Ketten liegen. Dies wird durch Angabe von "including" oder "excluding" gesteuert:

```
<propertydb:PropertyFilter id='intPatterns.MultiChain'  
rule='including' subsettype='3' max='1' min='1' />
```

Die folgenden Filter werden für die Suche von geometrischen Mustern verwendet. Geometrische Muster bestehen aus Abstands- und/oder Interaktionsfiltern, die zwi-

schen zwei Punkten definiert sind. Die Abstandsfiler haben ein minimales und ein maximales Abstandselement, während die Interaktionsfilter einen Interaktionstyp haben. Jeder Abstands-, Interaktions- und Punktfilter hat eine eindeutige ID, die die Filter voneinander unterscheidet. Punktfilter sind unterteilt in "ligandfilter", "aminoacidfilter", "nucleicacidfilter", "metalfilter" und "waterfilter":

```
<i:InteractionDBfilterchain>
<i:Pointfilter />
<i:Interactionfilter>
<distancefilter inter_point1='1-Any' inter_point2='2-Any'
id='6' inter_mxadist='4.4' inter_mindist='3.4'>
<firstpoint>
<ligandfilter point_fungroup='ANY' point_coord_y='-1.69'
point_coord_x='6.09' id='1-Any' point_intertype='ANY'
point_surface='-1' point_coord_z='-5.24'
point_element='7'></ligandfilter>
</firstpoint>
<secondpoint>
<aminoacidfilter point_aminoacid='UNKNOWN'
point_secstruct='UNKNOWN'
point_aminoacid_class='UNDEFINED'
point_backbone_sidechain='UNDEFINED' point_surface='0.1'
point_coord_y='0.17' point_coord_x='2.83' id='2-Any'
point_intertype='ANY' point_coord_z='-6.32'
point_element='7'></aminoacidfilter>
</secondpoint>
</distancefilter>
<distancefilter inter_point1='3-Any' inter_point2='2-Any'
id='7' inter_mxadist='8.4' inter_mindist='7.4'>
<firstpoint>
<waterfilter point_coord_y='2.36' point_coord_x='10.15'
id='3-Any' point_intertype='ANY' point_coord_z='-8.46'
point_surface='-1' point_element='7'></waterfilter>
</firstpoint>
<secondpoint>
<aminoacidfilter point_aminoacid='UNKNOWN'
point_secstruct='UNKNOWN'
```

```
point_aminoacid_class='UNDEFINED'  
point_backbone_sidechain='UNDEFINED' point_surface='0.1'  
point_coord_y='0.17' point_coord_x='2.83' id='2-Any'  
point_intertype='ANY' point_coord_z='-6.32'  
point_element='7'></aminoacidfilter>  
</secondpoint>  
</distancefilter>  
<distancefilter inter_point1='1-Any'  
inter_point2='10-Aromatic' id='12' inter_mxadist='1.6'  
inter_mindist='0.6'>  
<firstpoint>  
<ligandfilter point_fungroup='ANY' point_coord_y='-1.69'  
point_coord_x='6.09' id='1-Any' point_intertype='ANY'  
point_surface='-1' point_coord_z='-5.24'  
point_element='6'></ligandfilter>  
</firstpoint>  
<secondpoint>  
<metalfilter point_coord_y='-1.63' point_coord_x='6.15'  
id='10-Aromatic' point_surface='-1'  
point_intertype='AROMATIC' point_coord_z='-6.38'  
point_element='0'></metalfilter>  
</secondpoint>  
</distancefilter>  
<interactionfilter inter_intertype='PI_PI'  
inter_point1='9-Aromatic' inter_point2='10-Aromatic' id='11'>  
<firstpoint>  
<nucleicacidfilter point_aminoacid='UNKNOWN'  
point_secstruct='UNKNOWN'  
point_aminoacid_class='UNDEFINED'  
point_backbone_sidechain='UNDEFINED' point_surface='0'  
point_coord_y='-4.67' point_coord_x='8.54'  
id='9-Aromatic' point_intertype='AROMATIC'  
point_coord_z='-4.25' point_element='0'>  
</nucleicacidfilter>  
</firstpoint>  
<secondpoint>
```

```
<metalfilter point_coord_y='-1.63' point_coord_x='6.15'  
id='10-Aromatic' point_surface='-1'  
point_intertype='AROMATIC' point_coord_z='-6.38'  
point_element='0'></metalfilter>  
</secondpoint>  
</interactionfilter>  
</i:Interactionfilter>  
</i:InteractionDBfilterchain>
```

pointSMARTS können verwendet werden, um die Umgebung von Punktfilttern näher zu beschreiben. Die werden mithilfe der SMARTS-Sprache definiert:

```
<pointSMARTS id='2' smarts='N[0,C]'/>
```

Winkel können zwischen zwei Abständen, Wechselwirkungen, Ringnormalen oder Sekundärstrukturausrichtungen durch Angabe der entsprechenden ID und eines Winkelbereichs definiert werden:

```
<ip:AngleFilter_Chain>  
<AngleFilter secondinterid='7' id='13' max='1.234'  
firstinterid='6' min='0.710' />  
</ip:AngleFilter_Chain>
```

Zuletzt wird die XML-Filterdatei durch den schließenden Tag abgeschlossen:  
</GeoMineFilter>





## Anhang D

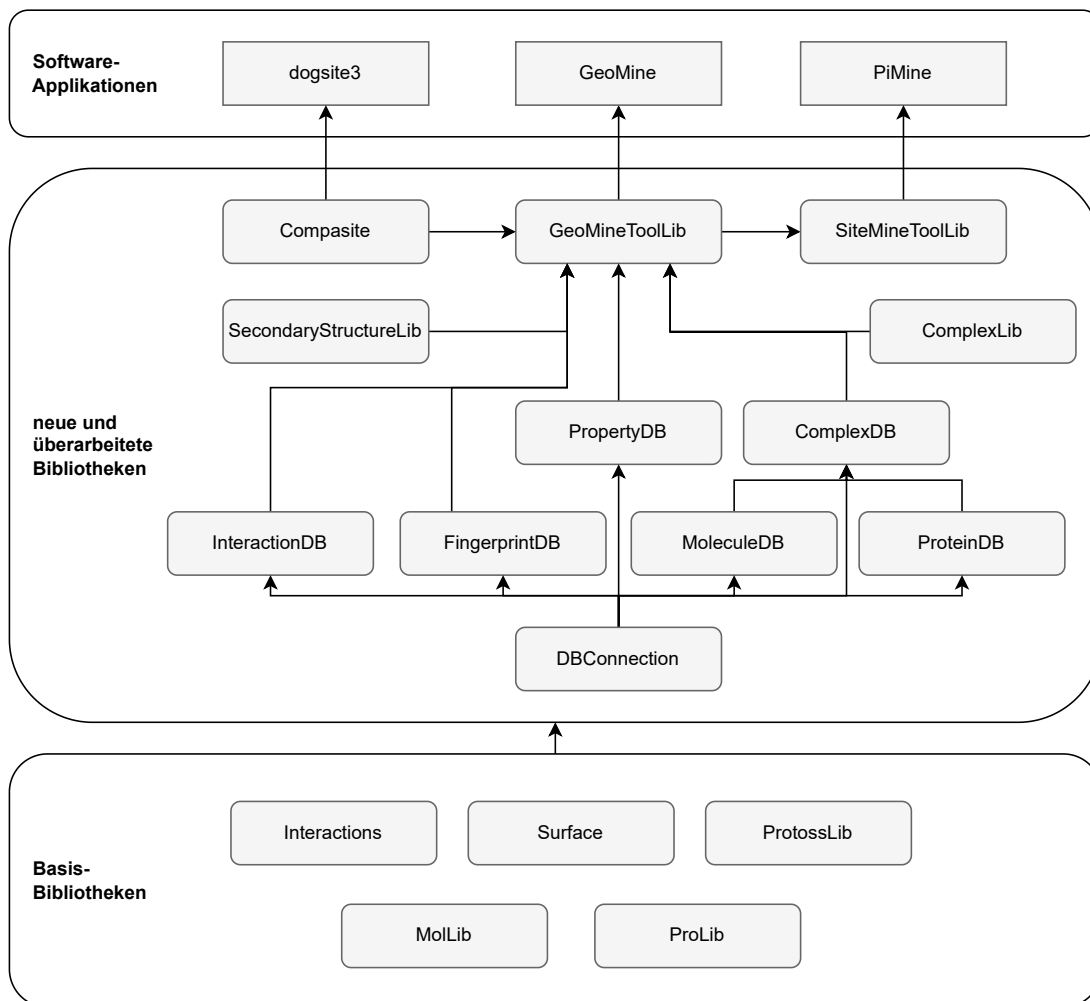
# Software-Architektur

In diesem Kapitel werden die entwickelte Software-Architektur und die darauf basierenden Applikationen vorgestellt. Die einzelnen Komponenten basieren dabei auf Teilen der NAOMI-Plattform [55–57] und sind in der Sprache C++ implementiert. Für eine Erleichterung der Implementierung wird außerdem, wie bereits in anderen Teilen der NAOMI-Plattform, das Qt-Framework verwendet, welches Datenstrukturen beinhaltet, die die Standard C++-Datenstrukturen erweitern oder neu hinzugefügt werden. Hauptfunktionalitäten wie das Einlesen und Übersetzen von Strukturdateien in die interne Darstellung, das Berechnen von molekularen Interaktionen oder die Vorhersage von Bindetaschen ist in Bibliotheken eingeteilt. Dies ermöglicht die Differenzierung und Wiederverwendung von Software und erleichtert die Verwendung in unterschiedlichen Anwendungen. Teilweise gibt es zudem sogenannte Tool- bzw. Anwendungsbibliotheken, die meist bereits eine größere Kombination aus mehreren Bibliotheken darstellen und diese für einen bestimmten Zweck verknüpfen. Diese sind zwar im Kontext einer Applikation entwickelt worden, werden allerdings als eigenständige Bibliotheken angeboten. Applikationen basieren zumeist auf mehreren Bibliotheken und fügen diesen eine Benutzerschnittstelle hinzu.

In Abbildung D.1 werden die Abhängigkeiten der entwickelten Software-Komponenten zu den neuen und erweiterten Bibliotheken dargestellt. Außerdem werden fünf Basis-Bibliotheken der NAOMI-Plattform benannt, die einen zentralen Bestandteil der Bibliotheken ausmachen.

Der Zweck dieses Kapitels ist es, einen Einblick in die getroffenen Designentscheidungen und interne Abhängigkeiten der Bibliotheken zu geben. Das Kapitel ist in zwei Sektionen eingeteilt: 1. Die entwickelten und überarbeiteten internen Bibliotheken mit

ihren Konzepten, Klassen und Abhängigkeiten. 2. Ein Überblick der neuen und erweiterten Applikationen.



**Abbildung D.1:** Abhängigkeiten der entwickelten Software-Komponenten. Pfeile symbolisieren die Verwendung einer Bibliothek in einer anderen Bibliothek oder Applikation. Oben sind die entwickelten Applikationen dargestellt. Darunter folgen neue und überarbeitete Bibliotheken, die in den Applikationen verwendet werden. Unten sind einige Basis-Bibliotheken aufgelistet, die mit Ausnahme der Surface-Bibliothek nicht überarbeitet wurden, aber einen zentralen Bestandteil in den anderen Bibliotheken darstellen. Die Anzahl der Basis-Bibliotheken ist der Übersichtlichkeit halber auf die Relevantesten reduziert.

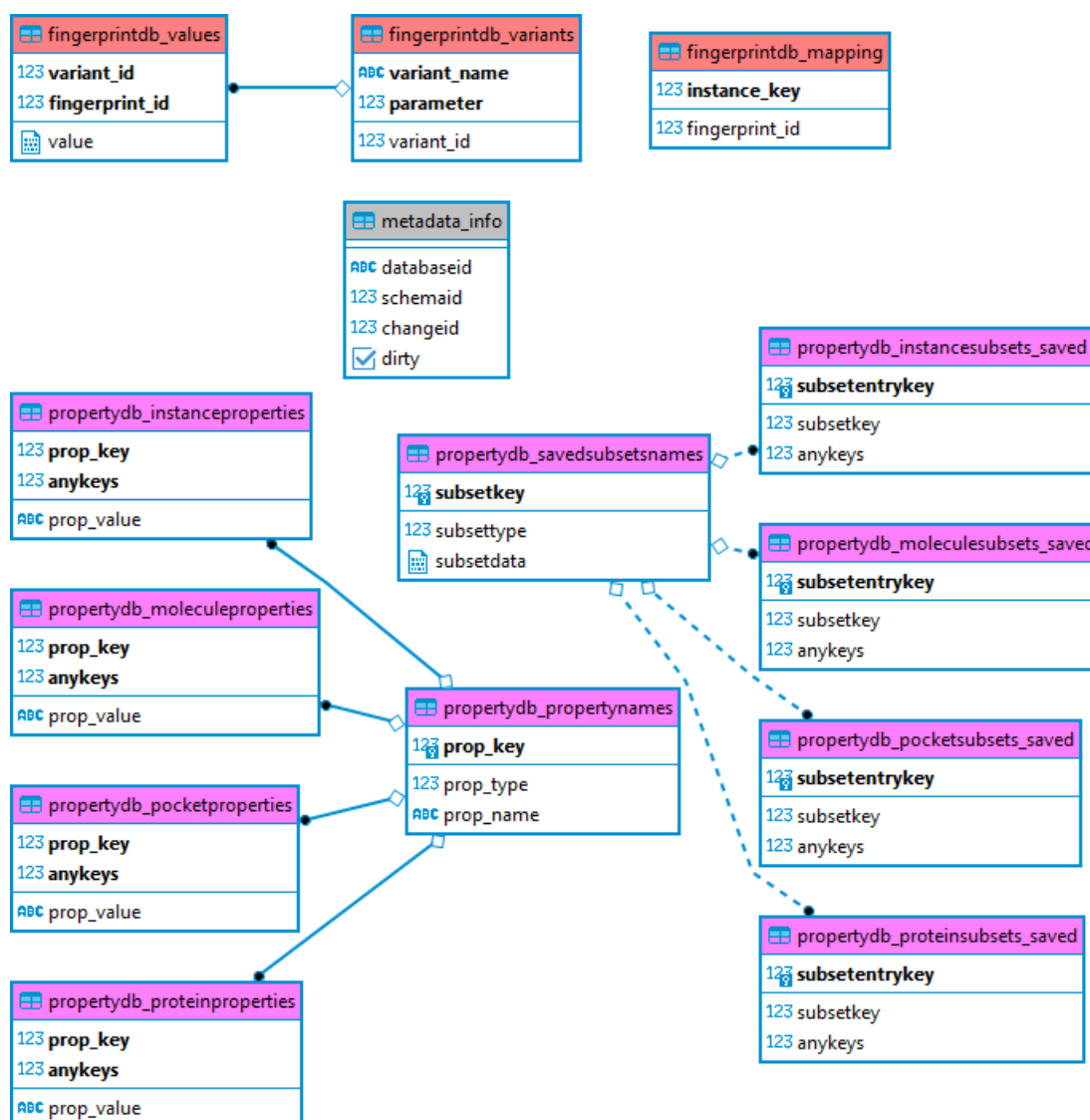
## D.1 Bibliotheken

### DBConnection

Die DBConnection-Bibliothek stellt die Schnittstelle aller Datenbank-Bibliotheken zur sogenannten *QSqlDatabase*-Klasse des Qt-Frameworks dar. Die Datenbank-Bibliotheken beinhalten die Erstellung und Verwaltung von Datenbanktabellen in einem bestimmten Kontext, wie z. B. solche bezüglich der effizienten Abspeicherung von Molekülen. Die *QSqlDatabase*-Klasse wird benötigt, um mit möglichst wenig Aufwand eine Verbindung zu einer bestehenden Datenbank aufzubauen.

Mit der Bibliothek können sowohl SQLite als auch PostgreSQL Datenbanken geöffnet werden. Jede geöffnete Datenbank muss eine sogenannte *metadata\_info*-Tabelle besitzen (siehe Abb. D.2, grau). Existiert diese nicht bereits, wird sie automatisch erstellt. Die Tabelle beinhaltet mehrere Spalten, welche genutzt werden, um die Datenbank-Bibliotheken anzugeben, die Teil der derzeit geöffneten Datenbank sind. Eine *schemaID* gibt darüber hinaus eine Versionsnummer der Bibliothek an, um eine potenzielle Kompatibilität zu anderen Datenbank-Bibliotheken handhaben zu können. Dies ist beispielsweise notwendig, wenn sich in einer Tabelle die Spaltenanzahl ändert und deshalb verwaltende Funktionen umgeschrieben werden mussten. Eine *changeID* ermöglicht es zudem festzustellen, dass Änderungen an dem Inhalt der Datenbank im Kontext einer Datenbank-Bibliothek vorgenommen wurden. Jedes Mal, wenn Daten hinzugefügt, geändert oder gelöscht werden, wird die *changeID* angepasst. Eine letzte Spalte *dirty* wird immer vor dem Ausführen von Datenbank-Anfragen auf *True* bzw. nach dem Ausführen auf *False* gesetzt. Dies erlaubt die Erkennung einer möglichen Datenkompromittierung im Falle eines Verbindungsverlusts.

Neben dem Aufbau der Verbindung zur Datenbank gewährt die DBConnection-Bibliothek ebenfalls einige nützliche Funktionen zum Durchführen von Anfragen. So können beispielsweise mehrere Anfragen mithilfe der *DatabaseTransaction*-Klasse gebündelt an die Datenbank gesendet werden, was den Netzwerk-Overhead reduzieren kann. Auch helfen einige Funktionen z. B. in der Überprüfung, ob eine bestimmte Datenbank-Bibliothek Teil der vorliegenden Datenbank ist oder ob Tabellen mit einem bestimmten Namen registriert sind.



**Abbildung D.2:** Übersicht der Datenbanktabellen der FingerprintDB- (rot), DBConnection- (grau), und PropertyDB-Bibliotheken (blau). Boxen geben Datenbanktabellen an. Diese beinhalten die Schlüsselattribute und sonstigen Attribute voneinander getrennt. Alle eindeutigen Attribute (d.h. ohne Duplikate) sind fettgedruckt hervorgehoben. Durchgezogene und gestrichelte Linien beschreiben jeweils Beziehungen, in denen ein Fremdschlüssel gleichzeitig auch ein bzw. kein Primärschlüssel darstellt.

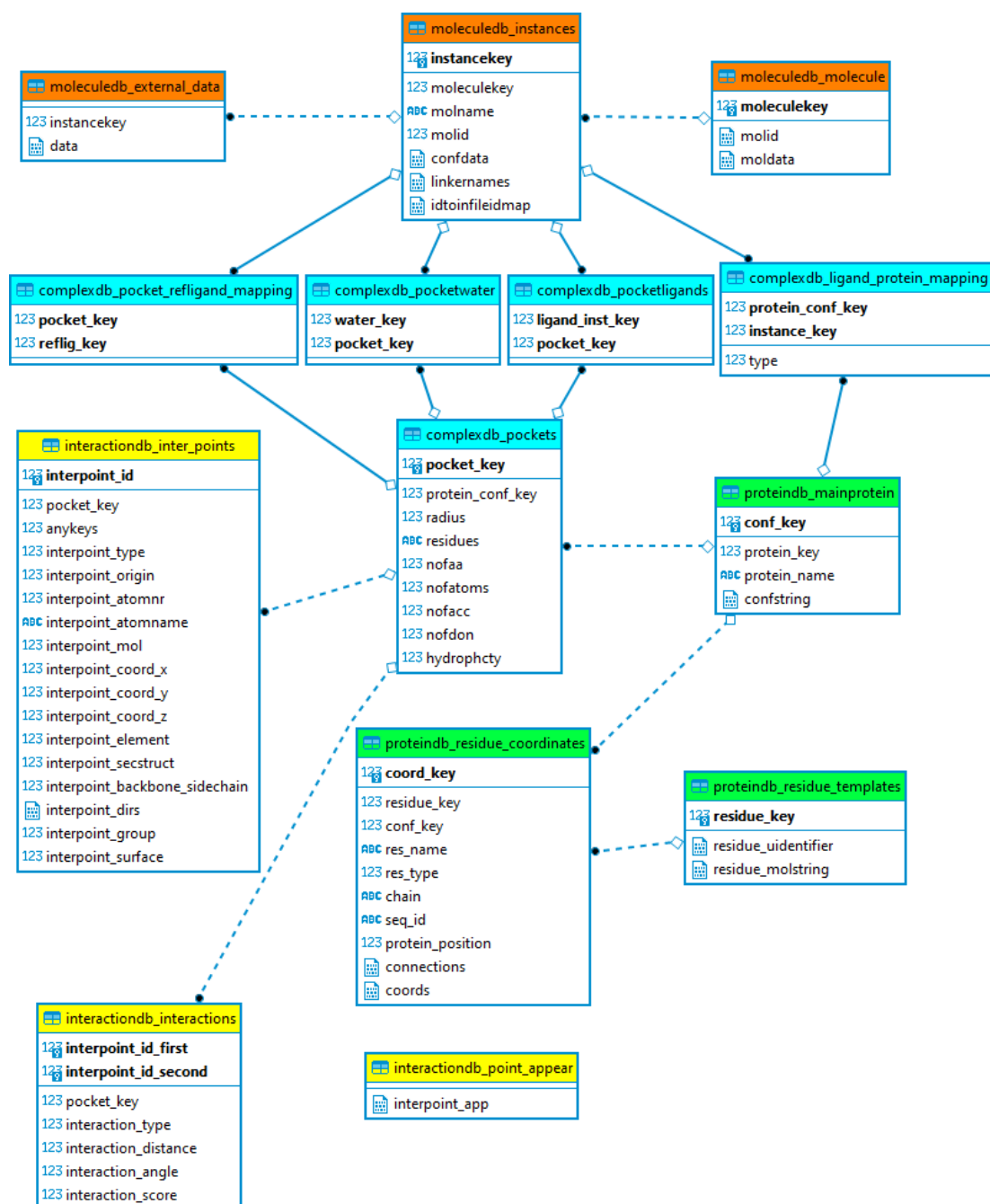
## MoleculeDB

Die MoleculeDB-Bibliothek hat den Zweck der Speicherung von kleinen Molekülen, wie Liganden, Wasser oder Metall-Ionen. Wie in jeder der Datenbank-Bibliotheken sind die Funktionen zur Verwendung der Datenbank-Tabellen in einer *Database*-Klasse definiert. Diese greift hauptsächlich auf Funktionen der *Interface*-Klasse zu, die die Schnittstelle zu den SQL-Datenbank-Befehlen in einer *Queries*-Klasse darstellt.

Die Datenbank ist in drei Tabellen eingeteilt (siehe Abb. D.3, orange): 1. *moleculedb\_molecule* speichert Moleküle mit allgemeinen Daten ohne Atom-Koordinaten; 2. *moleculedb\_instances* speichert Instanzen der Moleküle mit detaillierten Informationen wie Linkern und den Konformationen bzw. Koordinaten und 3. *moleculedb\_external\_data* speichert jegliche weitere Informationen über die Instanzen. Die Einteilung in diese Klassen ermöglicht das Separieren der Moleküle in ihre Topologie und 3D-Koordinaten, sodass Duplikate umgangen werden können. Die Topologie eines Moleküls wird dabei in Form eines sogenannten *MolStrings* gespeichert. Das Vergleichen von Topologien findet mithilfe eines einzigartigen SMILES [128] statt. Der Zugriff auf einzelne Moleküle oder deren Instanzen ist über die Datenbankschlüssel *molecule\_key* und *instance\_key* möglich. Diese können zur eindeutigen Identifikation und Extraktion verwendet werden.

### ProteinDB

Die ProteinDB ist das Gegenstück zur MoleculeDB und speichert Informationen über Proteine wie deren Ketten oder Aminosäuren. Sie besteht ebenfalls aus drei Tabellen (siehe Abb. D.3, grün): 1. *proteindb\_mainprotein* identifiziert jedes Protein mit einem eindeutigen Proteinkonformations-Datenbankschlüssel (*conf\_key*), einem nicht eindeutigen *protein\_key*, einem Namen und dem Pendant zum *MolString* dem *confstring*, welcher die Topologie beschreibt; 2. *proteindb\_residue\_templates* beschreibt Aminosäuren in ihrer Topologie ohne 3D-Koordinaten und 3. *proteindb\_residue\_coordinates* speichert detaillierte Informationen der Aminosäuren, die zur Rekonstruktion der Aminosäuren benötigt werden. Die wichtigsten Datenbankschlüssel dieser Bibliothek sind der *conf\_key*, *protein\_key*, *residue\_key* und *coord\_key*. Während der *conf\_key* eine eindeutige Identifikation eines Proteins mitsamt seiner Konformation beschreiben soll, kann der *protein\_key* verwendet werden, um das gleiche Protein in unterschiedlichen Konformationen zu handhaben. Der *residue\_key* und der *coord\_key* sind hingegen analog zum *molecule\_key* und *instance\_key* der MoleculeDB und beschreiben eine Aminosäuren-Topologie ohne bzw. mit Koordinaten. Mithilfe der verschiedenen Schlüssel können entweder einzelne Aminosäuren oder auch das ganze Protein aus der Datenbank wiederhergestellt werden.



**Abbildung D.3:** Übersicht der Datenbanktabellen der MoleculeDB- (orange), ProteinDB- (grün), ComplexDB- (blau) und InteractionDB-Bibliotheken (gelb). Boxen geben Datenbanktabellen an. Diese beinhalten die Schlüsselattribute und sonstige Attribute voneinander getrennt. Alle eindeutigen Attribute (d.h. ohne Duplikate) sind fettgedruckt hervorgehoben. Durchgezogene und gestrichelte Linien beschreiben jeweils Beziehungen, in denen ein Fremdschlüssel gleichzeitig auch ein bzw. kein Primärschlüssel darstellt.

## ComplexDB

Die ComplexDB verbindet und erweitert die MoleculeDB und die ProteinDB, um Informationen über Protein-Ligand-Komplexe und Taschen zu speichern. Sie besteht aus 4 Tabellen, welche in Abbildung D.3 in Blau dargestellt sind. Die Tabelle *complexdb\_ligand\_protein\_mapping* bildet *conf\_keys* der ProteinDB (hier als *protein\_conf\_keys* implementiert) auf *instance\_keys* der MoleculeDB ab. Diese Zuordnung stellt die kleinen Moleküle und Proteine dar, die einen Protein-Liganden-Komplex bilden. Taschen werden durch einen eindeutigen Schlüssel, dem *pocket\_key*, dargestellt. In der Tabelle *complexdb\_pockets* wird ein *pocket\_key* mit dem *conf\_key* als Fremdschlüssel verweisend auf das Protein, zu dem die Tasche gehört, gespeichert. Zudem wird in einem Textfeld *residues* eine Liste der Aminosäuren in Form ihrer Proteinpositionen angegeben. Falls eine Tasche Referenzliganden beinhaltet, werden der jeweilige *pocket\_key* zusammen mit einem *refligand\_keys*-Fremdschlüssel in der Tabelle *complexdb\_pocket\_refligand\_mapping* gespeichert. Letzterer Fremdschlüssel verweist dabei auf einen *instance\_key* der MoleculeDB. In dem Fall, dass eine Tasche keinen Liganden aufweist, wird kein Eintrag geschrieben. Durch die Angabe dieses Mappings müssen keine Duplikate in der *complexdb\_pockets*-Tabelle gespeichert werden, wenn eine Tasche mehrere Referenzliganden aufweist. Dies war in früheren Versionen der ComplexDB der Fall und wurde im Laufe dieser Arbeit korrigiert. Weiter gibt es zwei Tabellen *complexdb\_pocketligands* und *complexdb\_pocketwater*, die noch einmal getrennt auflisten, welche in der MoleculeDB gespeicherten nicht-Referenz-Molekül-Instanzen in welcher Tasche enthalten sind.

Mithilfe des *conf\_keys* und des *pocket\_keys* können entweder ganze Komplexe oder Proteine mit einzelnen Taschen aus der Datenbank extrahiert werden.

## PropertyDB

Die PropertyDB wird verwendet, um textuelle und numerische Eigenschaften von Molekülen, Molekülinstanzen, Proteinen oder Taschen zu speichern. Alle Tabellen der PropertyDB sind in Abbildung D.2 in violett dargestellt. Eine Eigenschaft wird mit einem eindeutigen *prop\_key* Datenbankschlüssel beschrieben und in derjenigen Tabelle gespeichert, zu der er gehört, z. B. eine Eigenschaft eines Proteins in der *propertydb\_proteinproperties*. Ein *anykeys* Schlüssel beschreibt zudem einen Fremdschlüssel, um eine Referenz auf das Objekt in der Datenbank zu erhalten, welchem die Eigenschaft zugeordnet ist. Der Wert der Eigenschaft ist in einem Textfeld *prop\_value* gespeichert. Eine Tabelle *propertydb\_propertynames* verwaltet die unterschiedlichen Eigenschaften und

speichert zu jedem *prop\_key* einen eindeutigen Namen und den Eigenschaftstyp, z. B. Protein oder Molekülinstanz. Um Ergebnisse von Suchen nachhaltig zu speichern, gibt es außerdem vier Tabellen mit dem Suffix „saved“ und eine *propertydb\_savedsubsetnames* Verwaltungstabelle. Diese werden in keinem der hier entwickelten Tools verwendet.

### InteractionDB

In der InteractionDB werden hauptsächlich molekulare Interaktionen und Punkte von Interesse wie beispielsweise Atome oder Ringzentren einer Bindetasche gespeichert. Die Bibliothek erstellt mindestens 3 Tabellen (siehe Abb. D.3, gelb): 1. *interactiondb\_inter\_points* welche für jeden Punkt von Interesse unter anderem eine *interpoint\_id* als eindeutigen Schlüssel, die zugehörige Tasche über einen *pocket\_key*-Fremdschlüssel, das Molekül oder den Komplex aus dem der Punkt stammt als *anykeys*-Fremdschlüssel und eine Reihe von Informationen wie die Koordinaten, das chemische Element, Sekundärstrukturtyp und die Oberflächenzugänglichkeit speichert; 2. *interpoint\_interactions* welche die *interpoint\_ids* zweier Punkte beinhaltet, zwischen denen eine molekulare Interaktion aufgebaut ist, den *pocket\_key* und beispielsweise den Typ der Interaktion wie z. B. Wasserstoffbrückenbindung oder Metallinteraktion; 3. *interactiondb\_point\_appear* wird verwendet, um die Anzahl an Punktkombinationen als Binärstring zu speichern. So beinhaltet der Binärstring z. B., wie viele Protein- oder Ligand-Wasserstoffbrückendonor-Punkte in der Datenbank enthalten sind. Neben diesen drei Tabellen können bis zu drei weitere Tabellen erstellt werden, die eine Indexstruktur zum Beschleunigen der Suche von Punkten enthalten. Keine dieser Tabellen wird in einer der hier entwickelten Anwendungen verwendet, weshalb diese hier nicht weiter erläutert werden. Eine nähere Beschreibung findet sich in der Dissertation von Therese Inhester [143].

### FingerprintDB

Die FingerprintDB wird dazu verwendet Fingerprints, also Repräsentationen von Molekülen in Form einer Zeichenkette wie dem ECFP Morgan-Fingerprint [212] oder CSFP [213], zu speichern und miteinander zu vergleichen. Eine grafische Darstellung der Tabellen der FingerprintDB ist in Abbildung D.2 in Rot abgebildet. In einer *fingerprintdb\_mapping*-Tabelle wird ein Molekül-Instanz-Fremdschlüssel *instance\_key* mit einem *fingerprint\_id*-Schlüssel verknüpft. Der *fingerprint\_id*-Schlüssel ist hierbei in der Tabelle *fingerprintdb\_values* zu finden und beschreibt mehrere Fingerprint-Varianten. Damit können beispielsweise für das gleiche Molekül mehrere ECFP-Fingerprints abgespeichert werden, die mit unterschiedlichem Radius erstellt wurden. Die einzelnen



Varianten werden durch Parameter und einen Namen in einer *fingerprintdb\_variants*-Tabelle abgespeichert. Zum Vergleichen und Auffinden ähnlicher Moleküle kann somit die Datenbank nach bestimmten Fingerprint-Varianten abgefragt werden. Anschließend kann beispielsweise der Tanimoto-Koeffizient [214] zwischen dem Fingerprint des Moleküls von Interesse und der Datenbank bestimmt werden, sodass alle Molekül-Instanz-Datenbankschlüssel ähnlicher Moleküle zurückgegeben werden.

### SecondaryStructureLib

Die SecondaryStructureLib beinhaltet Funktionalitäten zum Verarbeiten der Sekundärstrukturelemente aus beispielsweise PDB-formatierten Dateien und dem Berechnen von Sekundärstrukturen anhand der NAOMI internen Protein-Datenstruktur. Letztere basiert auf dem DSSP-Algorithmus von Kabsch und Sanders [140]. Dieser definiert die Sekundärstrukturelemente  $\alpha$ -Helix,  $\beta$ -Sheet, 310-Helix,  $\pi$ -Helix, Bridge, Ladder, n-Turn und Bend. Zur Bestimmung der Sekundärstrukturelemente werden als Erstes alle Wasserstoffbrücken zwischen den Aminosäuren der Proteinstruktur berechnet. Aus den Koordinaten der  $C\alpha$ -Atome werden außerdem die Bends berechnet, die Regionen mit einer hohen Krümmung (weniger als  $70^\circ$ ) darstellen. Anschließend werden über die Wasserstoffbrücken die Turns und Bridges berechnet. Ein n-Turn ist dabei zwischen den Aminosäuren  $i$  und  $i+n$  definiert, wenn eine Wasserstoffbrücke zwischen ihnen ausgebildet wird. Eine Bridge besteht zwischen zwei Aminosäuren  $i$  und  $j$ , wenn eines der folgenden Kriterien erfüllt ist, wobei WBB das Wasserstoffbrückennetzwerk beschreibt: (WBB[ $i-1,j$ ] und WBB[ $j,i+1$ ]) oder (WBB[ $j-1,i$ ] und WBB[ $i,j+1$ ]) oder (WBB[ $i,j$ ] und WBB[ $j,i$ ]) oder (WBB[ $i-1,j+1$ ] und WBB[ $j-1,i+1$ ]). Die ersten beiden Kriterien stellen dabei parallele und die letzten beiden antiparallele Bridges dar. Aus den Turns werden anschließend die Helices und aus den Bridges die Ladders bestimmt. Mithilfe der Hierarchie des DSSP-Algorithmus werden die finalen Helices und Sheets aus allen vorher bestimmten Sekundärstrukturelementen erstellt.

Die Bestimmung der Wasserstoffbrückenbindungen wird entweder mithilfe des Kabsch und Sander Ansatzes über elektrostatische Interaktionspotentiale, über die NAOMI-Interaktionsbibliothek oder mit einer Kombination aus beiden durchgeführt.

### ComplexLib

Die ComplexLib beinhaltet primär die Erstellung einer *Complex*-, einer *ActiveSite*- und einer *ProteinProteinInterface*-Datenstruktur, sowie Hilfsfunktionen zur Verwendung dieser. Mit der ersten Datenstruktur werden Proteine, deren Aminosäuren, Metall-Ionen,

Wasser und andere Moleküle zusammengefasst. Die *ActiveSite* beschreibt einen Bereich in einem Komplex, der von Interesse ist, normalerweise eine Protein-Ligand-Bindetasche. Häufig wird eine *ActiveSite* über einen Radius bestimmt. Dabei werden alle Atome des Proteins in einem gewählten Radius um ein Ligand-Molekül herum gefunden und daraus die Datenstruktur erstellt. *ProteinProteinInterface* speichert die Schnittstelle zwischen Proteinen und wird ähnlich zu den *ActiveSites* berechnet. Alle Ketten eines Proteins werden zuerst bestimmt. Anschließend werden alle Paare von Ketten nacheinander betrachtet und für jedes Atom über einen Radius überprüft, ob ein Atom der anderen Kette in der Nähe liegt. Ist dies der Fall, wird das Atom der Datenstruktur hinzugefügt. In der Nähe liegende Moleküle und Wasser werden abschließend ebenfalls bestimmt.

### Compassite

Die Compassite-Bibliothek beinhaltet drei Ansätze zur Gitter-basierten Bindetaschen-vorhersage. Zentraler Bestandteil des Algorithmus ist die *CompassGrid*-Klasse, die das Gitter für das jeweilige Protein berechnet und speichert, sowie die *CompassPocket*-Klasse, die die potenziellen Taschen anhand einer der Methoden bestimmt. Die verschiedenen Methoden sind:

1. DOG\_POC: Der Difference of Gaussians (DoG) Filter bzw. die Differenz von unterschiedlichen Gaußkurven wird verwendet, um Bereiche innerhalb des Proteingitters zu finden, die genug Platz für sphärische Objekte eines bestimmten Radius bzw. Atome beherbergen können.
2. PSP\_POC: Iteration über alle Gitterpunkte mithilfe eines Scanline-Verfahrens zur Bestimmung von Protein-Lösungsmittel-Protein Events.
3. MAN\_POC: Eine Box wird um jedes Ligand-Atom berechnet und die Distanz zwischen den Gitterpunkten der Box zu benachbarten Protein-Atomen überprüft. Die Distanz-Schranke wird durch den Benutzer definiert.

Nach der Bestimmung der Taschen werden ihre Eigenschaften wie die Tiefe oder das Volumen mithilfe der *CompassDescriptor*-Klasse bestimmt. Die Taschen-Datenstruktur kann außerdem als Binärdatei geschrieben bzw. aus einer solchen mit der *CompassPocketSerializer*-Klasse wieder eingelesen werden, um eine erneute Berechnung zu umgehen.

Eine detaillierte Beschreibung des Algorithmus zur Taschenvorhersage findet sich in Kapitel 2.2 und in der Publikation [D1].

## GeoMineToolLib

Die GeoMineToolLib ist eine Bibliothek, die primär für die Applikation GeoMine entwickelt, aber auch für Applikationen wie PiMine verwendet wird. Sie liefert alle Funktionen zur Erstellung und der Durchsuchung der GeoMine-Datenbank. Zentral für die Erstellung der Datenbank ist die *GeoMineDatabase*-Klasse. Diese ermöglicht das Öffnen der Datenbank über die Einbindung der zuvor vorgestellten Datenbank-Bibliotheken und die Prozessierung der Daten, zur Füllung der Datenbank. Zuerst wird die *ComplexReader*-Klasse aufgerufen, um Dateien im PDB- und mmCIF-Format einlesen zu können. Mithilfe einiger Basis-Bibliotheken wie der *ComplexLib* wird in diesem Prozess die *Complex*-Datenstruktur erstellt. Zur Präprozessierung der Komplexe wird anschließend die *ProtossLib*-Bibliothek verwendet. Durch diese werden beispielsweise Wassermoleküle vorhergesagt und Protonierungszustände bestimmt. Mithilfe der *Compassite*-Bibliothek werden dann je nach Anwendungszweck Bindetaschen oder mithilfe der *ComplexLib* Protein-Protein-Interfaces bestimmt. Diese Taschen bzw. Interfaces werden abschließend mithilfe der verschiedenen Datenbank-Bibliotheken verarbeitet und in die GeoMine-Datenbank geschrieben.

Die Filter-Suche innerhalb der GeoMine-Datenbank wird durch die *FilterExecuter*-Klasse durchgeführt. Diese ist in mehrere Suchschritte eingeteilt und startet mit den textuellen und numerischen Filtern zur Reduzierung der Suchmenge, um anschließend die geometrische Suche auszuführen und am Ende eventuelle SMARTS-Muster auszuwerten. Die Einteilung erlaubt dabei eine sukzessive Reduzierung der Suchmenge und ist nach Zeitaufwand und Nutzen sortiert.

## SiteMineToolLib

Der Zweck der SiteMineToolLib ist es, Punkte bzw. Atome und Ringzentren einer Bindetasche oder einer Protein-Protein-Bindestelle auf solche von Interesse zu reduzieren, aus diesen eine Menge an GeoMine-Suchfiltern zu erstellen und basierend auf deren Ergebnissen über eine Bewertungsfunktion ähnliche Taschen bzw. Bindestellen zu bestimmen.

Mithilfe einer *AtomSelector*-Klasse können entweder alle Punkte, nur Heteroatome und Ringzentren oder ausschließlich C $\alpha$ -Kohlenstoffe ausgewählt werden. Mithilfe der *GeometryBuilder*-Klasse werden Tetraeder für alle Punkte innerhalb definierter Distanzen enumeriert und aus diesen mithilfe der *FilterBuilder*-Klasse die GeoMine-Suchfilter erstellt. Die *FilterHandler*-Klasse ruft Funktionen in der GeoMineToolLib

auf und sucht in der Datenbank nach Treffern. Mit der *MatchHandler*-Klasse werden die Treffer abschließend bewertet und so die besten Überlagerungen gespeichert bzw. mithilfe der *StructureAlignment*-Klasse als Datei im PDB-Format exportiert.

## Basis-Bibliotheken

Einige Basis-Bibliotheken bieten Funktionen, die zentraler Bestandteil von oben beschriebenen Bibliotheken darstellen. So wird die Interactions-Bibliothek beispielsweise verwendet, um in der InteractionDB-Bibliothek die molekularen Interaktionen zu bestimmen. Die MolLib und ProLib sind integraler Bestandteil des Einlesens sowie beim Arbeiten mit Molekülen und Proteinen in jeglicher Art. Alle Oberflächenberechnungen in der GeoMineToolLib, Compasite und SiteMineToolLib werden mithilfe der Surface-Bibliothek durchgeführt. Die ProtossLib wird in der GeoMineToolLib genutzt, um die Komplexe zu präprozessieren.

## D.2 Optimierung der Surface-Bibliothek

Im Laufe dieses Promotionsprojekts ist die Ineffizienz der Oberflächenberechnung aufgefallen. Da diese in allen drei entwickelten Programmen verwendet wird und einen erheblichen Einfluss auf die Laufzeit hatte, wurde die Implementierung analysiert und optimiert. Eine Beschreibung der Optimierungen wurde bisher nicht veröffentlicht und wird deshalb im Folgenden zusammen mit einem groben Überblick der Methode gegeben.

Die Oberfläche eines Moleküls mithilfe der *Surface*-Bibliothek kann grob in zwei Schritte eingeteilt werden: 1. Zur Repräsentation der Oberfläche werden sogenannte Oberflächen-Sphären berechnet. Diese werden für jedes Atom erstellt, indem ihr van-der-Waals-Radius mit  $1,4 \text{ \AA}$  als Lösungsmittelzugänglicher Radius (SAS-Radius) addiert wird. Für Kohlenstoffe wird bei Vorhandensein von Wasserstoffen zudem  $0,1 \text{ \AA} * \text{Anzahl der Wasserstoffe}$  zum Radius hinzuaddiert. Nach der Erstellung der Sphären werden die Überlappungen untereinander berechnet und als Liste von Nachbarn abgespeichert. 2. Die eigentliche Moleküloberfläche wird anschließend über alle Sphären berechnet. Hierfür werden alle Sphären nacheinander mit ihren Nachbarn betrachtet. Über die Berechnung von konkaven und konvexen Flächen mithilfe der Sphären wird anschließend die Oberfläche beschrieben.

Die in diesem Promotionsprojekt durchgeführten Optimierungen dieser Berechnung behandeln einzig den 1. Schritt. So wurden zuvor die Überlappungen der Sphären jeweils

über alle Sphären berechnet, d.h. jede Sphäre wurde mit jeder anderen Sphäre verglichen, auch wenn diese räumlich nicht in der Nähe der ersten Sphäre lag. Durch die Berechnung eines Gitters um alle Sphären herum können die Sphären zu naheliegenden Gitterpunkten zugeordnet werden. Um mögliche Nachbarn einer Sphäre zu bestimmen, müssen dadurch nur noch die Sphären benachbarter Gitterpunkte auf Überlappe überprüft werden.

Eine weitere Optimierung betrifft ebenfalls die Berechnung der Überlappe. Da nicht immer die gesamte Proteinoberfläche von Interesse ist, sondern häufig nur ein spezieller Bereich wie eine Bindetasche, müssen nicht alle Überlappe bestimmt werden. Eine Reduzierung auf die Sphären dieses Bereichs resultiert im gleichen Ergebnis, reduziert die Berechnungszeit allerdings abhängig der eingesparten Anzahl an Sphären außerhalb des untersuchten Bereichs. Wird anschließend eine neue Oberflächenberechnung gestartet, die Sphären beinhaltet, von denen bereits Überlappe berechnet wurden, können diese durch eine weitere durchgeführte Optimierung des Algorithmus wiederverwendet werden.

## D.3 Applikationen

Die in diesem Kapitel vorgestellten Bibliotheken ermöglichen die Durchführung der in dieser Arbeit entwickelten Verfahren innerhalb der NAOMI-Plattform. Für eine Interaktionsebene für Nutzer:innen wurden darauf aufbauend Kommandozeilenprogramme entwickelt, die den thematischen Schwerpunkten dieser Arbeit entsprechen. Zur Verwendung des weiterentwickelten DoGSite-Algorithmus wurde die Applikation `dogsite3` entworfen. Diese ermöglicht die Vorhersage von Bindetaschen kleiner Moleküle in Proteinstrukturen. Dies umfasst die Einbeziehung von existierenden Liganden in der Eingabe-Struktur, sowie die reine Suche nach geometrisch günstigen Positionen ohne weiteres Wissen über die 3D-Struktur hinaus. Die GeoMine-Applikation verwendet diese vorhergesagten Bindetaschen und speichert sie mit weiteren Informationen wie textuellen und numerischen Eigenschaften der Taschen, Proteine und Moleküle in einer Datenbank ab. Alle gespeicherten Informationen lassen sich anschließend über das Programm abfragen, um z. B. geometrisch ähnliche oder gleiche Strukturmuster identifizieren. Die PiMine-Applikation verwendet wiederum die Datenbanksuche der GeoMine-Applikation, um zu einer gewählten Protein-Protein-Bindestelle ähnliche Bindestellen zu bestimmen.

Wie bereits erwähnt, fassen die Applikationen im Wesentlichen die Funktionalitäten der Bibliotheken zur Verwendung durch Nutzer:innen zusammen. Sie bestehen daher

primär aus der Kombination einzelner Bibliotheksfunktionen, dem übergeordneten Programmablauf, der Benutzerschnittstellendefinition, der Verarbeitung der Kommandozeilenargumente und der Ausgabe der Ergebnisse.

Auf Grundlage der Kommandozeilenapplikationen wurde außerdem eine Webserveranwendung für das dogsite3-Programm und eine REST-API Schnittstelle für selbiges Programm in einer neuen Version des Webservers <https://proteins.plus> des ZBH - Zentrum für Bioinformatik entwickelt. Ebenso wurde für diese neue Version eine REST-API Schnittstelle für die GeoMine-Applikation implementiert. Eine Schnittstelle für die PiMine-Applikation wurde nicht entwickelt, da die Ähnlichkeitsberechnung auf einer größeren Datenbank wie der vollständigen PDB mehrere Stunden und dadurch ebenfalls viele Ressourcen benötigen würde. Dies geht mit langen Wartezeiten für Nutzer:innen einher, sodass diese Möglichkeit als unattraktiv gewertet worden ist.

## Anhang E

# Bedienung der Software

Die in Kapitel D.3 beschriebenen Software-Applikationen wurden zur Anwendung der dogsite.v3-, GeoMine- und PiMine-Methoden entwickelt. Im Folgenden wird die Bedienung dieser Programme erläutert.

### E.1 dogsite3

Zur Nutzung des dogsite.v3-Algorithmus wurde das Kommandozeilenprogramm dogsite3 erstellt, welches die Vorhersage von Taschen und der Bestimmung ihrer Deskriptoren wie des Volumens oder der Oberfläche ermöglicht. Eine Übersicht der Kommandozeilenargumente des Programms ist in Tabelle E.1 und Tabelle E.2 angegeben. Ausgehend von einer beliebigen makromolekularen Struktur im PDB- oder mmCIF-Format können z. B. die (Sub-)Taschen berechnet, die prozentuale Abdeckung von Liganden durch ihre Tasche bestimmt, die Berechnung auf bestimmte Ketten reduziert und diverse Ausgaben erstellt werden. Auch kann die Parametrisierung des DoGSite-Algorithmus bei Bedarf über Einstellungen gesteuert und so die vorhergesagten Taschen angepasst werden.

Ein Beispielaufruf von dogsite3 zur Berechnung der Taschen mithilfe der in dieser Arbeit optimierten Parameter ist wie folgt. Die Ausgabe der Taschen ist durch diesen Aufruf auf die Deskriptoren beschränkt:

```
./dogsite3 -p protein.pdb
```

Weiter kann beispielsweise der *Ligand-Bias* verwendet und z. B. Taschengitter im CCP4-

**Tabelle E.1:** Kommandozeilenargumente der dogsite3-Applikation zur Vorhersage und Ausgabe von Bindetaschen.

	Flag	Funktion	Wertebereich/ Dateiformat
Allgemein	-h [--help]	Parameterübersicht	
	-v [--verbosity]	Detail-Level der Informationsausgabe während der Algorithmsgschritte	Quiet, Error, Warning, Info, Steps]
	--license	Lizenzregistrierung	ASCII-String
Eingabe	-p [--proteinFile]	Protein- bzw. Makromolekül-Eingabedatei	PDB oder mmCIF
	-l [--ligandFile]	Referenzligand(en)-Eingabedatei	MOL2 oder SDF
	-r [--refLigName]	Referenzligand-Name zur Verwendung eines Liganden der Eingabestruktur	ASCII-String
	-c [--chain]	Makromolekül-Kette zum Eingrenzen der Taschenvorhersage	ASCII-String
Ausgabe	-o [--outputName]	Name der Ausgabedatei	ASCII-String
	-d [--writeDescToFile]	Gib Taschen-Deskriptoren als CSV aus	
	--writeGridPDB	Gib Gitter aller Taschen als PDB aus	
	--writeSiteAtomsPDB	Gib Taschenatome als PDB aus	
	--writeSiteResiduesPDB	Gib Taschen-Reste als PDB aus	
	--writeGridCCP4	Gib Gitter aller Taschen als CCP4 aus	
	--writeSiteResiduesEDF	Gib Taschen-Reste als EDF aus	
	--writePymol	Gib alle Taschen zum Laden in PyMOL aus	
Parameter	-s [--subpocDetect]	Aktiviere Berechnung von Subtaschen	
	-u [--useLigandBias]	Aktiviere den <i>Ligand-Bias</i>	
	--useOnlyInputLigandsForBias	Wende den <i>Ligand-Bias</i> nur auf Eingabe-Moleküle an	

Format geschrieben werden:

```
./dogsit3 -p protein.pdb -u --writeGridCCP4
```



**Tabelle E.2:** Kommandozeilenargumente der dogsite3-Applikation zur Änderung der Parametrisierung.

	Flag	Funktion	Wertebereich/ Dateiformat
Weitere Parameter	--gridDelta	Gitterabstand [Å]	[0,2 - 1,0]
	--contourLevel	Schranke für das Kontur-Level	[2,0 - 4,0]
	--mapCutoff	Schranke für die DoG-Dichte	[-0,1 - 0,1]
	--sigmaEnlarge	Radius der Taschen- erweiterung [Å]	[1,0 - 3,0]
	--ligandSASRatioForAnnotation	max. SAS Verhältnis (gebunden/ungebunden) für <i>Ligand-Bias</i>	(0,0 - 1,0)

Auch können ein oder mehrere Liganden aus einer Datei eingelesen, die prozentuale Abdeckung der Liganden durch Taschen berechnet und gleichzeitig der *Ligand-Bias* verwendet werden. Somit können z. B. Ensembles von Liganden bei der Optimierung der Genauigkeit der Taschenvorhersage helfen:

```
./dogsit3 -p protein.pdb -l ligandEnsemble.sdf -u
```

## E.2 GeoMine

GeoMine ist wie dogsite3 eine Kommandozeilenanwendung. Diese kann sowohl für das Erstellen einer Datenbank als auch zur Suche in dieser verwendet werden. Die verfügbaren Kommandozeilenargumente sind in Tabellen E.3 und E.4 zusammengefasst und in Untergruppen unterteilt. Es gibt die Gruppen der allgemeinen-, PostgreSQL-, Datenbankerstellung-, Datenbanksuche- und Webserver-Parameter.

Datenbanken im SQLite3-Format können z. B. für einen Ordner von Eingabe-Komplexen wie folgt erstellt werden:

```
./geomine -o datenbank.sqlite -d /Pfad/zu/Komplexen
```

PostgreSQL Datenbanken können über das Hinzufügen des Parameters `-p` aktiviert werden. Weiter werden entweder der Benutzername und das Passwort als Umgebungs-

**Tabelle E.3:** Generelle, PostgreSQL spezifische und Datenbankerstellungskommandozeilenargumente der GeoMine-Applikation

	Flag	Funktion	Wertebereich/ Dateiformat
	--license	Lizenzregistrierung	ASCII-String
Generell	-o [--database]	Erstellt/Sucht in dieser Datenbank	SQLite3- oder PostgreSQL-Datenbankname
	-t [--threads]	Thread-Anzahl während der Datenbankerstellung	[x ≥ 1]
	-v [--verbosity]	Detail-Level der Informationsausgabe während der Algorithmusschritte	Quiet, Error, Warning, Info, Steps]
	-h [--help]	Parameterübersicht	
Postgres	-p [--dbIsPostgres]	Aktiviere das Öffnen oder Lesen einer PostgreSQL Datenbank	
	-u [--username]	Umgebungsvariable die den Benutzernamen enthält	ASCII-String
	-k [--password]	Umgebungsvariable, die das Passwort enthält	ASCII-String
	-K [--credentials]	Datei, die den Benutzernamen und das Passwort enthält	ASCII-Datei
	-n [--hostname]	Host IP-Adresse	IP-Adresse
	-P [--port]	Port der Verbindung	[0 - 65535]
Datenbank- erstellung	-i [--input]	Komplex-Eingabedatei	PDB oder mmCIF
	-d [--directory]	Ordner mit Eingabedateien	ASCII-String
	-l [--complexlist]	Datei die eine Liste von Eingabedateipfaden enthält	ASCII-Datei
	-C [--complexPocketsDirName]	Ordner für prozessierte DoGSite-Taschen	ASCII-String
	-r [--remove]	Zu entfernende PDB-Codes	ASCII-String
	-s [--chunkSizeDBCreation]	Anzahl an Komplexen, die gesammelt in die Datenbank geschrieben werden	[x ≥ 1]
	-b [--bindingSiteType]	Wähle die Art der zu berechnenden Taschen	ligandbasierte- oder DoGSite-Taschen
	-m [--maxSASLigandRatio]	Entferne Liganden mit einer hohen Lösungsmittelzugänglichkeit	[0.0 - 1.0]
	--buildProgress	Aktiviere Fortschrittsanzeige	

**Tabelle E.4:** Kommandozeilenargumente der GeoMine-Applikation zur Datenbanksuche und für die Verwendung mit einem Webserver.

	Flag	Funktion	Wertebereich/ Dateiformat
Datenbank- suche	-q [--query]	Suche in der Datenbank anhand einer XML-Anfrage	XML-Datei
	--statistics	Gibt Statistikdatei der Ergebnisse aus	ASCII-String
Webserver	-Q [--webserverPocketData]	Schreibt Taschendaten für Webserverdarstellung	
	-I [--webserverPocketDataPdb]	Pfad zu den Taschendaten der obigen Option	PDB-Datei
	-D [--webserverDBstatus]	Frage generelle Daten wie die Anzahl der Taschen in der Datenbank ab	
	-S [--webserverSearch]	Schreibe die Ergebnisse der Suchanfrage als JSON Datei	
	-O [--webserverOutput]	Pfad/Dateiname für Ausgabe der Taschendaten oder Suchanfrage-Ergebnisse	ASCII-String
	-M [--webserverMol2]	Schreibt Taschenatome der angegebenen Tasche als MOL2 Datei	ASCII-String

variablen übergeben und intern verarbeitet oder mithilfe einer Datei übergeben:

```
./geomine -p -o datenbank -d /Pfad/zu/Komplexen -K dbCredentials.txt
```

Suchen können in einer bereits erstellten Datenbank über den Parameter `-q` durchgeführt werden. Nähere Informationen über die Ergebnisse wie die Häufigkeit der Suchpunkteigenschaften oder Distanzen können darüber hinaus in einer Statistikdatei ausgegeben werden:

```
./geomine -o datenbank.sqlite -q anfrage.xml --statistics
```

Für die Einbindung der GeoMine Datenbanksuche in einen Webservice können beispielsweise die 3D-Ergebnisse einer mithilfe des Webservices erstellten Anfrage im JSON Format abgespeichert werden, um die anschließend im Webservice darstellen zu können. Hierfür muss das Schreiben der JSON Dateien aktiviert und ein Ausgabepfad

angegeben werden:

```
./geomine -o datenbank.sqlite -q anfrage.xml -S -0 suchausgabe.json
```

## E.3 PiMine

**Tabelle E.5:** Kommandozeilenargumente der PiMine-Applikation für generelle Angaben und die Datenbankerstellung.

	Flag	Funktion	Wertebereich/ Dateiformat
	--license	Lizenzregistrierung	ASCII-String
Generell	-o [--database]	Erstellt/Sucht in dieser Datenbank	SQLite3 oder PostgreSQL-Datenbankname
	-v [--verbosity]	Detail-Level der Informationsausgabe während der Algorithmusschritte	Quiet, Error, Warning, Info, Steps]
	-h [--help]	Parameterübersicht	
Datenbank- erstellung	-input	Komplex-Eingabedatei	PDB oder mmCIF
	-d [--directory]	Ordner mit Eingabedateien im PDB oder mmCIF-Format	ASCII-String
	-L [--ppiListFile]	Liste von PDB-Codes und den beiden Ketten-IDs zur Limitierung der zu speichernden Bindestellen. Beispiel: 1A00,A,B	CSV-Datei
	--remove	Zu entfernende PDB-Codes	ASCII-String
	--detectionRadius	Radius für die Bestimmung der Protein-Protein-Bindestellen	[x > 0.0]
	--chunkSizeDBCcreation	Anzahl an Komplexen, die gesammelt in die Datenbank geschrieben werden	[x ≥ 1]
	--buildBioUnits	Aktiviere die Verwendung der biologischen anstatt der asymmetrischen Einheiten	
	-t [--threads]	Anzahl der genutzten Threads	[x ≥ 1]

Mithilfe der PiMine-Anwendung können sowohl Datenbanken erstellt als auch PPI-

**Tabelle E.6:** Kommandozeilenargumente der PiMine-Applikation zum Starten einer Suche und Einstellung der Programm-Ausgabe.

	Flag	Funktion	Wertebereich/ Dateiformat
Suche	-i [--pdbid]	PDB-Code der Anfrage	ASCII-String
	-c [--ppiChainIdentifier]	Kommaseparierte Ketten-IDs der Bindestelle der Anfragestruktur	ASCII-String
	-f [--complexFile]	Komplex-Eingabedatei	PDB oder mmCIF
	-F [--proteinInterfaceFile]	Datei, die eine einzelne Protein-Bindestelle enthält	PDB oder mmCIF
	-r [--searchrestraints]	PDB-Code(s), auf die die Suche beschränkt werden soll	ASCII-String
	-S [--singleChainInterface Only]	Beschränkt die Suche auf das gegebene Interface, bei z.B. der Anfrage „B,A“ wird „A,B“ nicht prozessiert	ASCII-String
	--chunkSizeSearching	Anzahl an Komplexen, in denen gleichzeitig gesucht wird. Zur Reduzierung des benötigten Arbeitsspeichers.	ASCII-String
Ausgabe	--selfskipping	Deaktiviere die Ausgabe von Bewertungen des Eingabe-PPIs	
	--alignDir	Aktiviere das Erstellen von Dateien der berechneten Überlagerungen und verwende den angegebenen Speicherort	ASCII-String
	--maxNofAlignments	Maximale Anzahl der auszugebenden höchstbewerteten Überlagerungen	[x ≥ 1]
	--statistics	Aktiviere die Ausgabe von Bewertungsverteilungen auf der Kommandozeile	
	--times	Aktiviere die Ausgabe der Laufzeit von Algorithmus-Schritten auf der Kommandozeile	

Ähnlichkeitsanalysen zu den dort enthaltenen Bindestellen durchgeführt werden. Die Tabellen E.5, E.6, E.7 und E.8 geben die Kommandozeilenargumente der PiMine-Applikation wieder. Eine Datenbank kann für eine Menge an Dateien im PDB- oder mmCIF-Format erstellt werden. Über weitere Parameter kann außerdem eingestellt werden, dass anstelle der asymmetrischen die biologische Einheit basierend auf den Eingabedateien berechnet werden soll oder der Radius zur Bestimmung der Reste des Interfaces verändert wird. Es kann über eine Liste angegeben werden, welche PPIs in die Datenbank aufgenommen werden. Letzteres ermöglicht es für Multimere nur einzelne,

**Tabelle E.7:** Kommandozeilenargumente der PiMine-Applikation für die Anpassung der Such-Parameter.

	Flag	Funktion	Wertebereich/ Dateiformat
Such- Parameter	-H [--filterHierarchyLevel]	Art des Detail-Levels der Filter	Aminosäuren-Name, Aminosäuren-Typ, chemisches Element oder Atom-Typ
	--twoSidedScoring	Aktiviere die Berechnung von PPI-Ähnlichkeiten anhand der Kombination beider Protein-Interfaces anstelle der nacheinander folgenden Betrachtung der monomeren Interfaces	
	--printAllScores	Aktiviere die Ausgabe aller Scores, z.B. für Anfrage „A,B“ und Ziel „C,D“ die vier Scores: „A“ → „C“, „A“ → „D“, „B“ → „C“, und „B“ → „D“	
	-N [--scoringNorm]	Art der Bewertungsnormierung	Auf Basis der Anfrage, auf Basis der größeren Struktur oder nicht normiert
	-T [--scoringType]	Art der Bewertung	SP, Pharmacophore, Shape]
	-a [--atomSelection]	Für welche Atomkategorien sollen Filter erstellt werden	Alle Atome, wichtige Atome, C $\alpha$ Atome
	--nofilters	Anzahl an zu erstellenden Filtern	[x $\geq$ 1]
	--distTolerance	Toleranz der Abweichung in den Distanzen zwischen zwei erstellten Suchpunkten	[x $\geq$ 0.0]
	--minDist	Minimale Distanz zwischen zwei erstellten Suchpunkten	[0.0 - 13.0]
	--maxDist	Maximale Distanz zwischen zwei erstellten Suchpunkte	[0.0 - 13.0]
	--scoringRadius	Radius der Scoring-Sphäre	[ x > 0.0]

anstatt alle PPIs in die Datenbank aufzunehmen und damit beispielsweise Kristallartefakte auszuschließen. Eine Datenbank, die alle Ketteninterfaces, die auf Grundlage des festgelegten Radius gefunden werden, kann wie folgt erstellt werden:

**Tabelle E.8:** Kommandozeilenargumente der PiMine-Applikation für die Verbindung zu PostgreSQL-Datenbanken.

	Flag	Funktion	Wertebereich/ Dateiformat
Postgres	-p [--dbIsPostgres]	Aktiviere das Öffnen oder Lesen einer PostgreSQL Datenbank	
	-u [--username]	Umgebungsvariable, die den Benutzernamen enthält	ASCII-String
	-k [--password]	Umgebungsvariable, die das Passwort enthält	ASCII-String
	-K [--credentials]	Datei, die den Benutzernamen und das Passwort enthält	ASCII-Datei
	-n [--hostname]	Host IP-Adresse	IP-Adresse
	-P [--port]	Port der Verbindung	[0 - 65535]

```
./PiMine -o datenbank.sqlite -d pfad/zu/komplexen
```

Die Ähnlichkeitssuche in der Datenbank kann auf drei verschiedene Arten durchgeführt werden. Wenn bereits alle PPIs von Interesse in der Datenbank vorhanden sind, kann diese einfach unter Angabe des PDB-Codes der Anfrage und der Ketten-Kennzeichnung der Bindestelle wie folgt angefragt werden:

```
./PiMine -o datenbank.sqlite -i 1GGL -c A,B
```

Falls die Anfragestruktur noch nicht in der Datenbank vorhanden ist und diese auch nicht zu dieser hinzugefügt werden soll, kann die PiMine-Applikation auch PDB oder mmCIF formatierte Dateien lesen. Für die gelesene Struktur wird die Bindestelle anhand der Ketten-Kennzeichnung bestimmt und anschließend die Ähnlichkeit zu den PPIs der Datenbank berechnet:

```
./PiMine -o datenbank.sqlite -f komplex.pdb -c A,B
```

Neben diesen beiden Eingabemodi kann außerdem eine Proteinbindestelle direkt über eine PDB- oder mmCIF-Datei übergeben werden. Die Proteinbindestelle wird hierbei nicht von der PiMine-Applikation berechnet, weshalb dieser Modus genutzt werden kann, um z. B. mit anderen Applikationen vorhergesagte Bindestellen einlesen zu können. Um die Bindestelle korrekt laden zu können, wird außerdem der Herkunftscomplex

der Bindestelle benötigt:

```
./PiMine -o datenbank.sqlite -f komplex.pdb -F bindestelle.pdb
```

Zur Visualisierung der Resultate können Überlagerungen der Anfrage und ihrer gefundenen Treffer ausgegeben werden:

```
./PiMine -o datenbank.sqlite -i 1GGL -c A,B --alignDir ordnerpfad
```

Über verschiedene Suchparameter kann z. B. die Bewertung, die Auswahl der Atome zur Erstellung der Datenbank-Suchanfragen oder die Anzahl der erstellten Suchfilter angepasst werden.

Die Parameter zur Anbindung von PostgreSQL-Datenbanken entsprechen zugunsten der einheitlichen Bedienbarkeit denen der GeoMine-Applikation.

## E.4 ProteinsPlus-Server

dogsit3 und GeoMine sind Teil des am ZBH - Zentrum für Bioinformatik entwickelten ProteinsPlus Servers, welcher unter <https://proteins.plus/> erreichbar ist. Da bereits die erste auf FlexX basierte Version des DoGSite-Algorithmus in den ProteinsPlus Server (*DoGSiteScorer* genannt) integriert war, eine Anpassung der Druggability-Vorhersage allerdings noch in Entwicklung und Teil eines anderen Projektes ist, wurde die neue dogsit3 Anwendung als separater Dienst *DoGSite3* zur Verfügung gestellt. Wie mit dem *DoGSiteScorer* kann auch mit der neuen *DoGSite3*-Benutzerschnittstelle entschieden werden, ob Taschen oder *Subpockets* berechnet werden. Weiterhin kann für einen ausgewählten Liganden die *Coverage* bestimmt und Ketten zur Berücksichtigung in der Vorhersage selektiert werden. Außerdem kann der in Kapitel 2.2 beschriebene *Ligand-Bias* aktiviert werden. Eine mögliche Erweiterung der Anwendung wäre eine Option zum Verwenden eigener hochgeladener Liganden inklusive Koordinaten. Hierdurch könnte die *Coverage*, für diese Liganden bestimmt oder Taschen auf Basis dieser Moleküle mit dem *Ligand-Bias* berechnet werden.

Die GeoMine Webanwendung wurde von Konrad Diedrich entwickelt [D3] und ist nicht Teil des in dieser Arbeit behandelten Promotionsprojekts. Mit dieser können Anfragen erstellt und in einer vorberechneten GeoMine-Datenbank gesucht werden. Dabei können geometrischen Anfragen mithilfe einer Visualisierungsanwendung und einer



grafischen Repräsentation des Proteinkomplexes erstellt und über weitere Einstellungen spezifiziert werden. Textuelle und numerische Eigenschaften werden kategorisiert dargestellt und können hinzugefügt werden. Aus der Anfrage wird im Hintergrund anschließend eine XML-Datei erstellt, die von GeoMine eingelesen und zur Suche verwendet wird. Die 150 besten Ergebnisse sortiert nach dem geringsten RMSD, werden nach der Suche tabellarisch dargestellt und können überlagert im Viewer dargestellt werden.

Um bereits erstellte GeoMine-Anfragen im XML-Format direkt an die GeoMine Anwendung zu übergeben, ohne eine grafische Benutzeroberfläche zu verwenden, kann eine REST-API Schnittstelle genutzt werden. Diese REST-API Schnittstelle wurde im Rahmen dieses Projekts für eine neue Version des ProteinsPlus entwickelt und ist unter <https://proteins.plus/api/v2/> erreichbar. Ebenfalls wurde eine REST-API Schnittstelle für *DoGSite3* entwickelt und ist unter der gleichen Adresse erreichbar.



## Anhang F

# Publikationen der kumulativen Dissertation

### F.1 Binding Site Detection Remastered: Enabling Fast, Robust, and Reliable Binding Site Detection and Descriptor Calculation with DoGSite3

- [D1] **Graef, J.**, Ehrt, C., Rarey, M., Binding Site Detection Remastered: Enabling Fast, Robust, and Reliable Binding Site Detection and Descriptor Calculation with DoGSite3. *J. Chem. Inf. Model.* 63.10 (2023), S. 3128–3137. DOI: 10.1021/acs.jcim.3c00336.

Reprinted with permission from [D1]. Copyright 2023 American Chemical Society.

# Binding Site Detection Remastered: Enabling Fast, Robust, and Reliable Binding Site Detection and Descriptor Calculation with DoGSite3

Joel Graef,<sup>▽</sup> Christiane Ehrt,<sup>▽</sup> and Matthias Rarey\*<sup>▽</sup>

 Cite This: *J. Chem. Inf. Model.* 2023, 63, 3128–3137

 Read Online

ACCESS |

 Metrics & More

 Article Recommendations

 Supporting Information

**ABSTRACT:** Binding site prediction on protein structures is a crucial step in early phase drug discovery whenever experimental or predicted structure models are involved. DoGSite belongs to the widely used tools for this task. It is a grid-based method that uses a Difference-of-Gaussian filter to detect cavities on the protein surface. We recently reimplemented the first version of this method, released in 2010, focusing on improved binding site detection in the presence of ligands and optimized parameters for more robust, reliable, and fast predictions and binding site descriptor calculations. Here, we introduce the new version, DoGSite3, compare it to its predecessor, and re-evaluate DoGSite on published data sets for a large-scale comparative performance evaluation.



## INTRODUCTION

One of the most critical steps in early drug discovery is the analysis of 3D protein structures, many of which are available from the Protein Data Bank (PDB).<sup>1</sup> In addition, structure models of high quality are available today for hundreds of thousands of proteins.<sup>2,3</sup> By identifying binding sites, functions can be predicted and proteins classified. Further, examining binding sites helps to identify novel active and regulatory target binding sites and supports drug design. For the latter, one prerequisite is finding druggable pockets, i.e., sites that can be bound by compounds with drug-like properties as defined by Lipinski's Rule of Five.<sup>4–7</sup> Assessing a binding site's druggability is one of the first steps before exploring it further, for example, by molecular docking or structure-based virtual screening.<sup>8,9</sup> Predicting where a ligand molecule can interact with a protein structure is undoubtedly an essential step in the drug design process. It necessitates the development of very precise in silico algorithms capable of detecting ligand binding sites based on a protein's 3D structure. Especially the calculation of reasonable pocket boundaries is often difficult but critical for downstream processing. Because most druggability prediction methods rely on binding site descriptors,<sup>7</sup> major attention has to be drawn to their calculation.

Automatic prediction methods rapidly identify binding sites with unknown functions. An overview of some established methods with their availability and an outline of their algorithmic approach is given in Table 1. These methods employ different strategies that can be categorized into structure-based, sequence-based, and combination methods.

Structure-based methods rely on structural information such as atom coordinates. Geometry-based methods, as a subgroup of structure-based approaches, locate surface cavities by analyzing the molecular surface of the target protein, such as VOIDOO,<sup>10</sup> SURFNET,<sup>11</sup> PocketPicker,<sup>12</sup> and fpocket.<sup>13</sup> For example, fpocket analyzes the shape using Voronoi tessellation to calculate alpha shapes. A commercial representative is SiteMap.<sup>14,15</sup> It uses a grid-based approach, embedding the protein in a grid and searching for groups of nonprotein grid points. Energy-based methods search for energetically favorable regions on the protein surface. One such method is AutoSite<sup>16</sup> which computes affinity maps, filters out low-affinity points by applying a threshold, and identifies clusters according to local density. The ranking is done using a geometry score. SiteHound<sup>17</sup> calculates so-called Molecular Interaction Fields describing the interaction between protein residues and a probe. Template or knowledge-based methods infer the location of binding sites from known protein templates. COFACTOR,<sup>18</sup> available as a web server and standalone tool, combines different prediction pipelines. One of the pipelines compares the input structure against known pockets contained in the BioLiP library.<sup>19</sup> Another unique approach is based on machine learning.

Received: March 2, 2023

Published: May 2, 2023



Table 1. Exemplary Selection of Tools for Predicting Protein Binding Sites<sup>a</sup>

method	availability	algorithmic approach
AutoSite	standalone	energy-based computes maps and clusters high-affinity points
COACH	web server/standalone	combined methods generates predictions using structure and sequence ligand-binding templates and combining results
COFACTOR	web server/standalone	template-based genetic algorithm
ConCavity	standalone	sequence- and structure-based the protein is embedded in a grid grid points are annotated with sequence conservation values of nearby residues
fpocket	standalone	geometry-based shape analysis by Voronoi tessellation and alpha shape clustering
P2RANK	standalone	structure-based machine learning random forest with feature vectors of physicochemical and geometric properties of protein surface points
PocketPicker	standalone	geometry-based the protein is embedded in a grid nonprotein points are selected by using the buriedness and clustered into pockets
ROBBY	standalone	sequence-based SVM- and random forest-based prediction using evolutionary information
SiteHound	standalone	energy-based calculates Molecular Interaction Fields that describe the interaction between protein residues and a probe
SiteMap	standalone	geometry-based the protein is embedded in a grid nonprotein grid point clusters are searched
SURFNET	standalone	geometry-based the protein is embedded in a grid gap regions are found by using gap spheres which describe gaps between atom pairs of the protein surface
VOIDOO	standalone	geometry-based the protein is embedded in a grid the van der Waals radii of all surface atoms are repeatedly increased, which closes the entrance of cavities properties of detected cavities are then explored

<sup>a</sup>References are provided in the text.

P2RANK<sup>20</sup> is a random forest approach that describes the solvent-accessible protein surface as points. Each point is annotated by a feature vector of physicochemical and geometric properties from its surrounding atoms and residues.

Sequence-based methods are based on the assumption that residues belonging to a ligand binding site are conserved in the evolution of a protein family as they are indispensable for the protein's function. ROBBY<sup>21</sup> employs features derived from the protein sequence. Through a multiple sequence alignment performed by PSI-BLAST,<sup>22</sup> evolutionary information is retrieved from the LigASite<sup>23</sup> database. Then a support vector machine (SVM) and a random forest algorithm are used to predict binding site residues. ConCavity<sup>24</sup> calculates a grid for the protein and annotates its grid points with the sequence conservation values of nearby residues. Hence, sequence and 3D shape analysis based on the grid are combined to detect pockets. Finally, combined methods such as COACH<sup>25</sup> use structure- and sequence-based approaches to complement each other. The input of COACH can be either a structure or a sequence. COACH employs different methods, including COFACTOR<sup>18</sup> and TM-SITE,<sup>25</sup> to train an SVM.

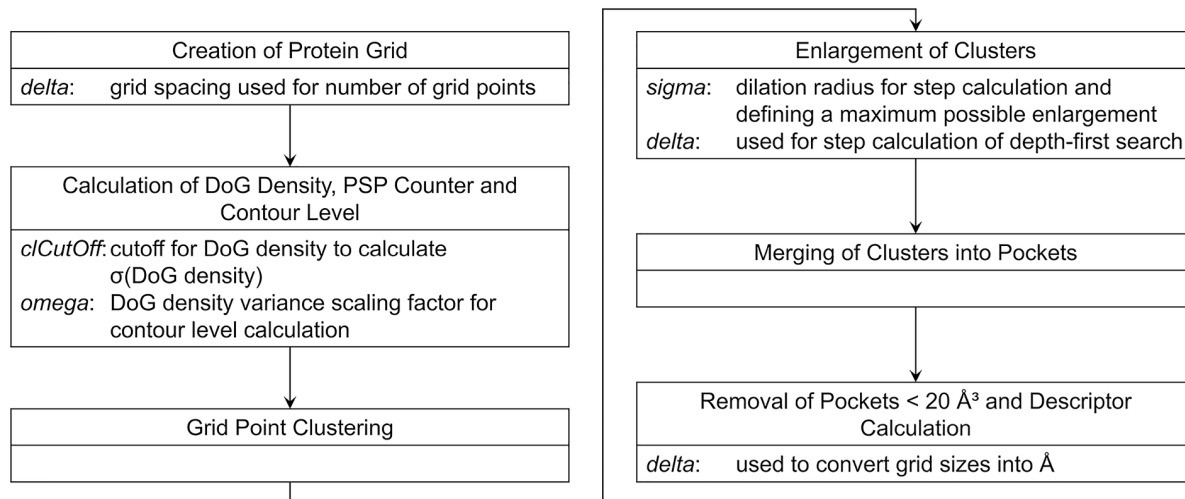
In 2010, we presented the binding site detection algorithm DoGSite<sup>26</sup> and, in 2012, a respective web service.<sup>6</sup> The service includes a druggability estimator and became part of the ProteinsPlus server,<sup>27</sup> which experiences popularity to this day. DoGSite predicts putative binding pockets and subpockets of a protein of interest. After the detection process, it reports geometric and physicochemical binding site properties.

DoGSite correctly predicted binding pockets for 92% of the PDBbind and the scPDB<sup>28–32</sup> data set of 2010. A prediction was assumed to be correct if the geometric center of the largest three pockets lies within 4 Å of any ligand atom. However, this criterion is insufficient as it does not consider the successful detection of reliable binding site boundaries. Solvent-exposed pocket atoms in a 4 Å radius of the pocket center might lead to false positives for inaccurately defined sites. The criterion is also unsuitable for large pockets, which might still cover the ligand. Still, the ligand is situated in a subpocket too distant from the geometric center leading to false negatives. In the meantime, more elaborate measures of prediction success have emerged.<sup>26,33–35</sup>

This work introduces improvements to DoGSite's prediction quality, stability, functionality, and computational speed. While the overall DoGSite algorithm remains the same, we changed the code base and reparametrized the method based on new data. In the following, we introduce this comprehensively remastered version of the DoGSite pocket detection method and compare it to its previous version and a set of alternative binding site prediction techniques.

## METHODS

DoGSite3 is based on the Difference-of-Gaussian (DoG) filter approach published earlier in 2010.<sup>26</sup> For simplicity, we describe the DoGSite3 improvements together with the original DoGSite algorithm.



**Figure 1.** Overview of the DoGSite algorithm and parameters used in each processing step illustrated as boxes.

DoGSite3 calculates potential protein binding pockets of all macromolecular chains, including nucleic acids, except residues with a nonstandard backbone structure contained in a PDB entry. For a graphical overview of the algorithm, see Figure 1. First, the protein is mapped onto a 3D grid. Then, we apply principal component analysis to minimize the required grid size and enable an orientation-independent pocket prediction, which was not available in the previous version. The resulting first three principal axes of the macromolecule are aligned to the  $x$ -,  $y$ -, and  $z$ -axis, respectively. Then, the grid boundaries are calculated by finding the maximum and minimum macromolecule atom coordinates in all directions. In the following, the grid points are created using these coordinates and a predefined grid spacing named  $\delta$ . For very large macromolecules resulting in more than nine million grid points, the grid size is readjusted such that the number of grid points is limited by nine million, leading to a maximum memory consumption of approximately 3.5 GB. The dynamically adjusted grid size (in  $x$ ,  $y$ ,  $z$  direction) is calculated by linear downscaling using the formula  $((\text{gridSize}.x * \text{gridSize}.y * \text{gridSize}.z) / 9,000,000)^{1/3}$ . Next, we iterate over all atoms and annotate each grid point as occupied if the distance to the current atom is smaller than its van der Waals radius plus a tolerance radius that was set to the grid spacing parameter  $\delta$  as opposed to the fixed radius of 0.4 Å in earlier versions of DoGSite to account for the uncertainties of the grid-based approach. Otherwise, the grid point is annotated as a solvent point.

Two annotations of the grid points are required to determine the pockets. The first is the Difference-of-Gaussian filter (DoG value) which favors sphere-like cavities of an atom-like radius of 1.75 Å. It is calculated on all grid points not covered by protein atoms. The mean and variance of all solvent points with a density higher than a DoG density cutoff (called  $clCutOff$ ) are calculated. Using a density variance scaling parameter named  $\omega$ , a contour level is derived as follows:  $\text{contourlevel} = \mu - \omega * \sigma$ . The second annotation is PSP events,<sup>36,37</sup> indicating the buriedness of a grid point. It is calculated using a scanline procedure across all three principal axes and four cube corners. If a PSP event is observed in any direction, we increase the PSP count in all solvent grid points on the scanline between the two protein-occupied points. A

maximum PSP count of seven corresponds to a highly buried grid point, while a minimum of zero indicates a shallow region.

We now iterate over all grid points to cluster the grid points into pockets. If the density value of the respective solvent grid point is below the calculated contour level, we iterate over its 26 neighbors. A depth-first search is started if at least five of the neighbors of this grid point are below the contour level. It runs over the neighboring points until no more solvent points below the contour level are found. This clustering process ends when all grid points have been checked. Finally, all grid points are annotated with their respective cluster number. A second depth-first search is started at each grid point to further enlarge these pockets with solvent points previously not found. This search checks whether in a maximum of  $(\sigma/\delta) + 1$  (with  $\sigma$  being a dilation radius) steps direct neighbors or distant neighbors, e.g., neighbors of neighbors, of the current grid point are (1) a solvent point and (2) not assigned to any cluster so far, (3) its distance to the grid point lies within a threshold of  $\sigma$  squared, and (4) at least two PSP events are present.

After that, we iterate over all clusters and look at each grid point's neighbors again. If two clusters have at least a specific number of neighboring grid points to each other, we merge the two clusters and repeat this step until all points are searched. The number of grid points required is calculated by  $((1.4 \text{ \AA} - rTolerance) / \delta)^2$ , where a default solvent radius of 1.4 Å describes a two-dimensional section between the two clusters. Next, we fill areas between clusters. This filling step is particularly useful when clusters have been merged in the previous step, e.g., if we merged two pockets with only a few grid points next to each other, there might still be solvent points in the proximity not assigned to a pocket. To this end, we iterate over all neighbors of each grid point. If a neighbor is in the same cluster as the current grid point, we scan along that axis within a fixed distance of 2 Å. If this scan finds a solvent point belonging to the same cluster, all points in between will be added to this cluster.

After this pocket assignment process, we sort all pockets by the number of assigned grid points and remove all pockets smaller than 20 Å<sup>3</sup>. Finally, we assign macromolecule atoms to each pocket by checking if the distance between a grid point and an atom is smaller or equal to the sum of the atom's van

der Waals radius plus the tolerance radius plus the cubic grid diagonal:  $dist(gp, atom) \leq vdW + cgd + rTolerance$ , with  $gp$  being the grid point,  $vdW$  the van der Waals radius and  $cgd$  the cubic grid diagonal. Also, we calculate the descriptors such as the volume, surface, depth, ellipsoid main axis ratio, enclosure, and hydrophobicity of the pockets and count and categorize the atoms belonging to each pocket.<sup>6</sup> All calculated descriptors with their explanations can be found in the SI (Table S1).

**Optimizations.** Eleven years have passed since our web service DoGSiteScorer was released. Due to the extensive development in performant programming, we decided to reimplement DoGSiteScorer. As a basis, we use our in-house software library NAOMI.<sup>38,39</sup> NAOMI allows, e.g., easy import of structures in standard formats like PDB and mmCIF, preprocessing steps like the optimization of hydrogen placements, and the export of the processed structures.

In NAOMI, molecules are built based on the local geometry of each atom. It does not rely on definite assignments in structure files to create the molecular model because especially low-resolution structures are often error-prone. Molecules are built in four steps: First, the atoms are described with their element, valence state, and atom type. Their covalent bonds are identified based on the interatomic distances. In the second step, the potential valence states for each atom are defined and scored based on the environment. Third, valence states and associated bond order combinations are generated. In the final step, these are scored to determine the most appropriate molecule representation.<sup>39</sup>

DoGSite3 contains a new mode for identifying ligand-occupied pockets that are difficult to detect by pocket geometry alone. This mode is intended to enhance the detection of binding pocket boundaries that are not considered if only selecting protein atoms in the proximity of ligand atoms. To detect reliable pockets without including highly solvent-exposed regions, we introduced a workflow to consider buried ligand fragments only. Based on the approach of Mahmoud and co-workers,<sup>40</sup> a depth-first search divides all ligands or a specific molecule of interest into nonflexible units. Next, the molecule is cut at each rotatable or exocyclic single bond to generate fragments. Each fragment must contain at least two atoms. The solvent-accessible surface area is then calculated for each fragment in both the bound and unbound states. If the ratio of these values is below a cutoff (0.35 by default), the grid points covered by buried fragments are adjusted so that pocket detection is more likely at these locations. The cutoff is used to ignore fragments and molecules that are highly solvent-exposed, i.e., if some atoms of a molecule are bound to and other atoms are protruding from the protein, only the bound atoms are used for binding pocket detection.

We applied the following changes to the original algorithm: (1) The tolerance radius is set as a fixed parameter of the program and corresponds to the chosen grid spacing. Modifying this parameter might lead to unreliable results in defining binding site atoms. (2) We further improved the pocket atom assignment by additionally considering the grid spacing in the form of the cubic grid diagonal, which was neglected in the former DoGSite version. This adjustment is necessary to account for the discretization of the solvent space by the grid. Thereby, we identify all protein atoms surrounding each pocket grid point. (3) As a major change, we now also include nucleic acid chains as macromolecules in the pocket prediction. (4) The pocket prediction no longer depends on the macromolecule orientation.

For more intuitive and precise pocket descriptors, we adjusted some descriptor calculations. The surface is estimated by calculating the solvent-accessible surface area (SAS) of the pocket atoms<sup>41</sup> and is no longer approximated via the number of pocket grid points. In this context, we optimized the surface calculation for pocket surfaces with respect to the runtime. Also, we now calculate the enclosure by  $1 - (lid/hull)$  instead of  $lid/hull$  since the value thus becomes higher when the lid becomes smaller and the pocket is more enclosed. Additionally, we added the lipophilic surface and the ligand SAS ratio, i.e., the bound ligand solvent accessible surface divided by the unbound ligand solvent accessible surface.

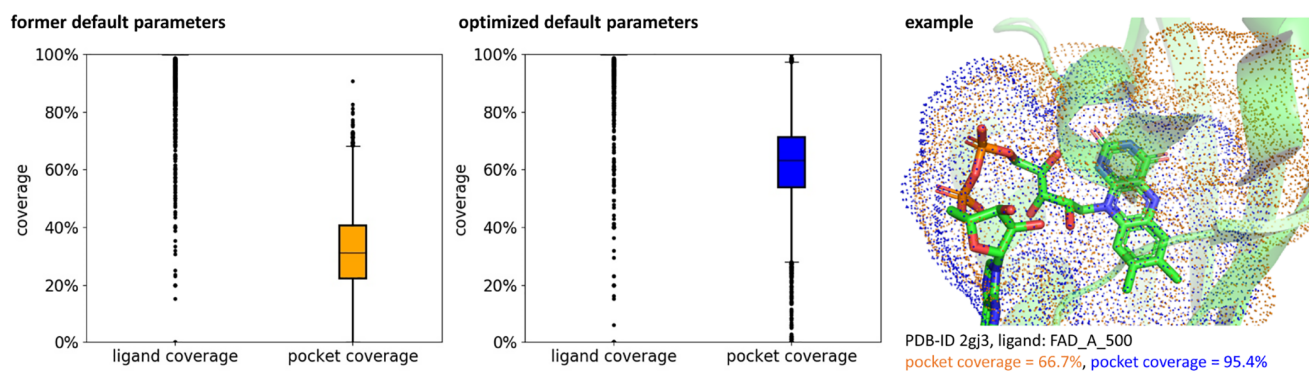
**Parameter Optimization.** The robustness of binding site descriptors is vital to reliable binding site characterization. Therefore, we decided to address this issue through parameter optimization. For this purpose, we used data sets 1 and 2 of the binding site comparison benchmark ProSPECCTs,<sup>42</sup> including various structures of proteins with identical sequences. Data set 2 encompasses various protein structure models from NMR ensembles (17 groups, 329 structures, NMR ensembles), while data set 1 contains multiple structures for 12 different proteins (12 groups, 326 structures, identical proteins).

Based on these data sets, we evaluated the parameters grid spacing ( $\delta$ , default: 0.4 Å), the density variance scaling ( $\omega$ , default: 3.25), the DoG density cutoff ( $c_{CutOff}$ , default: -0.001), and the dilation radius ( $\sigma$ , default: 2.25 Å). Note that for all DoGSite parameters, the written form is used instead of the Greek letter to separate them from statistical values. The grid spacing was varied between 0.4 and 1.0 Å in 0.2 Å steps. The density variance scaling was increased from 2.0 to 4.0 in steps of 0.25, while the DoG density cutoff values were tested from -0.1 to 0.1 in 0.025 steps. Finally, the dilation radius was varied between 1.0 and 3.0 in 0.25 steps. Besides that, default parameters were used for the DoGSite runs. Consequently,  $4 \times 9 \times 9 \times 9 = 2916$  parameter combinations were investigated.

The pockets were sorted according to their ligand coverage.<sup>26</sup> The pockets with the highest ligand coverage were compared to ensure that only corresponding pockets were considered. All parameter combinations ran successfully on the identical proteins, although using 270 parameter combinations did not predict pockets for all structures. For the NMR ensembles, all of the combinations could be evaluated. For 558 parameter combinations, however, no pockets were found for at least one protein structure.

For each parameter combination, we calculated the average of the pocket volume  $V$  standard deviations ( $\sigma(V)$ ) within each group. Furthermore, we calculated the average of the mean ligand ( $\mu(\text{ligcov})$ ) and pocket coverage ( $\mu(\text{poccov})$ ) per group. Finally, we normalized the mean of the standard deviations of all pocket descriptors (volume, surface, lipophilic surface, depth, ellipsoid volume, ellipsoidal main axes  $c/a$ , ellipsoidal main axes  $b/a$ , surface grid points, lid grid points, hull grid points, number of protein heavy atoms, number of solvent-exposed hydrogen bond acceptor atoms, number of solvent-exposed hydrogen bond donor atoms, number of aromatic atoms, hydrophobicity) per group.

**Benchmark of the Pocket Prediction Accuracy.** We analyzed the method's pocket detection capabilities for the ligand-defined scPDB pockets (version as of December 2017)<sup>31,32</sup> and various data sets for benchmarking binding site identification tools<sup>34</sup> to evaluate the optimized parameter



**Figure 2.** Analysis of the ligand and pocket coverage for ligand-defined binding sites in the scPDB with the former default parameters of DoGSite (orange) and the more robust and reliable default parameters of DoGSite3 (blue). Both ligand and pocket coverage are represented as box plots. The new feature of DoGSite3 for biasing the grid annotation by reference ligands was applied for the analysis. Therefore, the lower and upper quartiles and whiskers for the ligand coverage lie at 100%. The structure on the right represents an example of a pocket with low pocket coverage with the original DoGSite parameters and a much higher one with the optimized DoGSite3 default parameters (figure generated with PyMOL Molecular Graphics System, version 2.3.0).<sup>51</sup>

sets. The data sets include structures derived from the PDBbind core set,<sup>43</sup> cryptic binding sites as provided by the work of Cimermancic et al.,<sup>44</sup> and allosteric binding sites from the ASBench Core Diversity Set.<sup>45</sup> For the assessment of the pocket prediction accuracy, we used the residue-based prediction accuracy as defined by the relative residue overlap (RRO)<sup>34</sup> between the residues inside a 5 Å distance of the cavity-defining ligand (cavity-defining residues, CDR) and the predicted binding site residues by the pocket detection algorithm (PR) according to Equation 1.

$$\text{RRO} = \frac{|\text{CDR} \cap \text{PR}|}{|\text{CDR} \cup \text{PR}|} \quad (1)$$

An RRO of 0 indicates that none of the predicted residues is part of the ligand-defined cavity. In contrast, an RRO of 1 denotes a complete correspondence between predicted and cavity-defining residues. A prediction was assumed to be successful for an RRO of at least 0.5.

**Runtime Analysis.** Runtime calculations were performed for the data set of identical proteins on a PC equipped with an Intel i5-8500 (3.0 GHz) processor, 16 GB of main memory, and a Toshiba KXG50ZNV256G solid-state drive (256 GB, model NVMe) with an xfs file system.

## RESULTS AND DISCUSSION

**Including Ligand Bias to Obtain Ligand-Based Binding Site Descriptors.** As can be learned from various studies<sup>34,35</sup> and later in this work, some ligand-occupied pockets cannot be correctly predicted by any binding site detection algorithm. An unsuccessful pocket identification is especially obstructive if the druggability of specific binding sites is to be analyzed or the detection method is used to define the dimensions of, e.g., a fragment-bound binding site for molecular docking and fragment growing. The same holds for developing binding site databases for automated pocket comparisons, e.g., with GeoMine.<sup>46,47</sup> In these cases, DoGSite3 enables the user to annotate the DoGSite grid based on given reference ligands, ensuring the detection of pockets in the proximity of the reference ligands and allowing for analyses of binding sites that are difficult to detect. If this modification is applied for the druggable binding sites as stored in the scPDB, 99.0% of the binding sites are included with a ligand coverage

of at least 80% as compared to 76.4% if this ligand bias option is not enabled.

The possibility to bias ligand binding sites by reference ligands provided in SDF files enables reliable binding site definitions even if the original algorithm did not detect some of the interacting residues as part of the binding site (SI, Figure S1). It has to be noted that this detection is not restricted to the ligand-biased grid points but that the pocket is extended based on the DoGSite algorithm. Therefore, this option represents a significant advantage compared to ligand radius-based pockets as the pocket is enlarged toward regions that might be additionally addressed, which is highly useful in the context of, for example, structure-based virtual screening. Furthermore, the user can derive binding site definitions biased toward ligand ensembles from similar binding sites, e.g., as retrieved by SIENA,<sup>48</sup> for molecular docking studies.

**Ligand and Pocket Coverage Analysis.** One of the major challenges in evaluating the quality of binding site prediction tools is the definition of binding site boundaries. Usually, ligand radius-based approaches are used to define binding site extents. However, this method is not always reliable, e.g., in the case of protein binding sites in complex with fragments binding to subpockets or small recesses within a larger binding site.

For a ligand-occupied pocket detected with the DoGSite algorithm, the ligand coverage describes the percentage of ligand atoms covered by a predicted pocket. In analogy, the pocket coverage represents the percentage of pocket grid points in the proximity of ligand atoms. We first explored the herein-analyzed data sets to investigate the distribution of ligand and pocket coverage across a large set of binding sites. We predicted the extent of ligand-occupied binding sites with DoGSite with default parameters using the novel bias to annotate grid points based on a given ligand as binding site points. Since we enforce that all ligand fragments that are not solvent-exposed will be included in the pocket volume, we can thereby obtain an assessment of the average pocket coverage for the druggable binding sites in the scPDB (Figure 2). According to these results, the average acceptable pocket coverage for well-defined binding sites is rarely higher than 70% but varies between 20% and 40% depending on the type of ligand. For 89 structures, the ligand coverage is below 70%. The lower ligand coverage, despite the ligand bias, can be



attributed to solvent-exposed ligand atoms that are not considered for biasing but for ligand coverage calculations. The same analysis (biased and unbiased) was performed for the data sets of identical proteins and NMR ensembles<sup>42</sup> (SI, Figure S2).

As can be seen, DoGSite, with the original default settings, identifies pockets much larger than the volume of the occupying ligand, as the example in Figure 2 also indicates. Therefore, the present study's primary challenge was a more accurate binding site boundary prediction enabling a reasonable trade-off between maximum ligand and pocket coverage.

**Descriptor Robustness Optimization.** Attempting to improve the reliability of DoGSite, we evaluated the impact of the parameters grid spacing (*delta*, default: 0.4 Å), the density variance scaling (*omega*, default: 3.25), the DoG density cutoff (*dCutOff*, default: -0.001), and the dilation radius (*sigma*, default: 2.25 Å) on the method's performance. The former default parameters were analyzed with a version of DoGSite3 that does not rearrange the protein structures before the calculations, uses an older surface calculation implementation,<sup>41</sup> and performs the original merging procedure as it was optimized for this version of the tool. To enable the reliable detection of binding sites independent of minor structural fluctuations, we decided to focus first on the robustness of the binding site descriptors. These binding site characteristics were shown to fluctuate considerably in earlier studies.<sup>7</sup> These results could be reproduced in this study using the former default parameters and parameter combinations discussed in the following (SI, Figures S3–S8). Note that the ligand information was not included in the binding site prediction, but only the pockets with the highest ligand coverage were investigated for this part of the analysis.

Investigations of the separate parameters and their impact on the standard deviation of the pocket volumes per group of structures revealed a negligible influence of the grid spacing on the robust prediction of the binding sites in the data set of identical proteins, as only slightly smaller volume variations were observed for parameter combinations with a larger grid spacing. The standard deviation of the volume per group of identical proteins with increasing grid spacing for selected parameter combinations can be found in the SI (Figure S9).

In contrast, the contour level cutoff follows more distinct trends. An increase in the cutoff leads to more robust predictions with respect to the pocket volume according to its standard deviation in the groups of identical proteins. This trend persists for nearly all exemplary parameter combinations (SI, Figure S10). However, an opposite relation was observed for the DoG density cutoff. The higher this cutoff, the more the volumes of the predicted pockets in the groups of identical proteins vary (SI, Figure S11). Generally, a cutoff above 0 leads to a high variance in the pocket volume. Less prominent but apparent is a general trend in the dilation radius: the larger this radius, the higher the volume standard deviations within the groups of identical proteins (SI, Figure S12).

Similar trends were observed for the data set of NMR ensembles (SI, Figures S13–S16).

Next, we set out to identify the most reliable parameter combinations (see Table 2 for an overview). For both data sets, we chose the parameter combinations with the lowest sum of normalized mean descriptor standard deviations (see Methods section) that fulfill the criteria  $\sigma(V)$  of less than 110 Å<sup>3</sup>, a  $\mu(\text{ligcov})$  of more than 80%, and a  $\mu(\text{poccov})$  of

**Table 2. DoGSite3 Parameter Sets Investigated in Detail**

parameters	<i>delta</i> [Å]	<i>omega</i>	<i>dCutOff</i>	<i>sigma</i> [Å]
former default	0.4	3.25	-0.001	2.25
parameter set 1	0.8	2.5	-0.1	1.0
parameter set 2	0.8	3.0	-0.05	1.0
parameter set 3	0.4	3.75	-0.075	1.0
parameter set 4	0.4	3.25	-0.05	1.0
parameter set 5	0.6	4.0	-0.025	1.25

more than 40%. The cutoff for the volume standard deviations approximately describes the volume necessary to accommodate a molecule of cyclohexanol<sup>49</sup> and is chosen to reduce the number of considered combinations to about 200. To evaluate whether this selection of parameter combinations leads to an improvement in the robustness considering both data sets, we compared the  $\sigma(V)$ , the  $\mu(\text{ligcov})$ , and the  $\mu(\text{poccov})$  to those obtained with default parameters (Table 3). Parameter set 1 (see Table 2) performed best for the identical proteins (rank 328 in the parameter combinations sorted according to the sum of normalized mean descriptor standard deviations for the identical proteins). For the NMR ensembles, parameter set 2 led to the lowest standard deviations, nonetheless fulfilling all criteria (rank 6 in the parameter combinations sorted according to the sum of normalized mean descriptor standard deviations for the NMR ensembles). Besides the much higher robustness of the predicted pocket volume with the new parameter sets, we find that, despite the promising ligand coverage using the former default parameter set, the pocket coverage is low, suggesting artificially large predicted pockets. This situation considerably improves using the new parameter combinations, retaining a convincing mean ligand coverage. Finally, we could observe a significant decrease in the number of predicted pockets. In the following sections, we will show that this does not impact the prediction success of DoGSite3, hinting at a high number of irrelevant predicted sites with the former versions.

As one could argue that the consideration of the standard deviations of pocket volume, ligand coverage, and pocket coverage alone is not sufficient to guarantee a robust descriptor calculation, we also selected the parameter sets that led to the lowest sum of normalized mean descriptor standard deviations for both data sets leading to two other potential parameter sets (see parameter sets 3 and 4 in Table 2). Finally, the last parameter set was derived by ranking according to the sum of normalized mean descriptor standard deviations but adjusting the thresholds for  $\mu(\text{ligcov})$  ligand coverage and  $\mu(\text{poccov})$  pocket coverage to at least 60% and 30%, respectively. This selection led to parameter set 5 (see Table 2) for the data set of identical proteins. For the data set of NMR structures, the same parameter combination as already found in parameter set 4 was obtained. The box plots of the physicochemical and geometric descriptors for the default parameters and the new parameter combinations can be found in the SI (identical proteins: Figures S3–S5, NMR ensembles: Figures S6–S8).

For an external validation of these five parameter combinations, we use DoGSite3 to predict the druggable binding sites in the scPDB and monitored the ligand and pocket coverage (Table 4) for a final choice of parameter sets with the most promising detection success concerning ligand and pocket coverage. In addition, we analyzed the runtimes on the data set of identical proteins for the default and new parameter combinations.

**Table 3. Comparison of the Impact of Different Parameter Combinations on the Robustness of the Volume, the Ligand Coverage, the Pocket Coverage, and Number of Predicted Pockets for Structural Ensembles in the Data Sets of Identical Proteins and NMR Ensembles**

parameters	data set	$\sigma(V)$ [ $\text{\AA}^3$ ]	$\mu(\text{ligcov})$ (%)	$\mu(\text{poccov})$ (%)	no. pockets
former default	identical proteins	168.0	87.9	31.8	6221
	NMR ensembles	238.7	81.9	42.1	3474
parameter set 1	identical proteins	52.6	83.2	65.9	2173
	NMR ensembles	37.9	67.1	77.8	1305
parameter set 2	identical proteins	77.7	88.9	55.4	3993
	NMR ensembles	47.3	80.4	74.9	2576

**Table 4. Performance of the Former Default Parameters and the Five Selected New DoGSite Parameter Sets on Predicting the Druggable Binding Sites from the scPDB (17589 Binding Sites) and Runtimes for the Data Set of Identical Proteins (326 PDB Files)<sup>a</sup>**

parameters	mean ligand coverage (%)	mean pocket coverage (%)	percentage (ligcov $\geq$ 80%, poccov $\geq$ 40%)	runtime [s] <sup>b</sup>
former default	88.2	41.1	35.5	5067.1 $\pm$ 68.4
parameter set 1	79.2	80.0	55.8	357.1 $\pm$ 10.1
parameter set 2	87.9	71.7	66.6	385.0 $\pm$ 1.7
parameter set 3	30.7	50.3	7.8	4193.9 $\pm$ 55.1
parameter set 4	65.8	64.6	34.3	4437.1 $\pm$ 58.3
parameter set 5	71.1	65.3	44.7	943.0 $\pm$ 7.5

<sup>a</sup>The runtimes were obtained from five independent DoGSite runs. <sup>b</sup>mean  $\pm$  standard deviation.

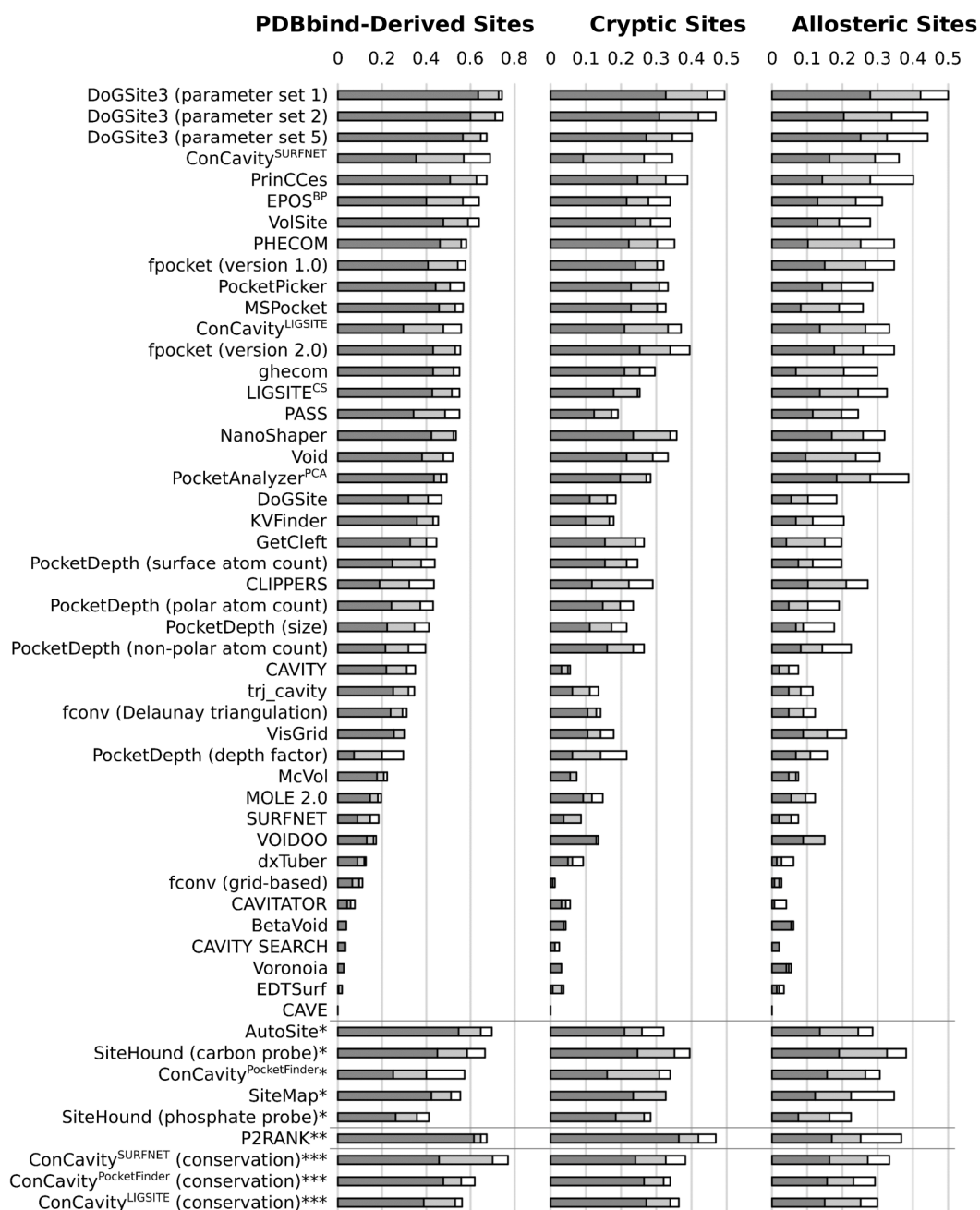
From this analysis, it can be concluded that parameter sets 1, 2, and 5 led to the most convincing results (mean ligand coverage of >70% and >40% of the detected binding sites with a ligand coverage and a pocket coverage of at least 80% and 40%, respectively). Therefore, we concluded that these combinations are most suitable for robust and reliable binding site predictions with DoGSite3. For these parameters, there is a well-balanced trade-off between optimum ligand and pocket coverage, as illustrated by the success rate regarding the retrieval of pockets with a ligand coverage of at least 80% and a pocket coverage of at least 40%. These findings are in accordance with the previously discussed trends for exemplary parameter combinations. A higher grid spacing leads to more robust results that are not biased by local variations in the DoG values. A low DoG density cutoff leads to more reliable results as it emphasizes points with a more sphere-like character. For higher values of this parameter, we observe improved robustness with an increasing density variance scaling parameter *omega* (as observable for parameter combinations 2 and 5). The clear finding that a low dilation radius leads to lower descriptor standard deviations is explicable by a more restrictive extension of pockets. In contrast, we observe an insufficient retrieval of scPDB binding sites for parameter sets 3 and 4, even worse than with the former default parameter sets. Both sets are characterized by a lower grid spacing rendering the method sensitive toward local differences in the DoG density values. Parameter set 1 is the fastest of all promising parameter sets and shows the most convincing balance between the pocket and ligand coverage.

**Reevaluation of Pocket Prediction Performance for the Optimized Parameter Sets.** The remaining best-performing three parameter sets (parameter sets 1, 2, and 5 (Table 2)) were further evaluated for the reliable prediction of binding sites. For this evaluation, we used three data sets from a previous benchmark study on pocket detection algorithms.<sup>34</sup> These data sets include pockets extracted from the PDBbind,<sup>43</sup> cryptic binding sites,<sup>45</sup> and allosteric binding sites.<sup>45</sup> Analyzing the prediction success of the selected parameter combinations

for these data sets, we find that all parameter combinations lead to an increase in prediction success compared to the former default parameters (SI, Table S2). However, parameter set 1 is slightly more successful than sets 2 and 5 for the cryptic and allosteric pockets. Furthermore, the percentage of successfully detected pockets in the PDBbind-derived sites is only marginally higher for parameter set 2 than for parameter set 1.

Based on this earlier benchmark study, we finally compared DoGSite3 with the new parameter combinations to 40 alternative geometry-based pocket detection algorithms and selected alternative tools. The results of the analysis are depicted in Figure 3. Regarding the percentage of successfully predicted pockets in the highest-scoring three pockets for geometry-based binding site prediction algorithms, DoGSite3 with parameter set 1 is on rank 1 (formerly rank 18), rank 1 (formerly rank 25), and rank 1 (formerly rank 25) for the PDBbind-derived data set, the data set of cryptic sites and the data set of allosteric sites, respectively. The DoGSite3 performance considerably improved compared to its previous versions (SI, Table S2) and shows that DoGSite3 is a fast and reliable geometry-based binding site detection method. Its prediction success is comparable to that of the analyzed knowledge-based prediction methods, which are substantially dependent on pre-existing similar protein–ligand complex structures.<sup>50</sup>

A visual inspection of the pocket and ligand coverage for the scPDB (Figure 2) and the data sets of identical proteins and NMR ensembles (SI, Figure S2 and Figure S3) provides evidence that the new version DoGSite3 with the optimized parameter set 1, besides its more robust descriptor calculation, predicts the most accurate and reliable binding site boundaries (improved pocket coverage). Furthermore, parameter set 1 is the best-performing set for binding site identification. Therefore, parameter set 1 ensures the overall most reliable predictions regarding descriptor robustness, ligand and pocket coverage, and prediction accuracy and can be recommended for DoGSite3 with a runtime of 0.5–4 s per structure.



**Figure 3.** Binding site prediction accuracy of DoGSite3 compared to alternative methods (see SI, Table S2 for the corresponding references). Given is the percentage of successfully predicted pockets, i.e., predicted pockets with an RRO of at least 0.5, for the top1 (dark gray), top2 (light gray), and top3 (white) best-scored pockets. The original DoGSite algorithm was evaluated with an academically licensed version downloaded from the web site of BioSolveIT (DoGSiteScorer 2.0.0). \*energy-based methods, \*\*knowledge-based methods, \*\*\*combined methods.

## CONCLUSION

In structure-based drug design, binding pockets play a crucial role. In the absence of cocrystallized ligands, the most accurate prediction models are needed. In the case of protein–ligand complex structures, the model building process is either completely focused on the ligand or ignores it at all, which is inappropriate. This results in a loss of knowledge for making precise predictions. With DoGSite3, we provide a binding pocket prediction that is significantly more accurate and robust than its predecessor and provides a valuable new feature enabling the optional consideration of existing ligands in the

grid calculations. Furthermore, we developed a considerably improved and stable prediction tool by refining the pocket merging strategy and enabling a unique grid orientation. Finally, the newly derived parameter combination enables more robust and reliable binding site detection with a substantially lower runtime (approx. ten times faster than the previous version with former default parameters).

The web interface from the predecessor of DoGSite3 has been extended with the new functionality. Thereby, we retain the well-known and established easy-to-use web interface.

We present DoGSite3 as a ready-to-use tool for reliable binding site prediction, robust binding site boundary definition, descriptor calculation, and the calculation of ligand-biased difficult-to-detect binding sites with considerably improved runtimes. Having said this, we hope that future screening efforts and binding site druggability prediction and classification efforts will benefit from the improved DoGSite3 capabilities.

## ■ ASSOCIATED CONTENT

### Data Availability Statement

All data used is generated from structures of the Protein Data Bank, which is freely available here.<sup>1</sup> The training data sets are available on GitHub under this link: <https://github.com/rarelab/DoGSite3-Datasets>. The benchmark data sets are part of the ProSPECCTs sets and were used unaltered. They are available under this link: <http://www.ewit.ccb.tu-dortmund.de/ag-koch/prospeccts/>. DoGSite3 is available as a free web service that can be accessed using the link <https://proteins.plus>. In addition, a standalone tool of DoGSite3 is available as part of the NAOMI ChemBio Suite, which is free for academic use as well as licensable for commercial use.

### SI Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jcim.3c00336>.

Ligand-biased DoGSite pockets (Figure S1), analysis of the ligand and pocket coverage of default and new parameters for the data sets identical proteins and NMR ensembles (Figure S2), robustness of the geometric and physicochemical descriptors for the data sets of identical proteins (Figures S3–S5) and NMR ensembles (Figures S6–S8), impact of the DoGSite parameters on the volume standard deviations for the data sets of identical proteins (Figures S9–S12) and NMR ensembles (Figures S13–S16), abbreviations and explanations of calculated descriptors with DoGSite3 (Table S1), binding site detection hit rates of DoGSite3 and alternative binding site identification tools for three data sets (Table S2), and overview of the analyzed binding site prediction tools (Table S3) (PDF)

## ■ AUTHOR INFORMATION

### Corresponding Author

Matthias Rarey – Universität Hamburg, ZBH - Center for Bioinformatics, 20146 Hamburg, Germany; [orcid.org/0000-0002-9553-6531](https://orcid.org/0000-0002-9553-6531); Email: [matthias.rarey@uni-hamburg.de](mailto:matthias.rarey@uni-hamburg.de)

### Authors

Joel Graef – Universität Hamburg, ZBH - Center for Bioinformatics, 20146 Hamburg, Germany; [orcid.org/0000-0001-8327-4936](https://orcid.org/0000-0001-8327-4936)

Christiane Ehrhart – Universität Hamburg, ZBH - Center for Bioinformatics, 20146 Hamburg, Germany; [orcid.org/0000-0003-1428-0042](https://orcid.org/0000-0003-1428-0042)

Complete contact information is available at: <https://pubs.acs.org/doi/10.1021/acs.jcim.3c00336>

### Author Contributions

<sup>†</sup>J.G. and C.E. contributed equally to this work. J.G. developed DoGSite3 and its new functionalities. C.E. performed the parameter optimization and prediction accuracy evaluations.

M.R. supervised the project. J.G., C.E., and M.R. wrote the manuscript.

### Notes

The authors declare the following competing financial interest(s): ProteinsPlus and the NAOMI ChemBio Suite use some methods that are jointly owned and/or licensed by/to BioSolveIT GmbH, Germany. M.R. is a shareholder of BioSolveIT GmbH.

## ■ ACKNOWLEDGMENTS

The authors thank the whole development team of the NAOMI library, forming the basis of this work, with special thanks to our colleague Konrad Diedrich for his help in creating the web interface. This work was supported by the German Federal Ministry of Education and Research as part of CompLS and de.NBI [031L0172, 031L0105]. C.E. is funded by Data Science in Hamburg - Helmholtz Graduate School for the Structure of Matter (Grant-ID: HIDSS-0002).

## ■ ABBREVIATIONS USED

PDB, Protein Data Bank; SVM, support vector machine; DoG, Difference-of-Gaussians; PSP, protein–solvent–protein; SAS, solvent-accessible surface area; RRO, relative residue overlap; CDR, cavity-defining residues; PR, predicted residues

## ■ REFERENCES

- (1) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.
- (2) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Židek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596*, 583–589.
- (3) EMBL-EBI AlphaFold Protein Structure Database. <https://alphafold.ebi.ac.uk/>, (accessed 23.02.2023).
- (4) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Adv. Drug Delivery Rev.* **1997**, *23*, 3–25.
- (5) Borrel, A.; Regad, L.; Xhaard, H.; Petitjean, M.; Camproux, A.-C. PockDrug: A Model for Predicting Pocket Druggability That Overcomes Pocket Estimation Uncertainties. *J. Chem. Inf. Model.* **2015**, *55*, 882–895.
- (6) Volkamer, A.; Kuhn, D.; Grombacher, T.; Rippmann, F.; Rarey, M. Combining Global and Local Measures for Structure-Based Druggability Predictions. *J. Chem. Inf. Model.* **2012**, *52*, 360–372.
- (7) Ehrhart, C.; Brinkjost, T.; Koch, O. Binding Site Characterization – Similarity, Promiscuity, and Druggability. *Med. Chem. Commun.* **2019**, *10*, 1145–1159.
- (8) Volkamer, A.; von Behren, M. M.; Bietz, S.; Rarey, M. In *Applied Chemoinformatics*; Engle, T., Gasteiger, J., Eds.; John Wiley & Sons, Ltd, 2018; Chapter 6.7, pp 283–311.
- (9) Hassan, N. M.; Alhossary, A. A.; Mu, Y.; Kwok, C.-K. Protein-Ligand Blind Docking Using QuickVina-W with Inter-Process Spatio-Temporal Integration. *Sci. Rep.* **2017**, *7*, 15451.
- (10) Kleywegt, G. J.; Jones, T. A. Detection, Delineation, Measurement and Display of Cavities in Macromolecular Structures. *Acta Crystallogr., Sect. D: Struct. Biol.* **1994**, *50*, 178–185.
- (11) Laskowski, R. A. SURFNET: A Program for Visualizing Molecular Surfaces, Cavities, and Intermolecular Interactions. *J. Mol. Graphics* **1995**, *13*, 323–330.

- (12) Weisel, M.; Proschak, E.; Schneider, G. PocketPicker: Analysis of Ligand Binding-Sites with Shape Descriptors. *Chem. Cent. J.* **2007**, *1*, 7.
- (13) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An Open Source Platform for Ligand Pocket Detection. *BMC Bioinf* **2009**, *10*, 168.
- (14) Halgren, T. New Method for Fast and Accurate Binding-Site Identification and Analysis. *Chem. Biol. Drug Des.* **2007**, *69*, 146–148.
- (15) Halgren, T. A. Identifying and Characterizing Binding Sites and Assessing Druggability. *J. Chem. Inf. Model.* **2009**, *49*, 377–389.
- (16) Ravindranath, P. A.; Sanner, M. F. AutoSite: An Automated Approach for Pseudo-Ligands Prediction—from Ligand-Binding Sites Identification to Predicting Key Ligand Atoms. *Bioinformatics* **2016**, *32*, 3142–3149.
- (17) Ghersi, D.; Sanchez, R. EasyMIFs and SiteHound: A Toolkit for the Identification of Ligand-Binding Sites in Protein Structures. *Bioinformatics* **2009**, *25*, 3185–3186.
- (18) Zhang, C.; Freddolino, P. L.; Zhang, Y. COFACTOR: Improved Protein Function Prediction by Combining Structure, Sequence and Protein–Protein Interaction Information. *Nucleic Acids Res.* **2017**, *45*, W291–W299.
- (19) Yang, J.; Roy, A.; Zhang, Y. BioLiP: A Semi-Manually Curated Database for Biologically Relevant Ligand–Protein Interactions. *Nucleic Acids Res.* **2012**, *41*, D1096–D1103.
- (20) Jendele, L.; Krivak, R.; Skoda, P.; Novotny, M.; Hoksza, D. PrankWeb: A Web Server for Ligand Binding Site Prediction and Visualization. *Nucleic Acids Res.* **2019**, *47*, W345–W349.
- (21) Pai, P. P.; Dattatreya, R. K.; Mondal, S. Ensemble Architecture for Prediction of Enzyme-Ligand Binding Residues Using Evolutionary Information. *Mol. Inf.* **2017**, *36*, 1700021.
- (22) Altschul, S. F.; Madden, T. L.; Schäffer, A. A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D. J. Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. *Nucleic Acids Res.* **1997**, *25*, 3389–3402.
- (23) Dessailly, B. H.; Lensink, M. F.; Orengo, C. A.; Wodak, S. J. LigASite—A Database of Biologically Relevant Binding Sites in Proteins with Known Apo-Structures. *Nucleic Acids Res.* **2007**, *36*, D667–D673.
- (24) Capra, J. A.; Laskowski, R. A.; Thornton, J. M.; Singh, M.; Funkhouser, T. A. Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure. *PLoS Comput. Biol.* **2009**, *5*, e1000585.
- (25) Yang, J.; Roy, A.; Zhang, Y. Protein–Ligand Binding Site Recognition Using Complementary Binding-Specific Substructure Comparison and Sequence Profile Alignment. *Bioinformatics* **2013**, *29*, 2588–2595.
- (26) Volkamer, A.; Griewel, A.; Grombacher, T.; Rarey, M. Analyzing the Topology of Active Sites: On the Prediction of Pockets and Subpockets. *J. Chem. Inf. Model.* **2010**, *50*, 2041–2052.
- (27) Fährrolfes, R.; Bietz, S.; Flachsenberg, F.; Meyder, A.; Nittinger, E.; Otto, T.; Volkamer, A.; Rarey, M. ProteinsPlus: A Web Portal for Structure Analysis of Macromolecules. *Nucleic Acids Res.* **2017**, *45*, W337–W343.
- (28) Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind Database: Methodologies and Updates. *J. Med. Chem.* **2005**, *48*, 4111–4119.
- (29) Paul, N.; Kellenberger, E.; Bret, G.; Müller, P.; Rognan, D. Recovering the True Targets of Specific Ligands by Virtual Screening of the Protein Data Bank. *Proteins: Struct., Funct., Bioinf.* **2004**, *54*, 671–680.
- (30) Kellenberger, E.; Müller, P.; Schalon, C.; Bret, G.; Foata, N.; Rognan, D. sc-PDB: An Annotated Database of Druggable Binding Sites from the Protein Data Bank. *J. Chem. Inf. Model.* **2006**, *46*, 717–727.
- (31) Desaphy, J.; Bret, G.; Rognan, D.; Kellenberger, E. sc-PDB: A 3D-Database of Ligandable Binding Sites—10 Years On. *Nucleic Acids Res.* **2015**, *43*, D399–D404.
- (32) Rognan, D. sc-PDB. <http://bioinfo-pharma.u-strasbg.fr/scPDB/>, (accessed 23.02.2023).
- (33) Krivák, R.; Hoksza, D. Improving Protein-Ligand Binding Site Prediction Accuracy by Classification of Inner Pocket Points Using Local Features. *J. Cheminf.* **2015**, *7*, 12.
- (34) Ehrh, C. Protein Binding Site Comparison. Ph.D. thesis, Technische Universität Dortmund, 2019.
- (35) Clark, J. J.; Orban, Z. J.; Carlson, H. A. Predicting Binding Sites from Unbound Versus Bound Protein Structures. *Sci. Rep.* **2020**, *10*, 15856.
- (36) Levitt, D. G.; Banaszak, L. J. POCKET: A Computer Graphics Method for Identifying and Displaying Protein Cavities and Their Surrounding Amino Acids. *J. Mol. Graphics* **1992**, *10*, 229–234.
- (37) Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: Automatic and Efficient Detection of Potential Small Molecule-Binding Sites in Proteins. *J. Mol. Graphics Modell.* **1997**, *15*, 359–363.
- (38) Urbaczek, S.; Kolodzik, A.; Fischer, J. R.; Lippert, T.; Heuser, S.; Groth, I.; Schulz-Gasch, T.; Rarey, M. NAOMI: On the Almost Trivial Task of Reading Molecules from Different File Formats. *J. Chem. Inf. Model.* **2011**, *51*, 3199–3207.
- (39) Urbaczek, S.; Kolodzik, A.; Groth, I.; Heuser, S.; Rarey, M. Reading PDB: Perception of Molecules from 3D Atomic Coordinates. *J. Chem. Inf. Model.* **2013**, *53*, 76–87.
- (40) Mahmoud, A. H.; Masters, M. R.; Yang, Y.; Lill, M. A. Elucidating the Multiple Roles of Hydration for Accurate Protein-Ligand Binding Prediction via Deep Learning. *Commun. Chem.* **2020**, *3*, 19.
- (41) Reulecke, I.; Lange, G.; Albrecht, J.; Klein, R.; Rarey, M. Towards an Integrated Description of Hydrogen Bonding and Dehydration: Decreasing False Positives in Virtual Screening with the HYDE Scoring Function. *ChemMedChem.* **2008**, *3*, 885–897.
- (42) Ehrh, C.; Brinkjost, T.; Koch, O. A Benchmark Driven Guide to Binding Site Comparison: An Exhaustive Evaluation Using Tailor-Made Data Sets (ProSPECCTs). *PLoS Comput. Biol.* **2018**, *14*, e1006483.
- (43) Li, Y.; Liu, Z.; Li, J.; Han, L.; Liu, J.; Zhao, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. *J. Chem. Inf. Model.* **2014**, *54*, 1700–1716.
- (44) Cimermanic, P.; Weinkam, P.; Rettenmaier, T. J.; Bichmann, L.; Keedy, D. A.; Woldeyes, R. A.; Schneidman-Duhovny, D.; Demerdash, O. N.; Mitchell, J. C.; Wells, J. A.; Fraser, J. S.; Sali, A. CryptoSite: Expanding the Druggable Proteome by Characterization and Prediction of Cryptic Binding Sites. *J. Mol. Biol.* **2016**, *428*, 709–719.
- (45) Huang, W.; Wang, G.; Shen, Q.; Liu, X.; Lu, S.; Geng, L.; Huang, Z.; Zhang, J. ASBench: Benchmarking Sets for Allosteric Discovery. *Bioinformatics* **2015**, *31*, 2598–2600.
- (46) Graef, J.; Ehrh, C.; Diedrich, K.; Poppinga, M.; Ritter, N.; Rarey, M. Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures. *J. Med. Chem.* **2022**, *65*, 1384–1395.
- (47) Diedrich, K.; Graef, J.; Schöning-Stierand, K.; Rarey, M. GeoMine: Interactive Pattern Mining of Protein–Ligand Interfaces in the Protein Data Bank. *Bioinformatics* **2021**, *37*, 424–425.
- (48) Bietz, S.; Rarey, M. SIENA: Efficient Compilation of Selective Protein Binding Site Ensembles. *J. Chem. Inf. Model.* **2016**, *56*, 248–259.
- (49) Zhao, Y. H.; Abraham, M. H.; Zissimos, A. M. Fast Calculation of van der Waals Volume as a Sum of Atomic and Bond Contributions and Its Application to Drug Compounds. *J. Org. Chem.* **2003**, *68*, 7368–7373.
- (50) Tubiana, J.; Schneidman-Duhovny, D.; Wolfson, H. J. ScanNet: An Interpretable Geometric Deep Learning Model for Structure-Based Protein Binding Site Prediction. *Nat. Methods* **2022**, *19*, 730–739.
- (51) Schrödinger, LLC. *The PyMOL Molecular Graphics System*, Version 2.3, 2015.

# Supporting Information

## Binding Site Detection Remastered - Enabling Fast, Robust, and Reliable Binding Site Detection and Descriptor Calculation with DoGSite3

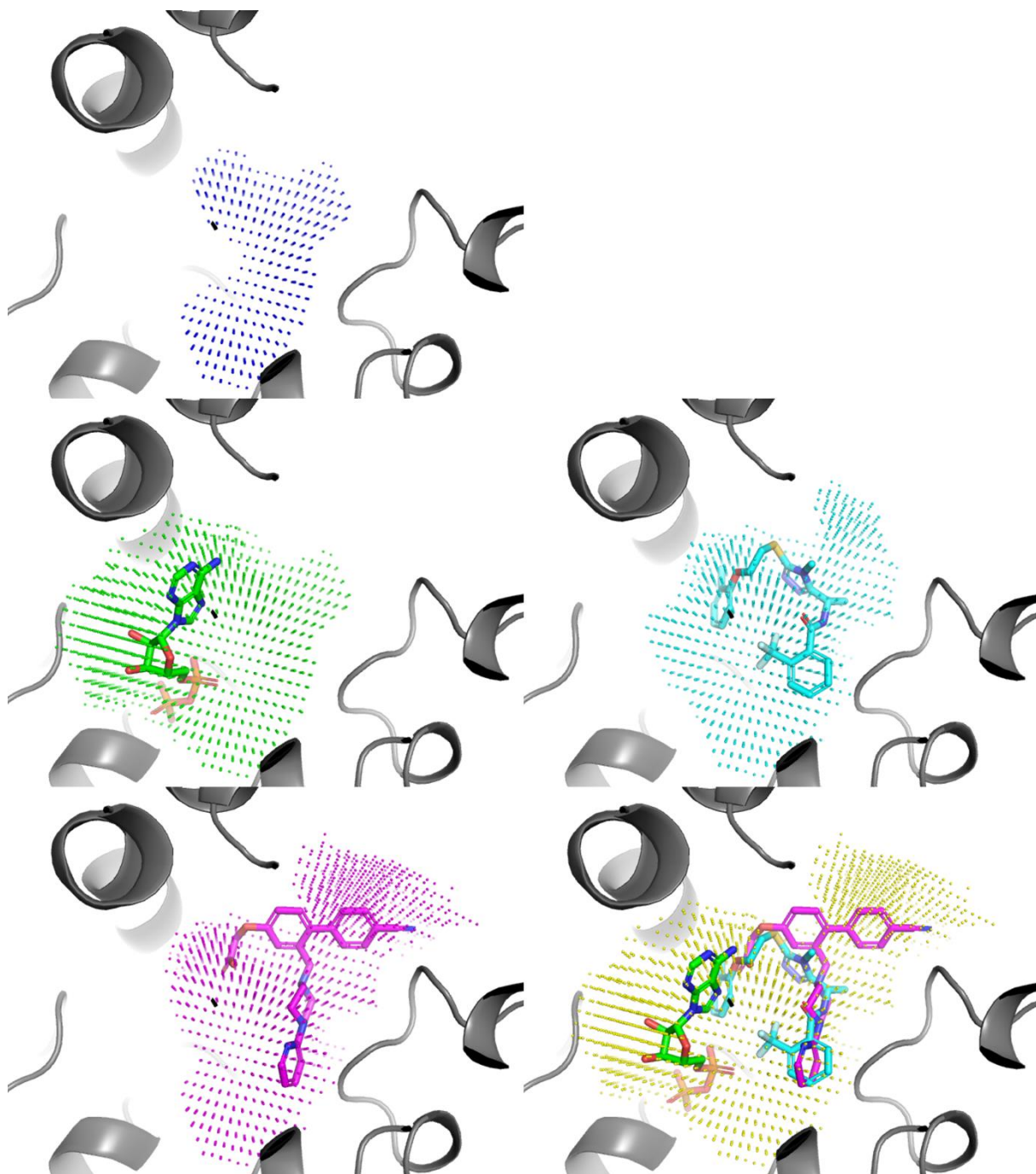
Author Names: Joel Graef, Christiane Ehrt, Matthias Rarey\*

Author Address: Universität Hamburg, Center for Bioinformatics, Bundesstrasse 43, 20146 Hamburg, Germany

E-mail: [matthias.rarey@uni-hamburg.de](mailto:matthias.rarey@uni-hamburg.de)

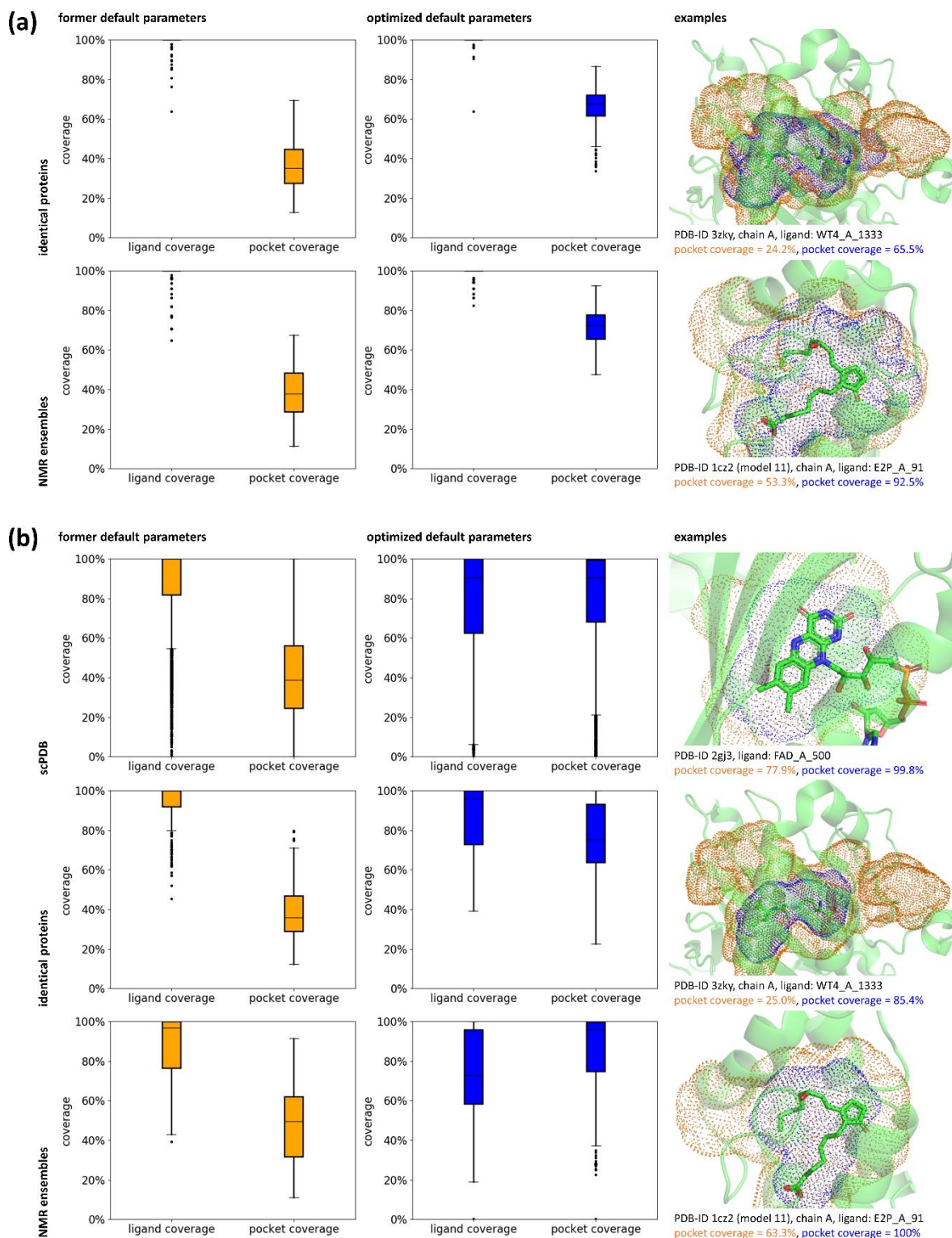
### Table of Contents

Figure S1.....	2
Figure S2.....	3
Figure S3.....	4
Figure S4.....	5
Figure S5.....	6
Figure S6.....	7
Figure S7.....	8
Figure S8.....	9
Figure S9.....	10
Figure S10.....	11
Figure S11.....	12
Figure S12.....	13
Figure S13.....	14
Figure S14.....	15
Figure S15.....	16
Figure S16.....	17
Table S1.....	18
Table S2.....	19
Table S3.....	21
References .....	23
	S1



**Figure S1.** Binding site definitions for pantothenate kinase (PanK) from *Mycobacterium tuberculosis* (PDB ID 2zsa, chain A, grey). The upper figure shows the pocket in the proximity of the ligand ADP (green stick representation) that DoGSite3 predicted without applying a bias by the ligand (pocket 1: blue). The middle left figure shows the predicted binding site if ADP is used to bias the pocket grid (green). The middle right figure shows the predicted binding site if the ligand ZVV of the aligned chain B of the structure of PanK with the PDB ID 4bfv is used for grid biasing (cyan). Correspondingly, the lower left figure shows the predicted site if the ligand ZVY of the aligned chain B of the structure of PanK with the PDB ID 4bfy is used for grid biasing (pink). Finally, the lower right figure shows the binding site definition if the ensemble of all three ligands for the three aligned PanK structures is used for grid biasing (yellow).

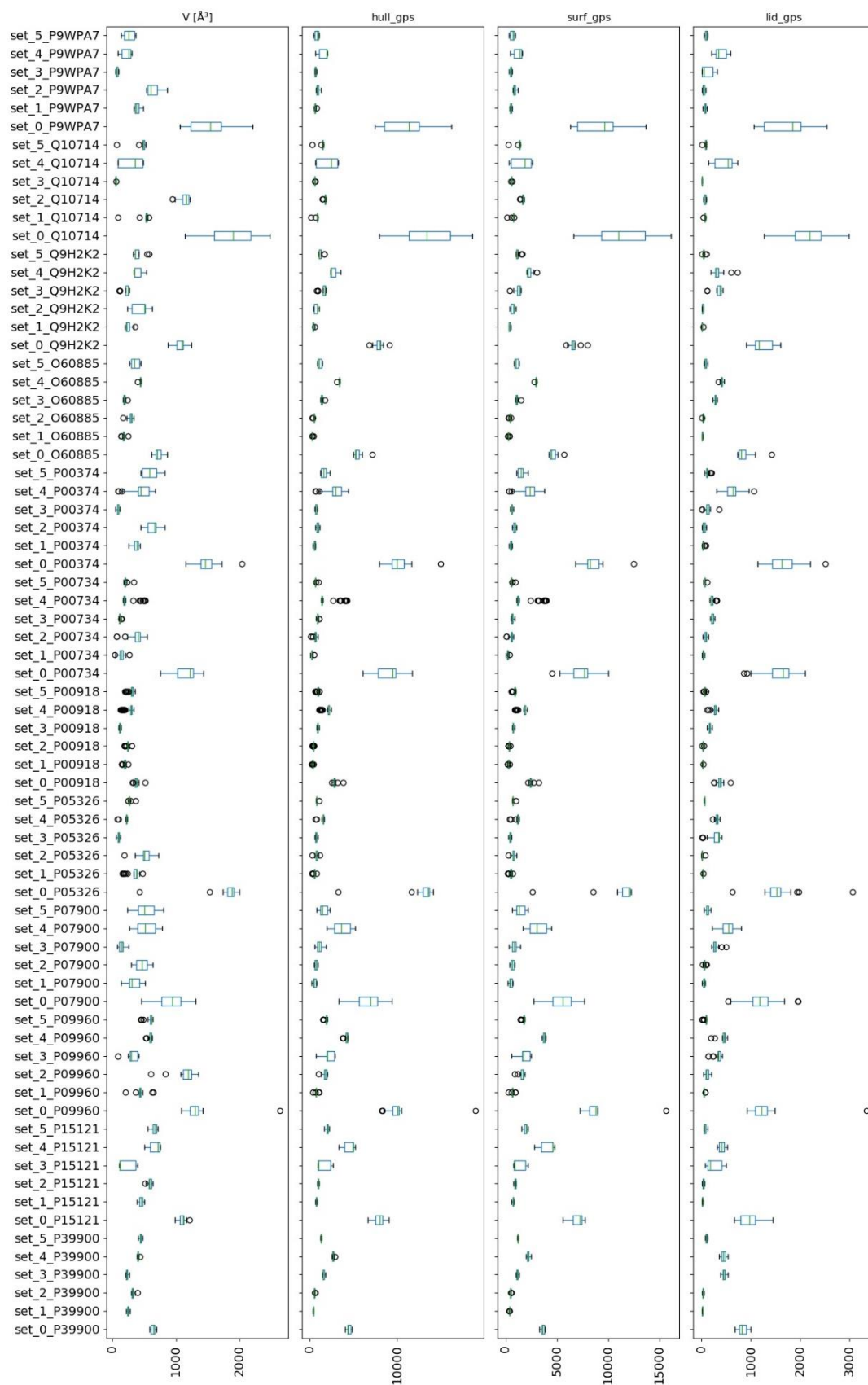
S2



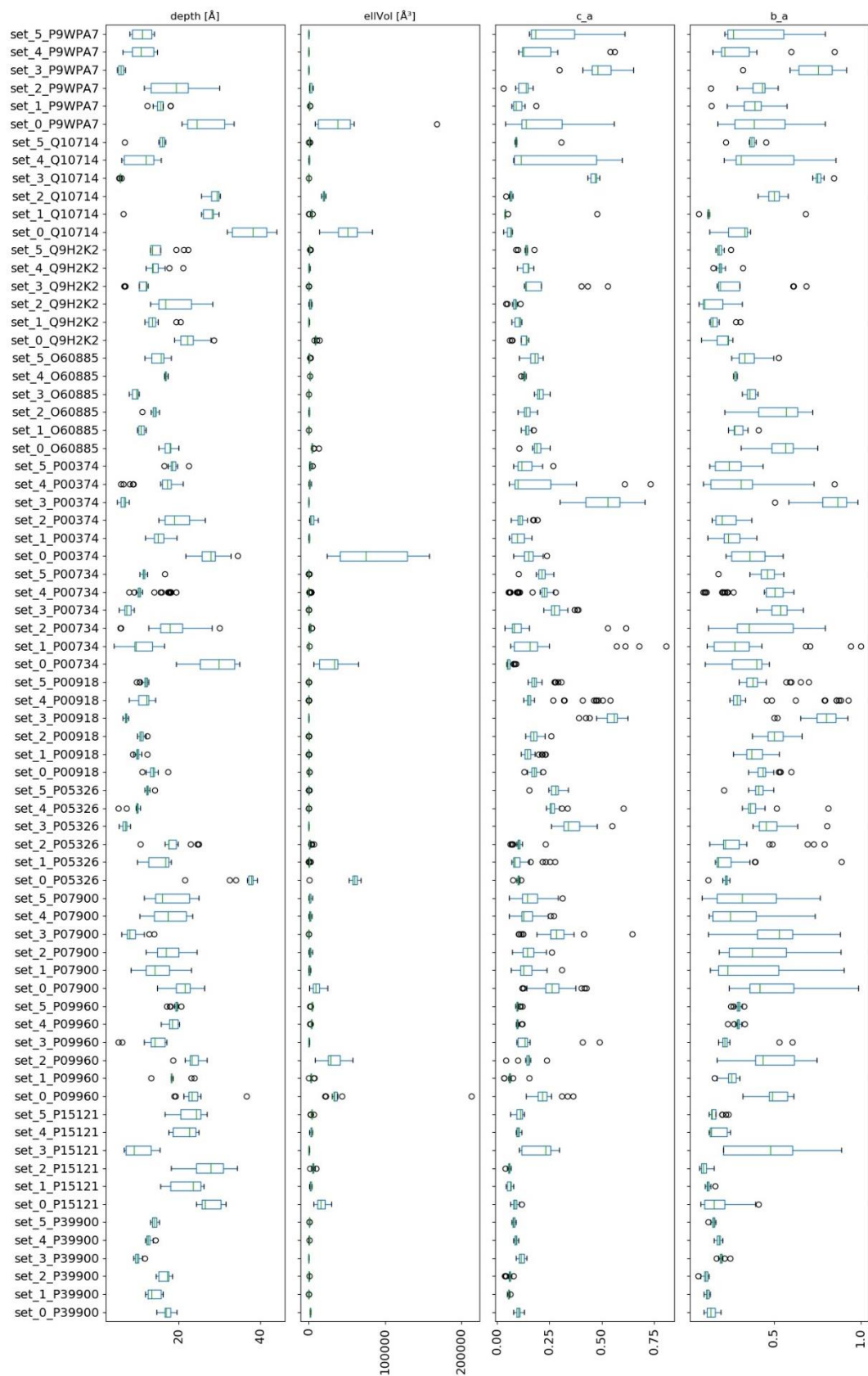
**Figure S2.** Analysis of the ligand and pocket coverage for ligand-defined binding sites in the datasets of identical proteins and NMR ensembles and the scPDB with the original default parameters of DoGSite (orange) and the more robust and reliable default parameters of DoGSite3 (blue). For the analysis presented in (a), the new feature of DoGSite3 for biasing the grid annotation by reference ligands was applied. The outcomes when analyzing unbiased results are depicted in (b). The structures depicted on the right are examples of pockets with very low pocket coverage with the original DoGSite parameters and a much higher one with the optimized DoGSite3 default parameters. The corresponding figures were generated with PyMOL(TM) Molecular Graphics System, version 2.3.0.

S3

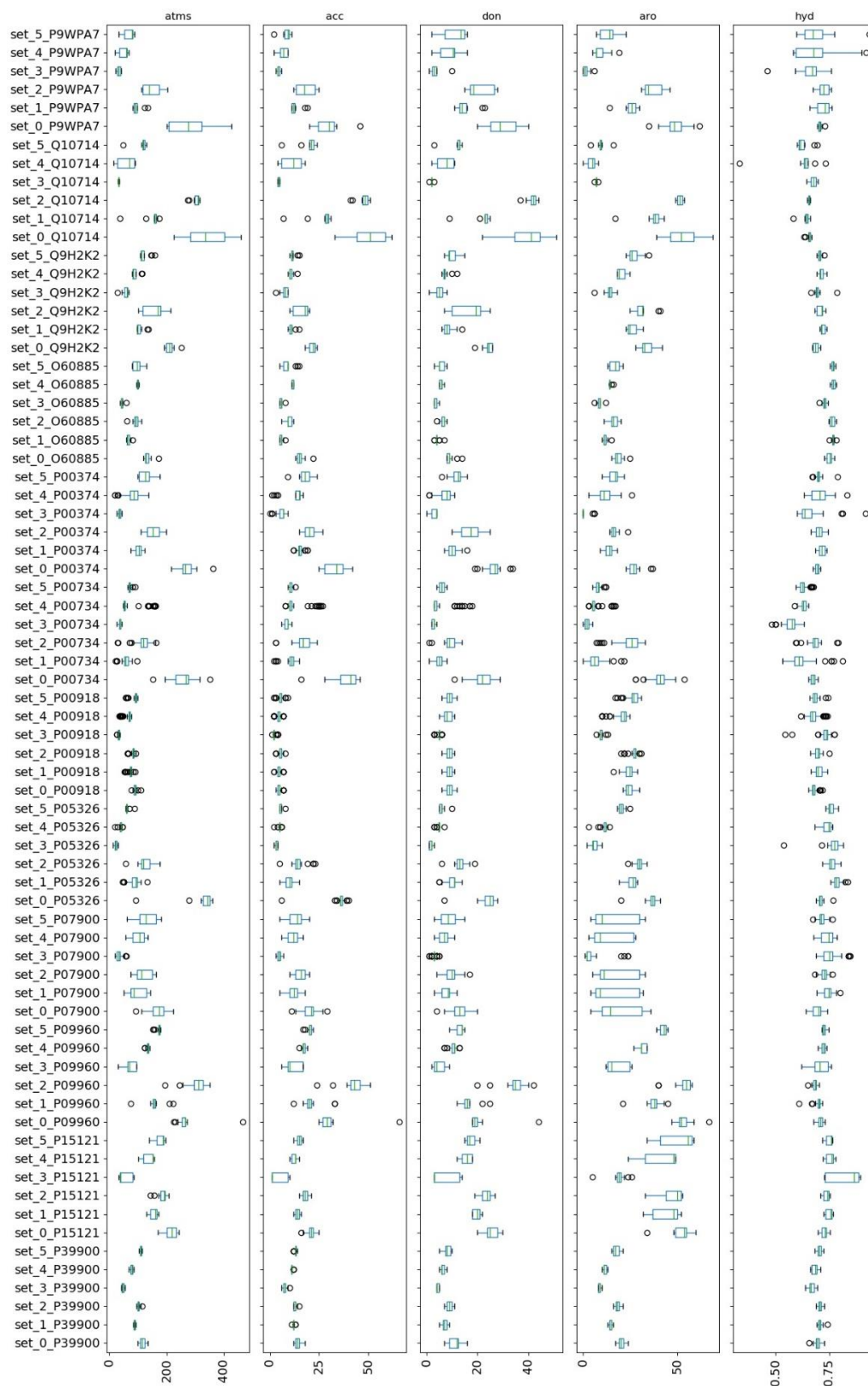




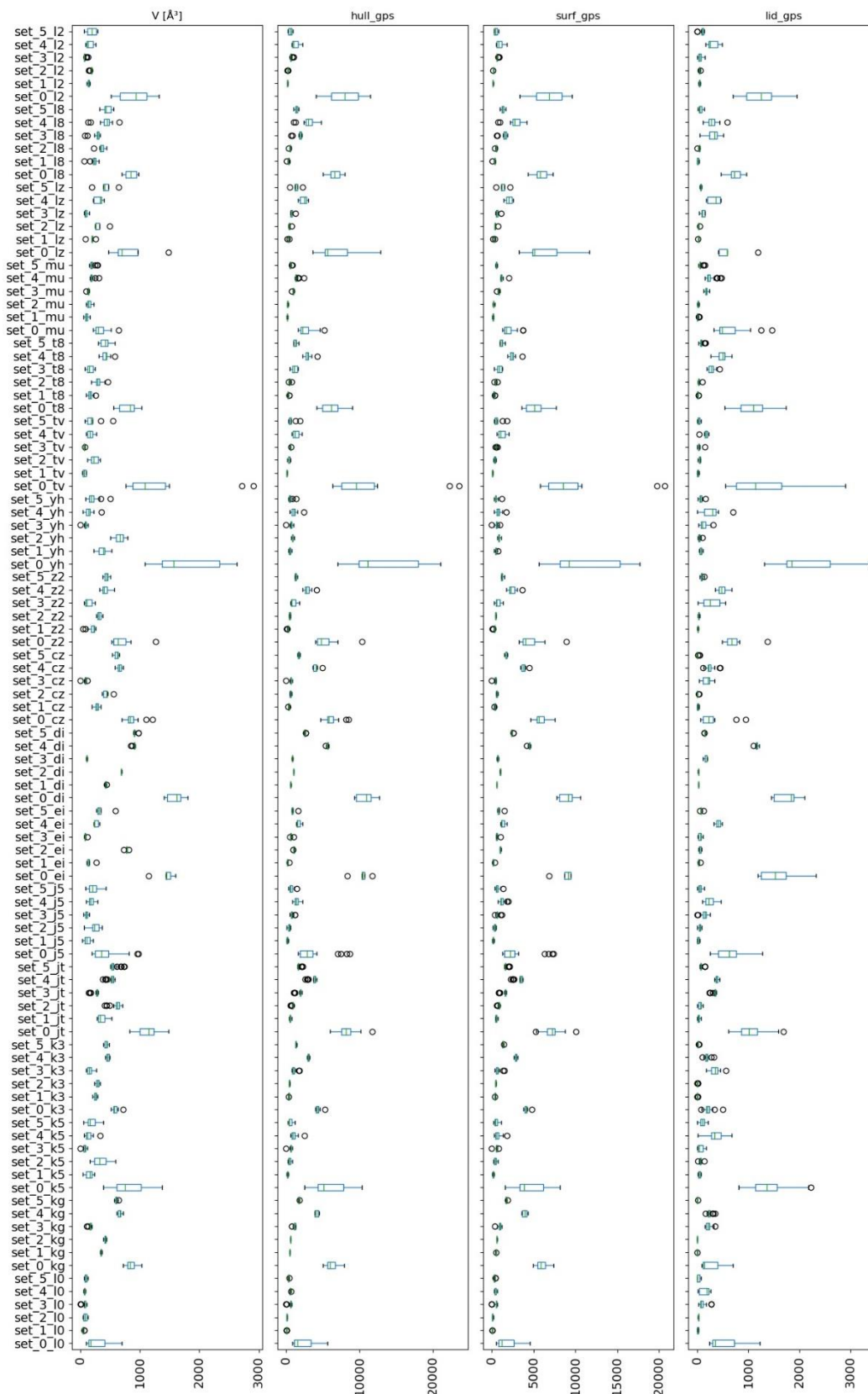
**Figure S3.** Robustness of the geometric pocket descriptors volume (V), hull grid points (hull\_gps), surface grid points (surf\_gps), and lid surface grid points (lid\_gps) within the groups of identical proteins for the former default parameters (set 0) and the newly selected parameter combinations (sets 1 to 5). The proteins' UniProt Accession Numbers are provided.



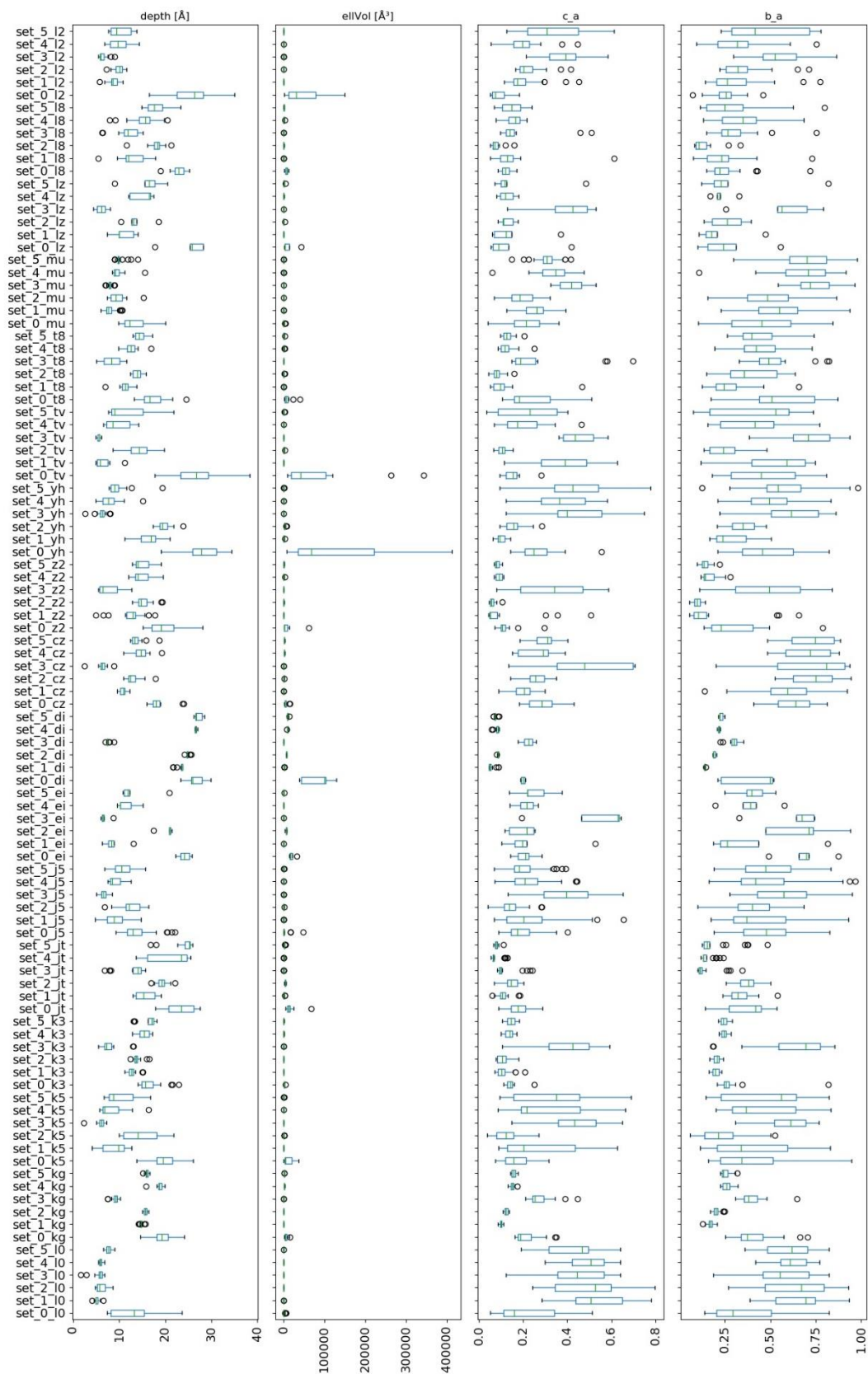
**Figure S4.** Robustness of the geometric descriptors pocket depth (depth), ellipsoid volume (ellVol), the relationship of the ellipsoid axes c and b (c\_b), and b and a (b\_a) within the groups of identical proteins for the former default parameters (set 0) and the newly selected parameter combinations (sets 1 to 5). The proteins' UniProt Accession Numbers are provided.



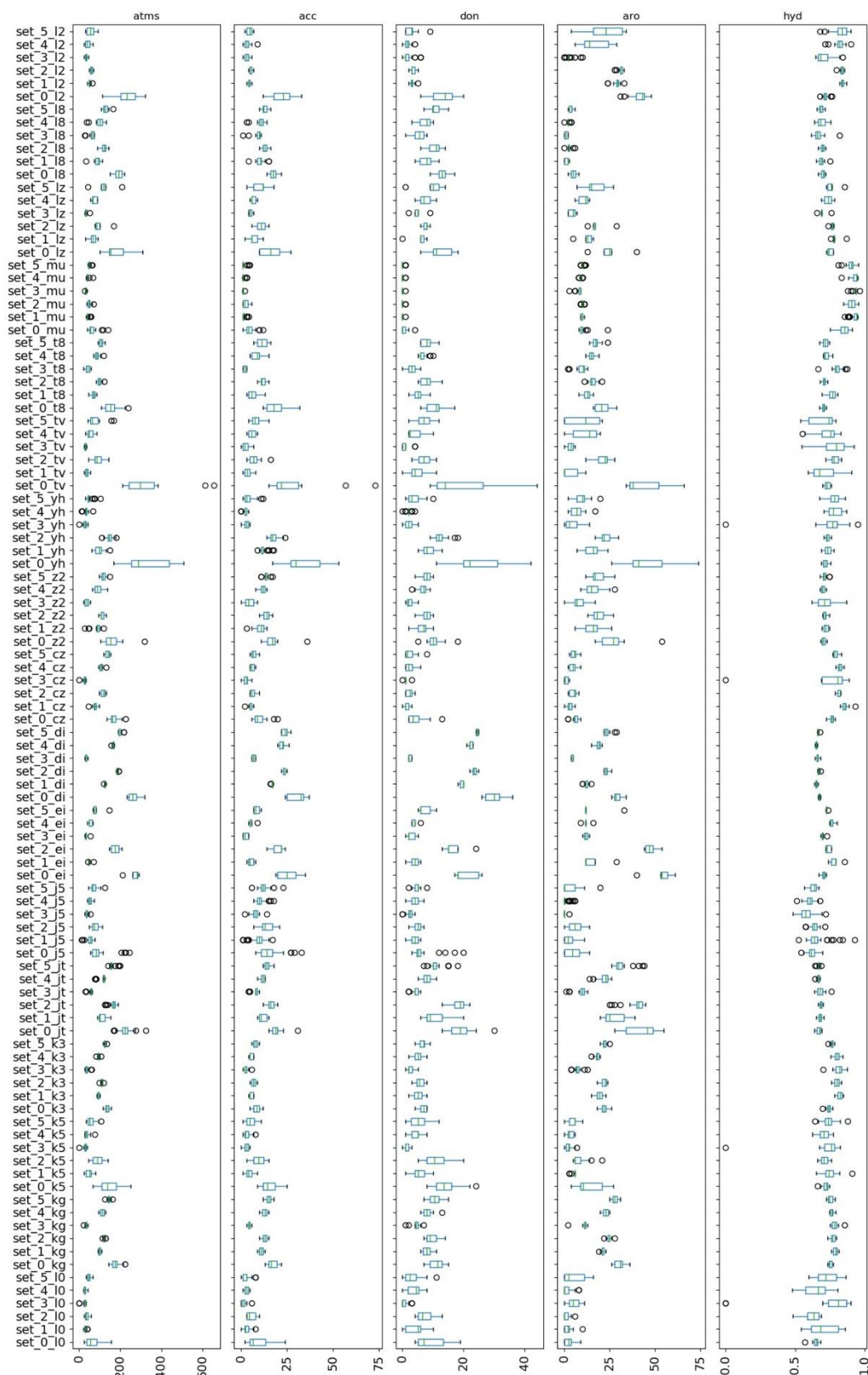
**Figure S5.** Robustness of the chemical descriptors number of atoms (atms), number of hydrogen bond acceptor atoms (acc), number of hydrogen bond donor atoms (don), number of aromatic atoms (aro), and hydrophobicity (hyd) within the groups of identical proteins for the former default parameters (set 0) and the newly selected parameter combinations (sets 1 to 5). The proteins' UniProt Accession Numbers are provided.



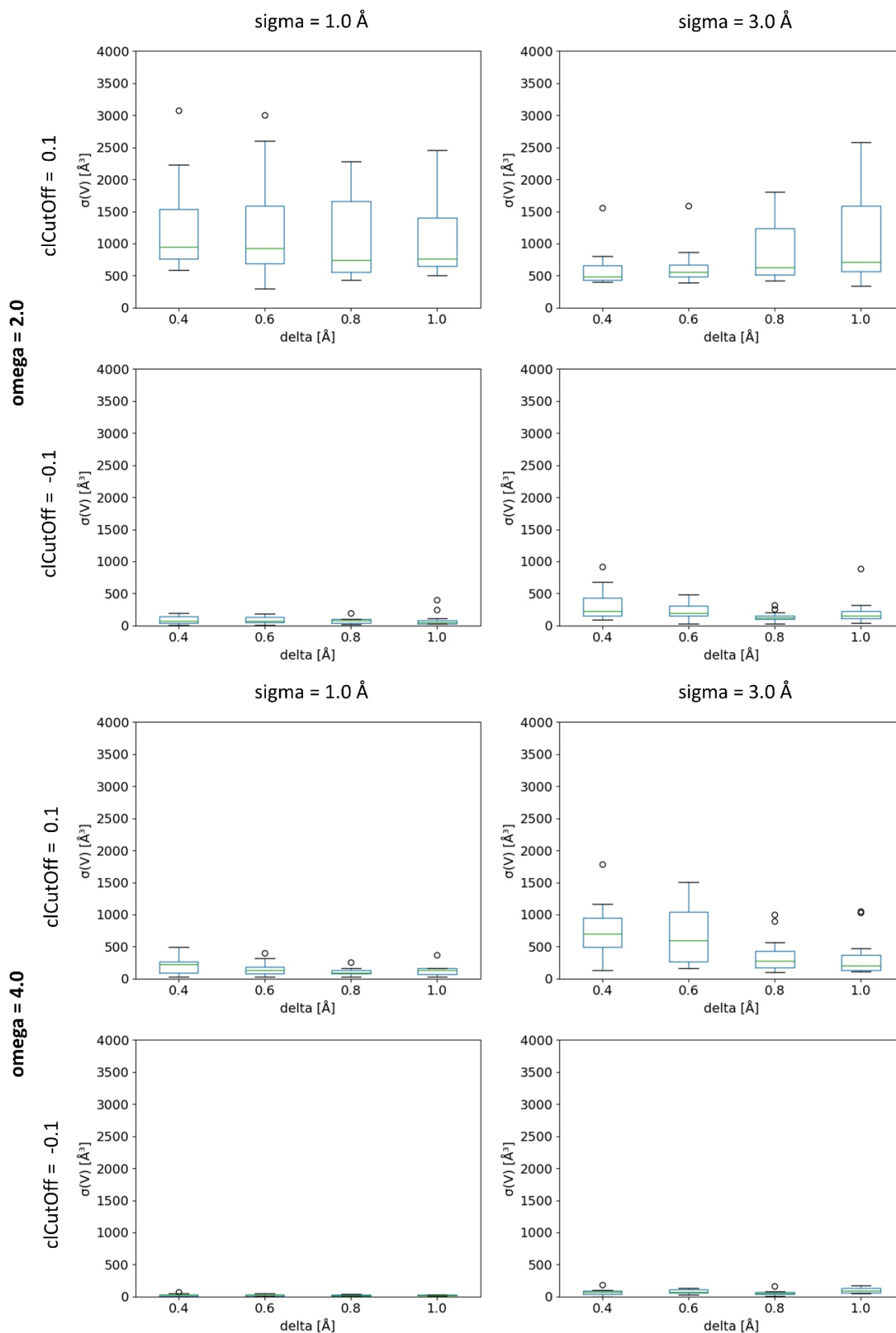
**Figure S6.** Robustness of the geometric pocket descriptors volume (V), hull grid points (hull\_gps), surface grid points (surf\_gps), and lid surface grid points (lid\_gps) within the groups of NMR ensembles for the former default parameters (set 0) and the newly selected parameter combinations (sets 1 to 5). The proteins' two central characters of the PDB code are given.



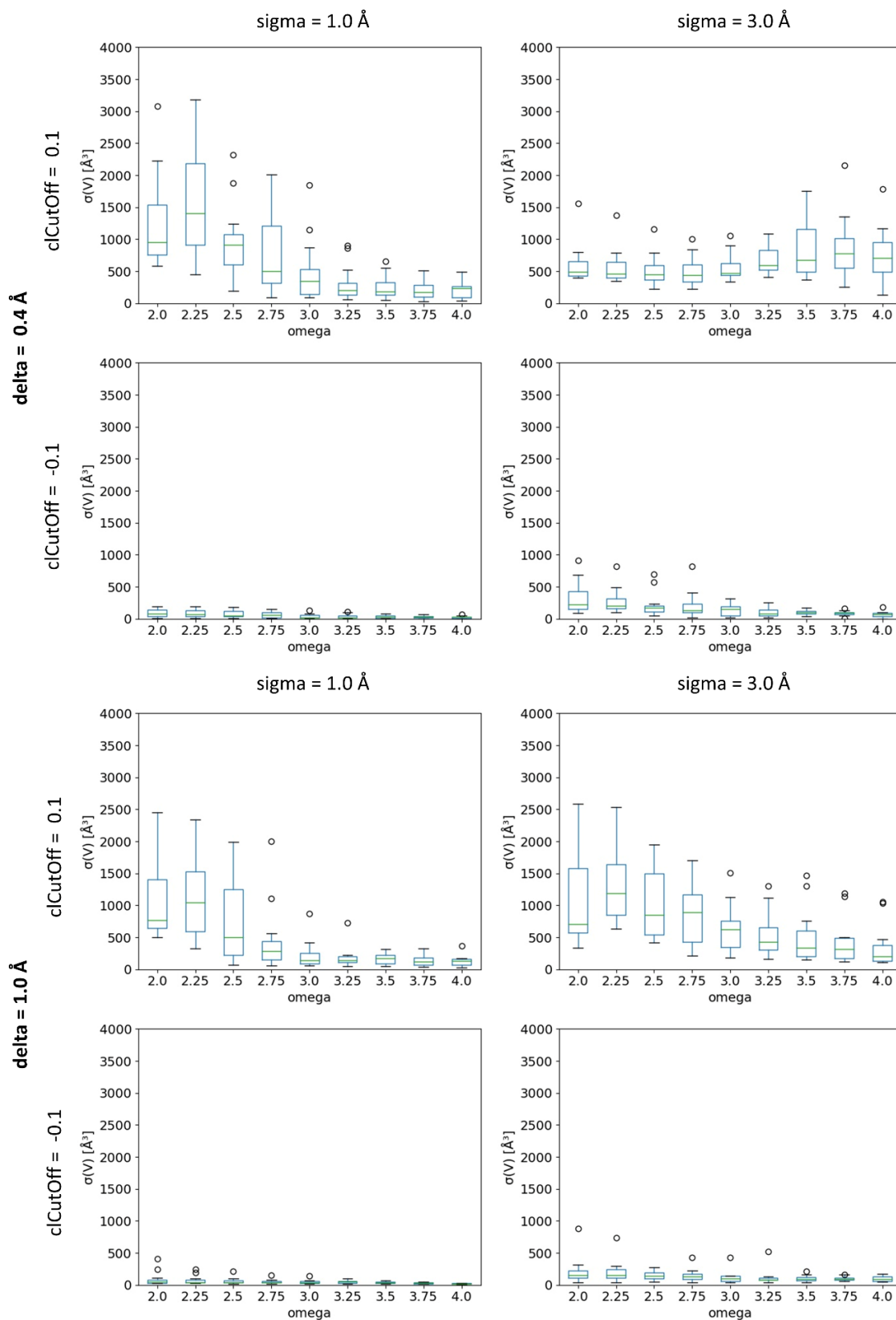
**Figure S7.** Robustness of the geometric descriptors pocket depth (depth), ellipsoid volume (ellVol), the relationship of the ellipsoid axes c and b (c\_b), and b and a (b\_a) within the groups of NMR ensembles for the former default parameters (set 0) and the newly selected parameter combinations (sets 1 to 5). The proteins' two central characters of the PDB code are given.



**Figure S8.** Robustness of the chemical descriptors number of atoms (atms), number of hydrogen bond acceptor atoms (acc), number of hydrogen bond donor atoms (don), number of aromatic atoms (aro) and hydrophobicity (hyd) within the groups of NMR ensembles for the former default parameters (set 0) and the newly selected parameter combinations (sets 1 to 5). The proteins' two central characters of the PDB code are given.

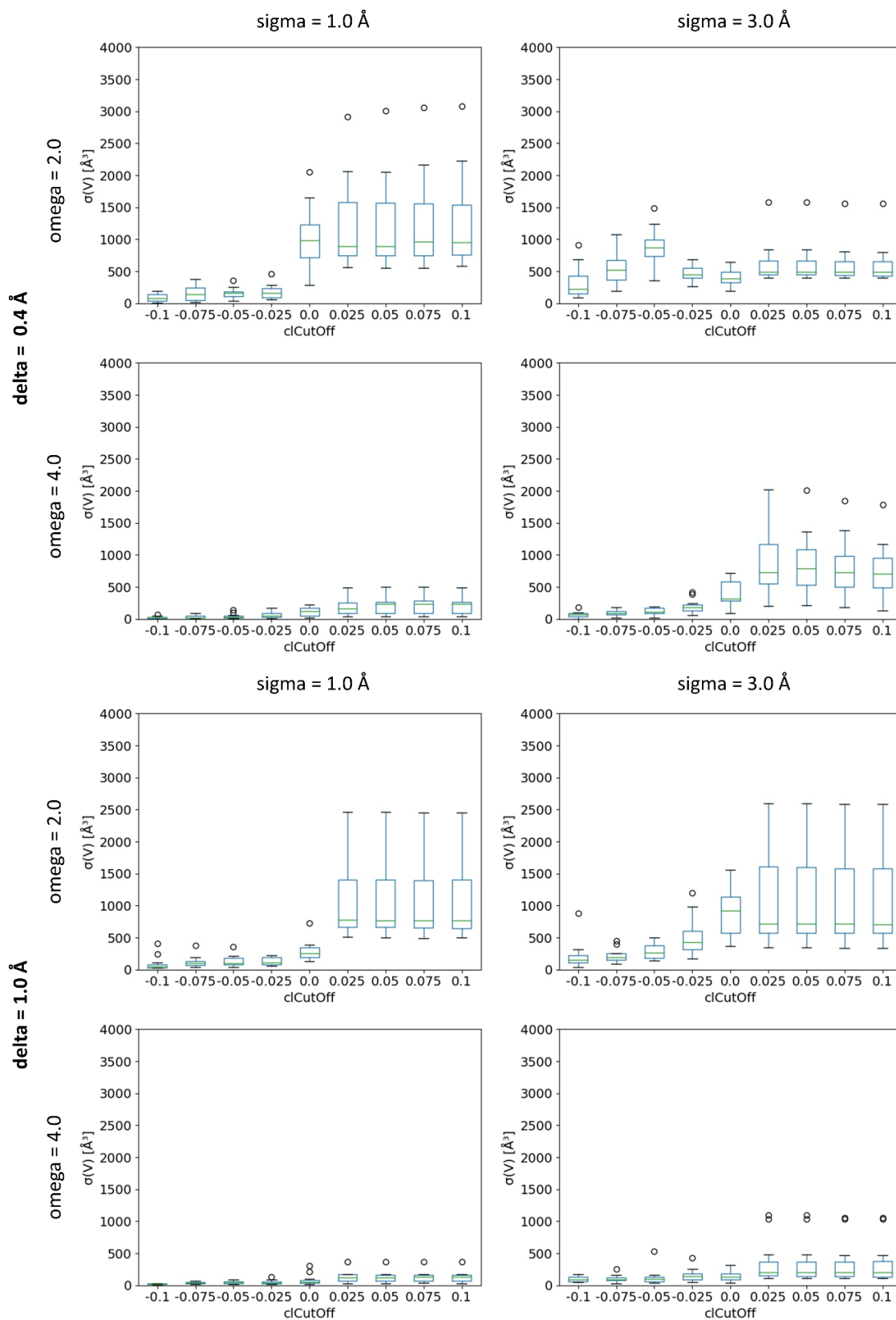


**Figure S9.** Impact of the grid spacing (delta) on the volume standard deviations in the groups of the identical proteins dataset. The parameters density variance scaling (omega), DoG density cutoff (cICutOff), and dilation radius (sigma) were kept constant.

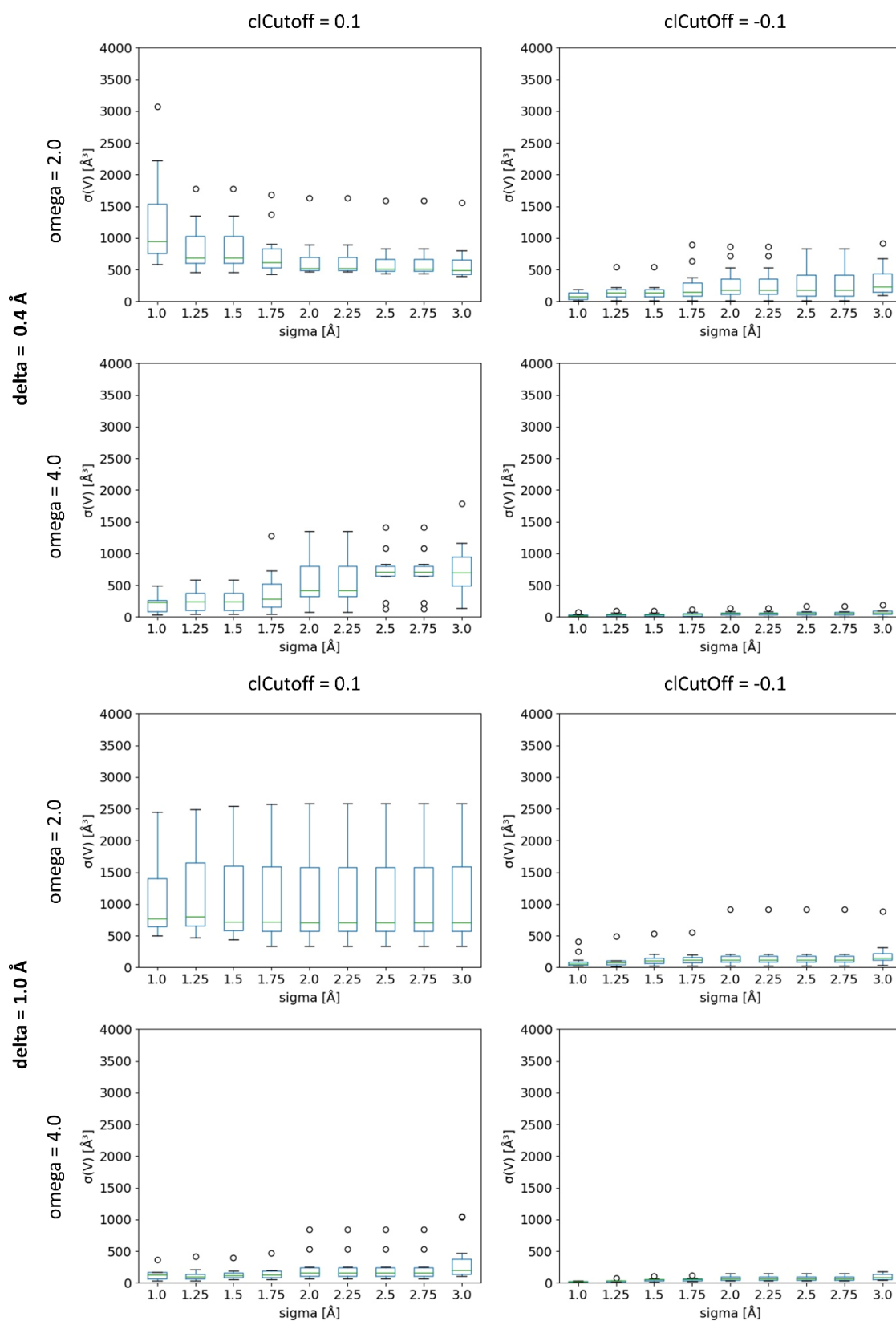


**Figure S10.** Impact of the density variance scaling ( $\omega$ ) on the volume standard deviations in the groups of the identical proteins dataset. The parameters grid spacing ( $\delta$ ), DoG density cutoff ( $cCutOff$ ), and dilation radius ( $\sigma$ ) were kept constant.

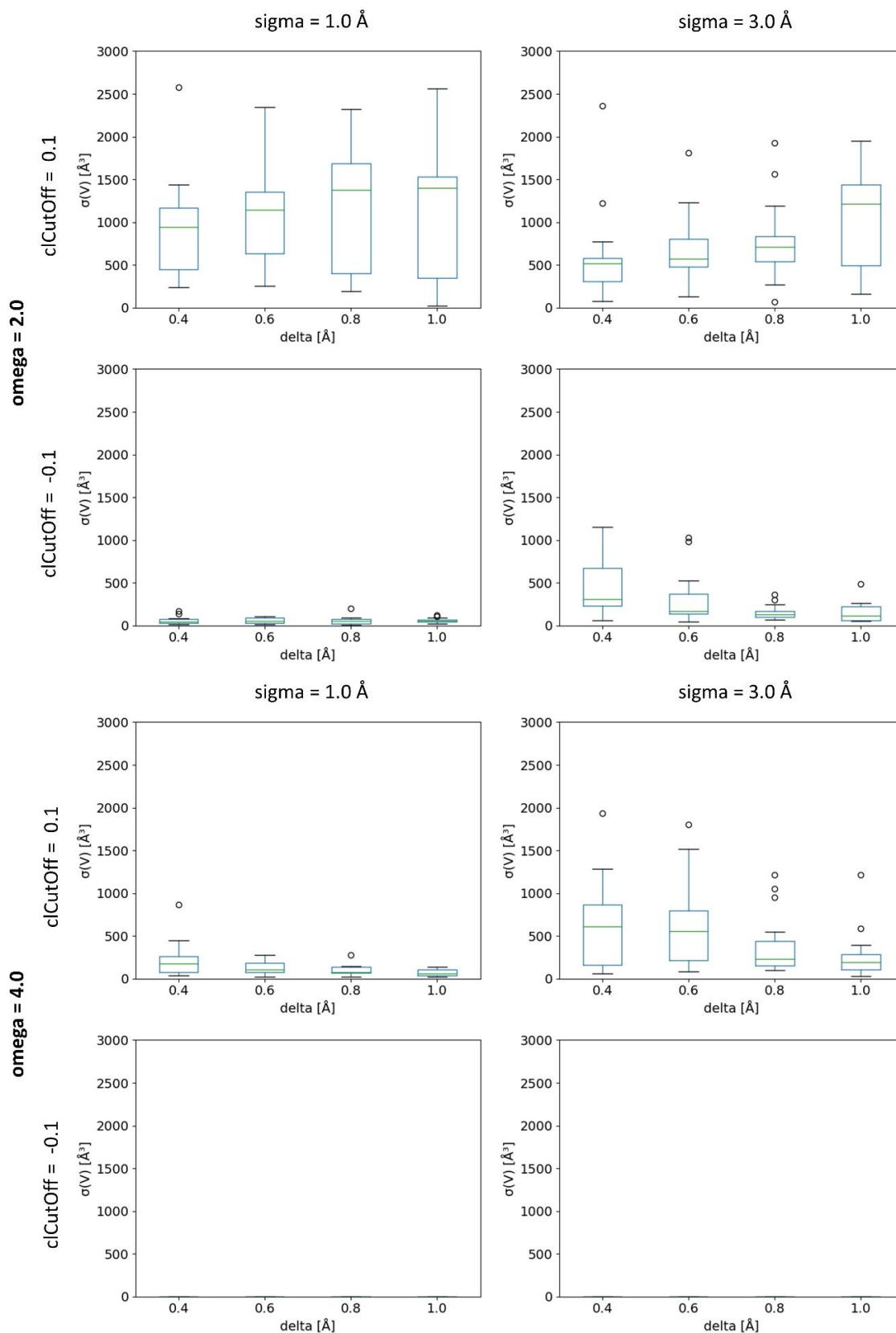




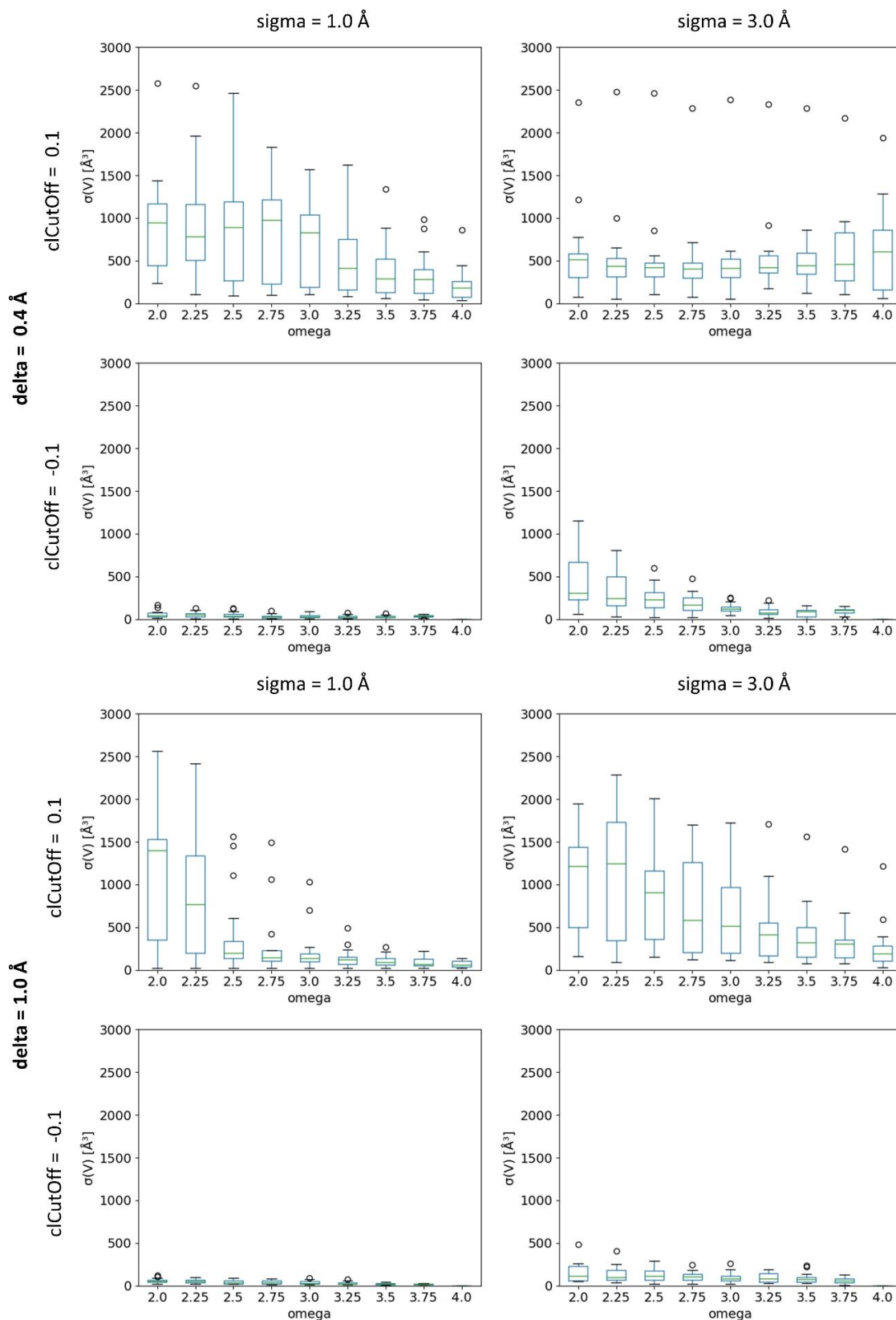
**Figure S11.** Impact of the DoG density cutoff ( $\text{clCutOff}$ ) on the volume standard deviations in the groups of the identical proteins dataset. The parameters grid spacing ( $\delta$ ), density variance scaling ( $\omega$ ), and dilation radius ( $\sigma$ ) were kept constant.



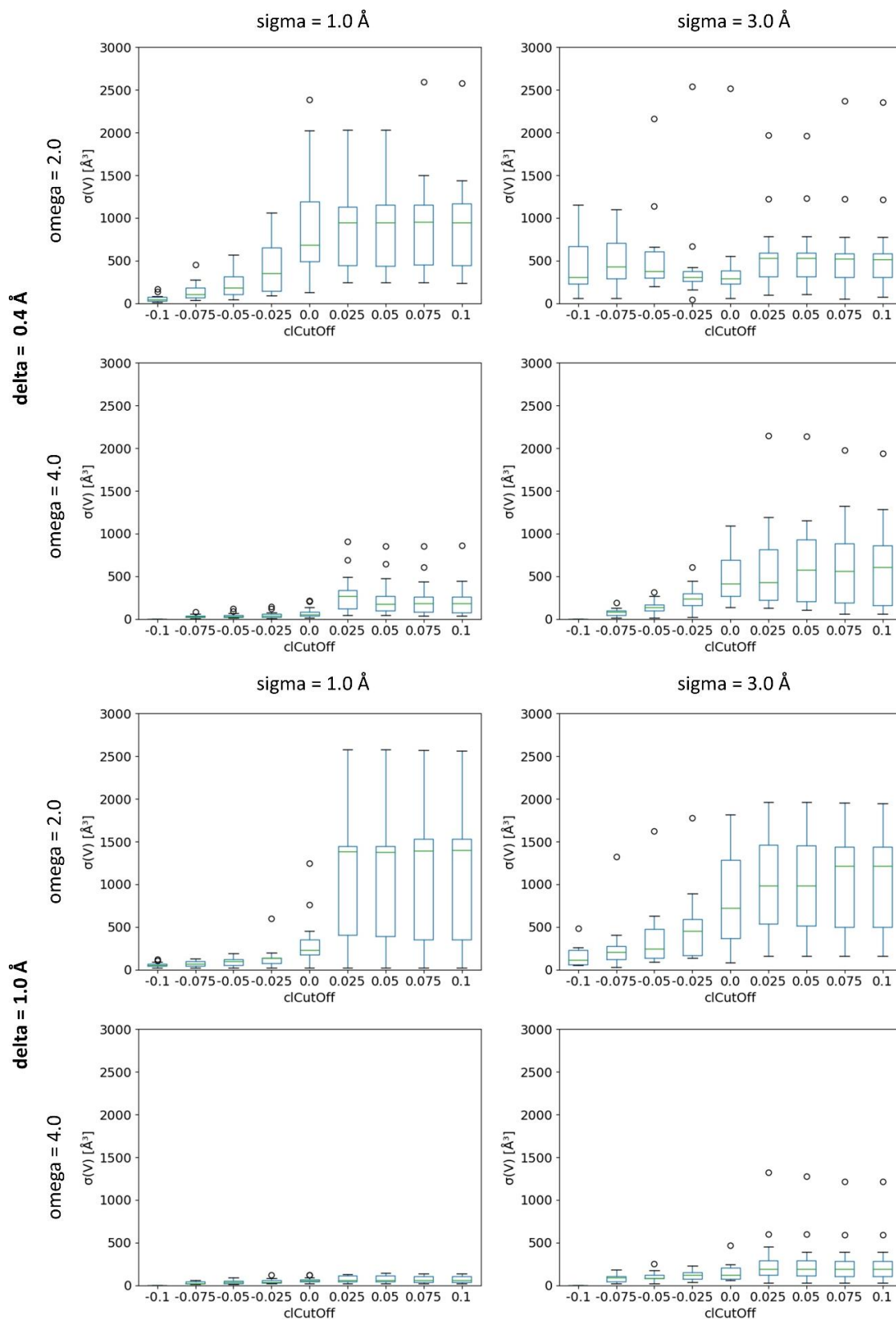
**Figure S12.** Impact of the dilation radius ( $\sigma$ ) on the volume standard deviations in the groups of the identical proteins dataset. The parameters grid spacing ( $\delta$ ), density variance scaling ( $\omega$ ), and DoG density cutoff ( $cICutoff$ ) were kept constant.



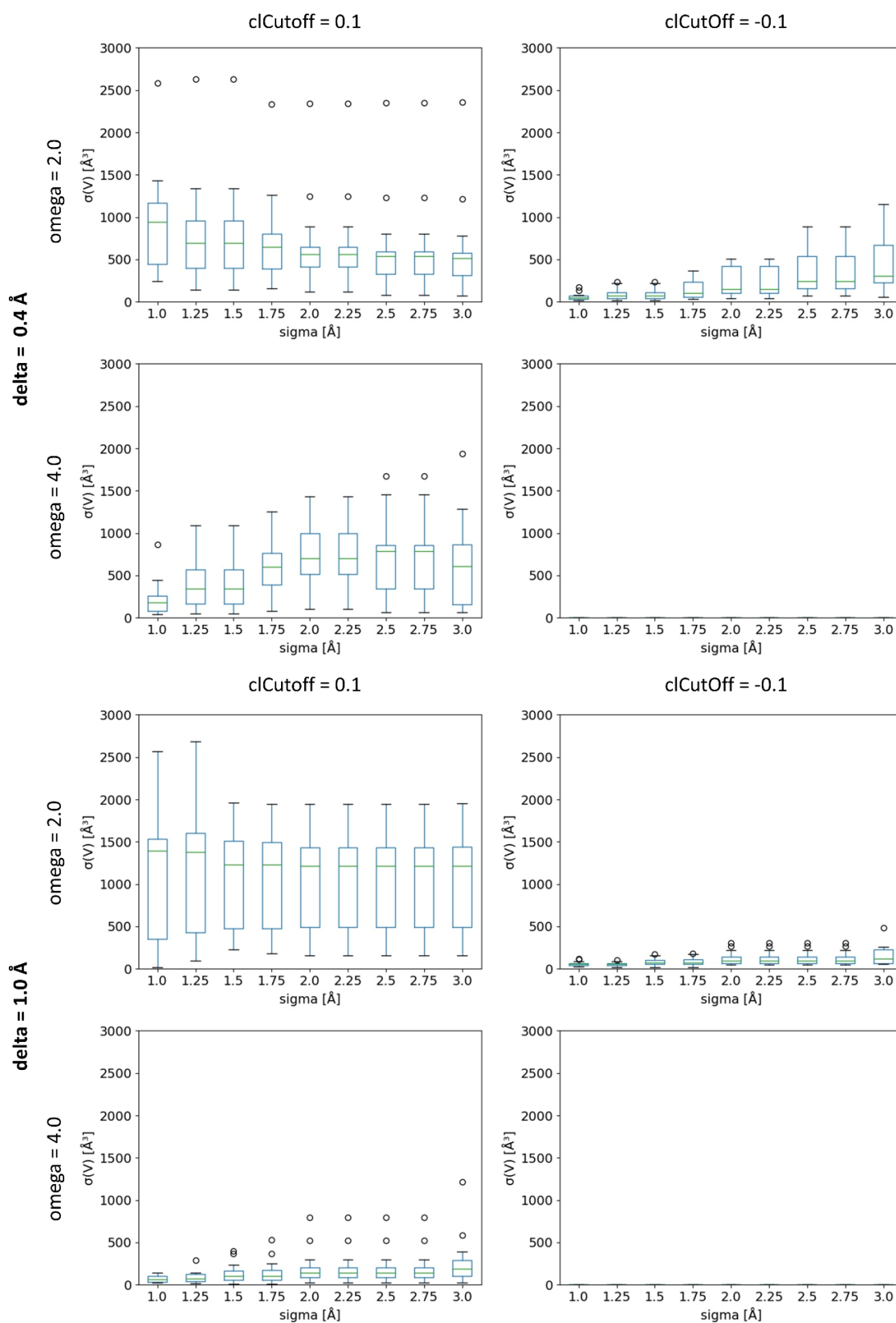
**Figure S13.** Impact of the grid spacing ( $\delta$ ) on the volume standard deviations in the groups of the NMR ensembles dataset. The parameters density variance scaling ( $\omega$ ), DoG density cutoff ( $cICutOff$ ), and dilation radius ( $\sigma$ ) were kept constant. The parameter combinations in the bottom two figures (with a DoG density cutoff of  $-0.1$  and a density variance scaling of  $4.0$ ) were not evaluated, as no pockets were detected for all structures in at least one ensemble.



**Figure S14.** Impact of the density variance scaling ( $\omega$ ) on the volume standard deviations in the groups of the NMR ensembles dataset. The parameters grid spacing ( $\delta$ ), DoG density cutoff ( $cCutOff$ ), and dilation radius ( $\sigma$ ) were kept constant. The parameter combinations with a DoG density cutoff of -0.1 and a density variance scaling of 4.0 were not evaluated, as no pockets were detected for all structures in at least one ensemble.



**Figure S15.** Impact of the DoG density cutoff (clCutOff) on the volume standard deviations in the groups of the NMR ensembles dataset. The parameters grid spacing (delta), density variance scaling (omega), and dilation radius (sigma) were kept constant. The parameter combinations with a DoG density cutoff of -0.1 and a density variance scaling of 4.0 were not evaluated, as no pockets were detected for all structures in at least one ensemble.



**Figure S16.** Impact of the dilation radius ( $\sigma$ ) on the volume standard deviations in the groups of the NMR ensembles dataset. The parameters grid spacing ( $\delta$ ), density variance scaling ( $\omega$ ), and DoG density cutoff ( $clCutOff$ ) were kept constant. The parameter combinations with a DoG density cutoff of -0.1 and a density variance scaling of 4.0 were not evaluated, as no pockets were detected for all structures in at least one ensemble.

**Table S1.** Abbreviations and explanations of calculated descriptors of DoGSite3.

<b>Abbreviation</b>	<b>Explanation</b>
name	pocket identifier
lig_cov	percentage of ligand covered by the predicted pocket
poc_cov	percentage of the pocket covered by the co-crystallized ligand
lig_name	ligand identifier
4A_crit	1 or true if for a specific protein structure if the geometric center of the largest pocket lies within 4 Å of any ligand atom (Weisel criterion <sup>[1]</sup> )
ligSASRatio	bound ligand solvent-accessible surface divided by unbound ligand solvent accessible surface
volume	pocket volume in Å <sup>3</sup> calculated via grid points
enclosure	enclosure of pocket calculated via 1-(lid/hull)
surface	pocket surface in Å <sup>2</sup> calculated via solvent accessible surface area of pocket atoms
lipoSurface	pocket lipophilic surface in Å <sup>2</sup> calculated via solvent-accessible surface area of pocket atoms
depth	depth of the pocket in Å calculated via grid points and a depth-first search
surf/vol	ratio of surface to volume calculated via variables above
lid/hull	ratio of lid to hull calculated via grid points
ellVol	ellipsoid volume of pocket
ell c/a, ell b/a	ellipsoid main axes ratios, with a > b > c
surfGPs	number of grid points representing the surface of the pocket
lidGPs	number of grid points representing the lid of the pocket
hullGPs	sum of surface and lid grid points
spoc	number of subpockets
siteAtms	number of all pocket atoms
heavyAtms	number of all pocket heavy atoms
accept	number of solvent-accessible hydrogen bond acceptors
donor	number of solvent-accessible hydrogen bond donors
aromat	number of all aromatic atoms
hydrophobicity	hydrophobicity of pocket calculated via hydrophobic pocket atoms
metal	number of all metal ions
Cs, Ns, Os, Ss, Xs	number of elements of a specific type; types: C, N, O, S or other (X)
acidicAA	number of all acidic protein residues in pocket
basicAA	number of all basic protein residues in pocket
polarAA	number of all polar protein residues in pocket
apolarAA	number of all apolar protein residues in pocket
sumAA	number of protein residues in pocket
ALA, ARG, ...	number of protein residues in pocket; 3-letter code of the 20 proteinogenic amino acids
A, DA, ...	number of nucleic acid residues in pocket
UNK	number of unknown residues in pocket

**Table S2.** Hit rates of DoGSite3 (RLO  $\geq 0.5$ ) and alternative binding site identification tools for three datasets. The results for the alternative binding site identification tools were extracted from<sup>[2]</sup>.

Method	PDBbind-Derived Sites			Cryptic Sites			Allosteric Sites		
	<i>top1</i>	<i>top2</i>	<i>top3</i>	<i>top1</i>	<i>top2</i>	<i>top3</i>	<i>top1</i>	<i>top2</i>	<i>top3</i>
<b>BetaVoid</b>	3.84%	3.84%	3.84%	3.70%	4.32%	4.32%	5.44%	6.12%	6.12%
<b>CAVE</b>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<b>CAVITATOR</b>	4.23%	5.76%	7.69%	3.08%	4.32%	5.55%	0.68%	0.68%	4.08%
<b>CAVITY</b>	21.92%	31.15%	35.00%	3.08%	4.93%	5.55%	2.04%	4.76%	7.48%
<b>CAVITY SEARCH</b>	3.07%	3.46%	3.46%	1.23%	1.23%	2.46%	2.04%	2.04%	2.04%
<b>CLIPPERS</b>	18.84%	32.30%	43.46%	11.72%	22.22%	29.01%	10.20%	21.08%	27.21%
<b>ConCavity<sup>LIGSITE</sup></b>	29.61%	47.69%	55.76%	20.98%	33.33%	37.03%	13.60%	26.53%	33.33%
<b>ConCavity<sup>SURFNET</sup></b>	35.38%	56.92%	68.84%	9.25%	26.54%	34.56%	16.32%	29.25%	36.05%
<b>DoGSite2.0</b>	31.92%	40.76%	46.92%	11.11%	16.04%	18.51%	5.44%	10.20%	18.36%
<b>DoGSite (former default parameters)</b>	32.30%	40.00%	45.80%	11.10%	17.90%	21.00%	5.40%	10.00%	19.00%
<b>DoGSite3 (parameter set 1)</b>	63.46%	72.69%	74.23%	32.71%	44.44%	49.38%	27.89%	42.17%	51.02%
<b>DoGSite3 (parameter set 2)</b>	60.00%	71.15%	74.61%	30.86%	41.97%	46.91%	20.40%	34.01%	44.21%
<b>DoGSite3 (parameter set 5)</b>	56.53%	64.61%	67.30%	27.16%	34.56%	40.12%	25.17%	32.65%	44.21%
<b>dxTuber</b>	8.84%	11.92%	12.69%	4.93%	6.17%	9.25%	1.36%	2.72%	6.12%
<b>EDTSurf</b>	0.38%	1.92%	1.92%	0.61%	3.08%	3.70%	1.36%	2.04%	3.40%
<b>EPOS<sup>BP</sup></b>	40.00%	56.53%	63.84%	21.60%	27.77%	33.95%	12.92%	23.80%	31.29%
<b>fconv (Delaunay triangulation)</b>	23.84%	29.23%	31.15%	10.49%	12.96%	14.19%	4.76%	8.84%	12.24%
<b>fconv (grid-based)</b>	6.53%	9.61%	11.15%	0.61%	0.61%	1.23%	0.68%	2.04%	2.72%
<b>fpocket (version 1.0)</b>	40.76%	54.23%	57.69%	24.07%	30.24%	32.09%	14.96%	26.53%	34.69%
<b>fpocket (version 2.0)</b>	43.07%	53.07%	55.38%	25.30%	33.95%	39.50%	17.68%	25.85%	34.69%
<b>GetCleft</b>	32.69%	40.00%	44.61%	15.43%	24.07%	26.54%	4.08%	14.96%	19.72%
<b>ghecom</b>	43.07%	52.30%	55.00%	20.98%	25.30%	29.62%	6.80%	20.40%	29.93%
<b>KVFinder</b>	35.76%	43.07%	45.38%	9.87%	16.66%	17.90%	6.80%	11.56%	20.40%
<b>LIGSITE<sup>CS</sup></b>	42.69%	51.53%	55.00%	17.90%	24.69%	25.30%	13.60%	24.48%	32.65%
<b>McVol</b>	17.69%	20.76%	22.30%	5.55%	7.40%	7.40%	4.76%	6.80%	7.48%
<b>MOLE 2.0</b>	14.61%	18.07%	19.61%	9.25%	11.72%	14.81%	5.44%	9.52%	12.24%
<b>MSPocket</b>	45.76%	53.07%	56.53%	22.83%	30.24%	32.71%	8.16%	19.04%	25.85%
<b>NanoShaper</b>	42.30%	52.30%	53.46%	23.45%	33.95%	35.80%	17.00%	25.85%	31.97%
<b>PASS</b>	34.23%	48.46%	55.00%	12.34%	17.28%	19.13%	11.56%	19.72%	24.48%
<b>PHECOM</b>	46.15%	55.76%	58.07%	22.22%	30.24%	35.18%	10.20%	25.17%	34.69%
<b>PocketAnalyzer<sup>PCA</sup></b>	43.46%	46.53%	49.23%	19.75%	27.16%	28.39%	18.36%	27.89%	38.77%
<b>PocketDepth (depth factor)</b>	7.30%	20.00%	29.61%	6.17%	14.19%	21.60%	6.80%	10.88%	15.64%
<b>PocketDepth (non-polar atom count)</b>	21.53%	31.92%	39.61%	16.04%	23.45%	26.54%	8.16%	14.28%	22.44%
<b>PocketDepth (polar atom count)</b>	24.23%	37.30%	43.07%	14.81%	19.75%	23.45%	4.76%	10.20%	19.04%
<b>PocketDepth (size)</b>	22.30%	34.61%	41.15%	11.11%	17.28%	21.60%	6.80%	8.84%	17.68%
<b>PocketDepth (surface atom count)</b>	24.61%	37.69%	43.84%	15.43%	21.60%	24.69%	7.48%	11.56%	19.72%



Method	PDBbind-Derived Sites			Cryptic Sites			Allosteric Sites		
	<i>top1</i>	<i>top2</i>	<i>top3</i>	<i>top1</i>	<i>top2</i>	<i>top3</i>	<i>top1</i>	<i>top2</i>	<i>top3</i>
<b>PocketPicker</b>	44.23%	50.76%	56.92%	22.83%	30.86%	33.33%	14.28%	19.72%	28.57%
<b>PrinCCes</b>	50.76%	62.69%	67.30%	24.69%	32.71%	38.88%	14.28%	27.89%	40.13%
<b>SURFNET</b>	8.84%	14.61%	18.46%	3.70%	8.64%	8.64%	2.04%	5.44%	7.48%
<b>trj_cavity</b>	25.00%	31.92%	34.61%	6.17%	11.11%	13.58%	4.76%	8.16%	11.56%
<b>VisGrid</b>	25.38%	30.00%	30.38%	10.49%	14.19%	17.90%	8.84%	15.64%	21.08%
<b>Void</b>	38.07%	47.69%	51.92%	21.60%	29.01%	33.33%	9.52%	23.80%	30.61%
<b>VOIDOO</b>	13.07%	16.15%	17.30%	12.96%	12.96%	13.58%	8.84%	14.96%	14.96%
<b>VolSite</b>	47.69%	58.84%	63.84%	24.07%	28.39%	33.95%	12.92%	19.04%	27.89%
<b>Voronoia</b>	2.69%	2.69%	2.69%	3.08%	3.08%	3.08%	4.08%	4.76%	5.44%

**Table S3. Overview of benchmarked binding site prediction methods.** The method name and the availability are given (with the corresponding URL whenever applicable). The basic approach for pocket detection (not ranking) is provided (G for geometry-based, E for energy-based, K for knowledge-based, and C for approaches that combine the results of more than one method) and the year of publication.

Method	Availability	URL	Approach	Year
<b>VOIDOO</b> <sup>[3]</sup>	standalone	<a href="http://xray.bmc.uu.se/usf/">http://xray.bmc.uu.se/usf/</a>	G	1994
<b>SURFNET</b> <sup>[4]</sup>	standalone	<a href="http://www.ebi.ac.uk/thornton-srv/software/SURFNET/">http://www.ebi.ac.uk/thornton-srv/software/SURFNET/</a>	G	1995
<b>LIGSITE</b> <sup>[5]</sup>	not available	n/a	G	1997
<b>PASS</b> <sup>[6]</sup>	standalone	<a href="http://www.ccl.net/cca/software/UNIX/pass/overview.shtml">http://www.ccl.net/cca/software/UNIX/pass/overview.shtml</a>	G	2000
<b>LIGSITE</b> <sup>CS[7]</sup>	standalone	<a href="http://projects.biotech.tu-dresden.de/pocket/download.html">http://projects.biotech.tu-dresden.de/pocket/download.html</a>	G	2006
<b>CLIPPERS (Travel Depth)</b> <sup>[8]</sup>	standalone	<a href="https://github.com/ryancoleman/traveldistance">https://github.com/ryancoleman/traveldistance</a>	G	2006
<b>SiteMap</b> <sup>[9]</sup>	Schrödinger (standalone)	<a href="https://www.schrodinger.com/">https://www.schrodinger.com/</a>	E	2007
<b>EPOS</b> <sup>BP[10]</sup>	standalone	<a href="http://gepard.bioinformatik.uni-saarland.de/software/epos-bp">http://gepard.bioinformatik.uni-saarland.de/software/epos-bp</a>	G	2007
<b>PHECOM</b> <sup>[11]</sup>	standalone	<a href="http://strcomp.protein.osaka-u.ac.jp/ghecom/download_src.html">http://strcomp.protein.osaka-u.ac.jp/ghecom/download_src.html</a>	G	2007
<b>PocketPicker</b> <sup>[1]</sup>	standalone	<a href="http://gecco.org.chemie.uni-frankfurt.de/pocketpicker/download.html">http://gecco.org.chemie.uni-frankfurt.de/pocketpicker/download.html</a>	G	2007
<b>VisGrid</b> <sup>[12]</sup>	standalone	<a href="http://kiharalab.org/VisGrid/">http://kiharalab.org/VisGrid/</a>	G	2008
<b>PocketDepth</b> <sup>[13]</sup>	web server / standalone	<a href="http://proline.physics.iisc.ernet.in/pocketdepth/">http://proline.physics.iisc.ernet.in/pocketdepth/</a>	G	2008
<b>EDTSurf</b> <sup>[14]</sup>	standalone	<a href="https://zhanglab.ccmb.med.umich.edu/EDTSurf/">https://zhanglab.ccmb.med.umich.edu/EDTSurf/</a>	G	2009
<b>fpocket</b> <sup>[15]</sup>	standalone	<a href="https://sourceforge.net/projects/fpocket/files/fpocket-1.0/fpocket-src-1.0/">https://sourceforge.net/projects/fpocket/files/fpocket-1.0/fpocket-src-1.0/</a>	G	2009
<b>Voronoia</b> <sup>[16]</sup>	standalone	<a href="http://bioinformatics.charite.de/voronoia/index.php?site=download">http://bioinformatics.charite.de/voronoia/index.php?site=download</a>	G	2009
<b>SiteHound</b> <sup>[17]</sup>	standalone	<a href="http://scbx.mssm.edu/sitehound/sitehound-download/download.html">http://scbx.mssm.edu/sitehound/sitehound-download/download.html</a>	E	2009
<b>ConCavity</b> <sup>[18]</sup>	standalone	<a href="http://compbio.cs.princeton.edu/concavity/">http://compbio.cs.princeton.edu/concavity/</a>	C	2009
<b>DoGSite</b> <sup>[19]</sup>	standalone	<a href="https://www.biosolveit.de/DoGSiteScorer/">https://www.biosolveit.de/DoGSiteScorer/</a>	G	2010
<b>ghecom</b> <sup>[20]</sup>	standalone	<a href="http://strcomp.protein.osaka-u.ac.jp/ghecom/download_src.html">http://strcomp.protein.osaka-u.ac.jp/ghecom/download_src.html</a>	G	2010
<b>McVol</b> <sup>[21]</sup>	standalone	<a href="http://www.bisb.uni-bayreuth.de/index.php?page=data/mcvol/mcvol">http://www.bisb.uni-bayreuth.de/index.php?page=data/mcvol/mcvol</a>	G	2010
<b>fpocket2</b> <sup>[22]</sup>	standalone	<a href="https://sourceforge.net/projects/fpocket/files/latest/download">https://sourceforge.net/projects/fpocket/files/latest/download</a>	G	2010

Method	Availability	URL	Approach	Year
<b>CAVE</b> <sup>[23]</sup>	standalone	<a href="http://cpc.cs.qub.ac.uk/summaries/AEHC_v1_0.html">http://cpc.cs.qub.ac.uk/summaries/AEHC_v1_0.html</a>	G	2010
<b>fconv</b> <sup>[24]</sup>	standalone	<a href="http://pc1664.pharmazie.uni-marburg.de/download/">http://pc1664.pharmazie.uni-marburg.de/download/</a>	G	2011
<b>MSPocket</b> <sup>[25]</sup>	standalone	<a href="http://projects.biotech.tu-dresden.de/MSPocket/">http://projects.biotech.tu-dresden.de/MSPocket/</a>	G	2011
<b>PocketAnalyzer</b> <sup>PCA[26]</sup>	standalone	<a href="https://sourceforge.net/projects/papca/">https://sourceforge.net/projects/papca/</a>	G	2011
<b>Void (Union Ball)</b> <sup>[27]</sup>	upon request	n/a	G	2011
<b>dxTuber</b> <sup>[28]</sup>	standalone	<a href="https://github.com/willhelm-mueller/dxTuber">https://github.com/willhelm-mueller/dxTuber</a>	G	2011
<b>CAVITATOR</b> <sup>[29]</sup>	standalone	<a href="https://sites.gatech.edu/cssb/cavitator/">https://sites.gatech.edu/cssb/cavitator/</a>	G	2013
<b>CAVITY</b> <sup>[30]</sup>	web server / standalone	<a href="http://www.pkumdl.cn:8000/cavityplus/index.php#toolbox/">http://www.pkumdl.cn:8000/cavityplus/index.php#toolbox/</a> / <a href="http://repharma.pku.edu.cn/ligbuilder/download.html">http://repharma.pku.edu.cn/ligbuilder/download.html</a>	G	2013
<b>MOLE 2.0</b> <sup>[31]</sup>	standalone	<a href="https://webchem.ncbr.muni.cz/Platform/App/Mole">https://webchem.ncbr.muni.cz/Platform/App/Mole</a>	G	2013
<b>NanoShaper</b> <sup>[32]</sup>	standalone	<a href="https://www.electrostaticszone.eu/downloads">https://www.electrostaticszone.eu/downloads</a>	G	2013
<b>BetaVoid</b> <sup>[33]</sup>	standalone	<a href="http://voronoi.hanyang.ac.kr/software.htm#BetaVoid">http://voronoi.hanyang.ac.kr/software.htm#BetaVoid</a>	G	2014
<b>KVFinder</b> <sup>[34]</sup>	standalone	<a href="https://sourceforge.net/projects/kvfinder/">https://sourceforge.net/projects/kvfinder/</a>	G	2014
<b>trj_cavity</b> <sup>[35]</sup>	standalone	<a href="https://sourceforge.net/projects/trjcavity/files/latest/download">https://sourceforge.net/projects/trjcavity/files/latest/download</a>	G	2014
<b>GetCleft</b> <sup>[36]</sup>	standalone	<a href="http://biophys.umontreal.ca/nrg/NRG/Resources.html">http://biophys.umontreal.ca/nrg/NRG/Resources.html</a>	G	2015
<b>PrinCCes</b> <sup>[37]</sup>	standalone	<a href="https://github.com/CzirjakGabor/PrinCCes">https://github.com/CzirjakGabor/PrinCCes</a>	G	2015
<b>P2RANK</b> <sup>[38]</sup>	standalone	<a href="https://github.com/rdk/p2rank">https://github.com/rdk/p2rank</a>	K	2015
<b>AutoSite</b> <sup>[39]</sup>	standalone	<a href="http://adfr.scripps.edu/AutoDockFR/downloads.html">http://adfr.scripps.edu/AutoDockFR/downloads.html</a>	E	2016
<b>VolSite</b> <sup>[40]</sup>	standalone	<a href="http://bioinfo-pharma.u-strasbg.fr/labwebsite/download.html">http://bioinfo-pharma.u-strasbg.fr/labwebsite/download.html</a>	G	2012
<b>CAVITY SEARCH</b> <sup>[41]</sup>	standalone	<a href="https://sourceforge.net/projects/cavity-search/files/">https://sourceforge.net/projects/cavity-search/files/</a>	G	2011

## References

- [1] Weisel, M.; Proschak, E.; Schneider, G. PocketPicker: Analysis of ligand binding-sites with shape descriptors. *Chem. Cent. J.* **2007**, *1*, 7.
- [2] Ehrhart, C. Protein binding site comparison. Ph.D. thesis, Technische Universität Dortmund, 2019.
- [3] Kleywegt, G. J.; Jones, T. A. Detection, delineation, measurement and display of cavities in macromolecular structures. *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **1994**, *50* (Pt 2), 178–185.
- [4] Laskowski, R. A. SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions. *J. Mol. Graphics* **1995**, *13* (5), 323–30, 307–8.
- [5] Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins. *J. Mol. Graphics Modell.* **1997**, *15* (6), 359.
- [6] Brady, G. P.; Stouten, P. F. Fast prediction and visualization of protein binding pockets with PASS. *J. Comput.-Aided Mol. Des.* **2000**, *14* (4), 383–401.
- [7] Huang, B.; Schroeder, M. LIGSITE<sup>CSC</sup>: Predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct. Biol.* **2006**, *6*, 19.
- [8] Coleman, R. G.; Sharp, K. A. Travel depth, a new shape descriptor for macromolecules: Application to ligand binding. *J. Mol. Biol.* **2006**, *362* (3), 441–458.
- [9] Halgren, T. A. New method for fast and accurate binding-site identification and analysis. *Chem. Biol. Drug Des.* **2007**, *69* (2), 146–148.
- [10] Eyrich, S.; Helms, V. Transient pockets on protein surfaces involved in protein-protein interaction. *J. Med. Chem.* **2007**, *50* (15), 3457–3464.
- [11] Kawabata, T.; Go, N. Detection of pockets on protein surfaces using small and large probe spheres to find putative ligand binding sites. *Proteins* **2007**, *68* (2), 516–529.
- [12] Li, B.; Turuvekere, S.; Agrawal, M.; La, D.; Ramani, K.; Kihara, D. Characterization of local geometry of protein surfaces with the visibility criterion. *Proteins* **2008**, *71* (2), 670–683.
- [13] Kalidas, Y.; Chandra, N. PocketDepth: A new depth based algorithm for identification of ligand binding sites in proteins. *J. Struct. Biol.* **2008**, *161* (1), 31–42.
- [14] Xu, D.; Zhang, Y. Generating triangulated macromolecular surfaces by Euclidean Distance Transform. *PLoS ONE* **2009**, *4* (12), e8140.
- [15] Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinf.* **2009**, *10*, 168.
- [16] Rother, K.; Hildebrand, P. W.; Goede, A.; Gruening, B.; Preissner, R. Voronoia: Analyzing packing in protein structures. *Nucleic Acids Res.* **2009**, *37* (Database issue), 5.
- [17] Ghersi, D.; Sanchez, R. EasyMIFS and SiteHound: A toolkit for the identification of ligand-binding sites in protein structures. *Bioinformatics* **2009**, *25* (23), 3185–3186.
- [18] Capra, J. A.; Laskowski, R. A.; Thornton, J. M.; Singh, M.; Funkhouser, T. A. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLoS Comput. Biol.* **2009**, *5* (12), e1000585.
- [19] Volkamer, A.; Griewel, A.; Grombacher, T.; Rarey, M. Analyzing the topology of active sites: on the prediction of pockets and subpockets. *J. Chem. Inf. Model.* **2010**, *50* (11), 2041–2052.
- [20] Kawabata, T. Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins* **2010**, *78* (5), 1195–1211.
- [21] Till, M. S.; Ullmann, G. M. McVol - A program for calculating protein volumes and identifying cavities by a Monte Carlo algorithm. *J. Mol. Model.* **2010**, *16* (3), 419–429.
- [22] Schmidtke, P.; Barril, X. Understanding and predicting druggability. A high-throughput method for detection of drug binding sites. *J. Med. Chem.* **2010**, *53* (15), 5858–5867.
- [23] Buša, J.; Hayryan, S.; Hu, C.-K.; Skřivánek, J.; Wu, M.-C. CAVE: A package for detection and quantitative analysis of internal cavities in a system of overlapping balls: Application to proteins. *Comput. Phys. Commun.* **2010**, *181* (12), 2116–2125.
- [24] Neudert, G.; Klebe, G. fconv: Format conversion, manipulation and feature computation of molecular data. *Bioinformatics* **2011**, *27* (7), 1021–1022.

- [25] Zhu, H.; Pisabarro, M. T. MSPocket: An orientation-independent algorithm for the detection of ligand binding pockets. *Bioinformatics* **2011**, *27* (3), 351–358.
- [26] Craig, I. R.; Pflieger, C.; Gohlke, H.; Essex, J. W.; Spiegel, K. Pocket-space maps to identify novel binding-site conformations in proteins. *J. Chem. Inf. Model.* **2011**, *51* (10), 2666–2679.
- [27] Mach, P.; Koehl, P. Geometric measures of large biomolecules: Surface, volume, and pockets. *J. Comput. Chem.* **2011**, *32* (14), 3023–3038.
- [28] Raunest, M.; Kandt, C. dxTuber: Detecting protein cavities, tunnels and clefts based on protein and solvent dynamics. *J. Mol. Graphics Modell.* **2011**, *29* (7), 895–905.
- [20] Gao, M.; Skolnick, J. APoc: Large-scale identification of similar protein pockets. *Bioinformatics* **2013**, *29* (5), 597–604.
- [30] Yuan, Y.; Pei, J.; Lai, L. Binding site detection and druggability prediction of protein targets for structure-based drug design. *Curr. Pharm. Des.* **2013**, *19* (12), 2326–2333.
- [31] Sehnal, D.; Svobodová Vařeková, R.; Berka, K.; Pravda, L.; Navrátilová, V.; Banáš, P.; Ionescu, C.-M.; Otyepka, M.; Koča, J. MOLE 2.0: Advanced approach for analysis of biomacromolecular channels. *J. Cheminf.* **2013**, *5* (1), 39.
- [32] Decherchi, S.; Rocchia, W. A general and robust ray-casting-based algorithm for triangulating surfaces at the nanoscale. *PLoS ONE* **2013**, *8* (4), e59744.
- [33] Kim, J.-K.; Cho, Y.; Laskowski, R. A.; Ryu, S. E.; Sugihara, K.; Kim, D.-S. BetaVoid: Molecular voids via  $\beta$ -complexes and Voronoi diagrams. *Proteins* **2014**, *82* (9), 1829–1849.
- [34] Oliveira, S. H. P.; Ferraz, F. A. N.; Honorato, R. V.; Xavier-Neto, J.; Sobreira, T. J. P.; de Oliveira, Paulo S L. KVFinder: Steered identification of protein cavities as a PyMOL plugin. *BMC Bioinf.* **2014**, *15*, 197.
- [35] Paramo, T.; East, A.; Garzón, D.; Ulmschneider, M. B.; Bond, P. J. Efficient characterization of protein cavities within molecular simulation trajectories: Trj\_cavity. *J. Chem. Theory Comput.* **2014**, *10* (5), 2151–2164.
- [36] Gaudreault, F.; Morency, L.-P.; Najmanovich, R. J. NRGsuite: a PyMOL plugin to perform docking simulations in real time using FlexAID. *Bioinformatics* **2015**, *31* (23), 3856–3858.
- [37] Czirájk, G. PrinCCes: Continuity-based geometric decomposition and systematic visualization of the void repertoire of proteins. *J. Mol. Graphics Modell.* **2015**, *62*, 118–127.
- [38] Krivák, R.; Hoksza, D. P2RANK: Knowledge-based ligand binding site prediction using aggregated local features. In *Algorithms for Computational Biology*; Dediu, A.-H., Hernández-Quiroz, F., Martín-Vide, C., Rosenblueth, D. A., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2015; pp 41–52.
- [39] Ravindranath, P. A.; Sanner, M. F. AutoSite: An automated approach for pseudo-ligands prediction-from ligand-binding sites identification to predicting key ligand atoms. *Bioinformatics* **2016**, *32* (20), 3142–3149.
- [40] Desaphy J.; Azdimousa K.; Kellenberger E.; Rognan D. Comparison and druggability prediction of protein-ligand binding sites from pharmacophore-annotated cavity shapes. *J. Chem. Inf. Model.* **2012**, *52* (8), 2287-2299.
- [41] Khangulov, V. Protein Cavity Search. <https://sourceforge.net/projects/cavity-search/files/> (accessed 24.03.2022).



## F.2 Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures

- [D2] **Graef, J.**, Ehrt, C., Diedrich, K., Poppinga, M., Ritter, N., Rarey, M., Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures. *J. Med. Chem.* 65.2 (2022), S. 1384–1395. DOI: 10.1021/acs.jmedchem.1c01046.

Reprinted with permission from [D2]. Copyright 2021 American Chemical Society.

## Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures

Joel Graef, Christiane Ehrt, Konrad Diedrich, Martin Poppinga, Norbert Ritter, and Matthias Rarey\*

Cite This: *J. Med. Chem.* 2022, 65, 1384–1395

Read Online

ACCESS |



Metrics &amp; More



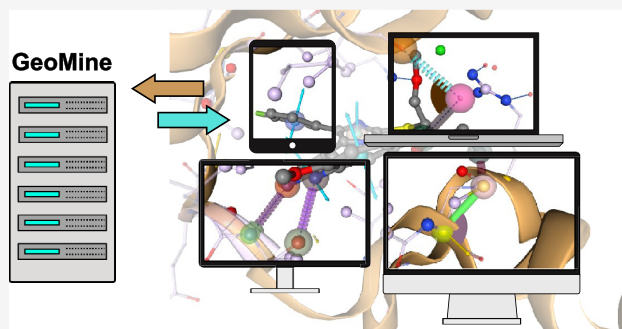
Article Recommendations



Supporting Information

**ABSTRACT:** The ever-growing number of protein-ligand complex structures can give fundamental insights into protein functions and protein-ligand interactions, especially in the field of protein kinase research. The number of tools to mine this data for individually defined structural motifs is restricted due to the challenging task of developing efficient index structures for 3D data in relational databases. Herein we present GeoMine, a database system with web front-end mining of more than 900 000 binding sites. It enables database searches for geometric (interaction) patterns in protein-ligand interfaces by, for example, textual, numerical, substructure, similarity, and 3D searches. GeoMine processes reasonably selective user-defined queries within minutes.

We demonstrate its usability for advancing protein kinase research with a special emphasis on unusual interactions, their use in designing selective kinase inhibitors, and the analysis of reactive cysteine residues that are amenable to covalent kinase inhibitors. GeoMine is freely available as part of our modeling support server at <https://proteins.plus>.



## INTRODUCTION

The analysis of protein-ligand interactions is a key element for understanding the structure-to-function relationships and selectivity profiles of protein kinase inhibitors. The identification and optimization of small-molecule binders as a central task in early drug discovery relies on the detailed knowledge of molecular recognition. Therefore, an analysis and comparison of spatial arrangements in protein-ligand interfaces is of the utmost relevance for life science research. Searching for spatial atomic arrangements is highly useful for numerous applications. Novel ligands can be designed by employing similar binding sites, allowing the transfer of interacting functional groups from one ligand to another. Similar geometric patterns can give significant insights into a protein's function and selectivity. Analyzing the environment of functional groups helps to obtain a better understanding of interaction geometries, to name just the most intuitive use cases that come to mind.

Due to the rising number of protein structures and known protein kinase-ligand complexes, efficient three-dimensional (3D) search algorithms are required but are challenging to develop. Even with a focus on a single protein class, proteins share structural similarities such that data analysis tools should be able to cope with the entire Protein Data Bank (PDB).<sup>1</sup> The search method of choice must enable a search for spatial atomic arrangements in a reasonable period of time. In addition, the search must be flexible so that complex queries can be composed for multiple structural features as well as classical textual, numerical, and substructural elements. The method must be chemistry-aware to allow detailed atomic interaction modeling.

Finally, a web application is desirable to avoid complex installation processes and offer an easy-to-use interface.

Up to now, several tools have been developed to search for geometric spatial arrangements, including CSD-CrossMiner,<sup>2</sup> PRDB,<sup>3</sup> PROLIX,<sup>4</sup> Relibase and Relibase+,<sup>5</sup> PDBeMotif and MSDmotif,<sup>6</sup> PELIKAN,<sup>7</sup> and GSP4PDB.<sup>8</sup> Recently, a new motif search was proposed by Bittrich et al. called strucomotif-search, which uses an efficient index for inter-amino acid distances.<sup>9</sup> The differences between all these tools can be roughly divided into the following five categories: (1) query variability, in other words, which types of textual, numerical, chemical, and geometric features can be used to create a search query. This also refers to the precision of the query, i.e., whether fuzziness can be introduced; (2) structure database, i.e., which data collections are available for search; (3) data processing and storage, i.e., how the data is extracted from raw PDB<sup>1</sup> files and what database technologies are used; (4) search capabilities and algorithms, i.e., how the query is evaluated and the precision of the search, and (5) result presentation, i.e., how results are reported and visualized.

**Special Issue:** New Horizons in Drug Discovery - Understanding and Advancing Kinase Inhibitors

**Received:** June 11, 2021

**Published:** September 7, 2021





Table 1 gives an overview of the existing tools related to their query variability and searchable structures. More detailed information can be found in Tables S3 and S4. Overall, the two most prominent disadvantages are that some tools are not publicly available and others are even discontinued and retired, e.g., Relibase. In addition, some of the tools are desktop applications that have to be installed, and the databases have to be updated in a semiautomated fashion. None of the tools include predicted sites, solvent exposure, and secondary structure information. An overview of the capabilities of the mentioned tools is given in Table 2.

In the following, we will introduce our new approach GeoMine, a flexible, geometrically reliable, and efficient method to search for spatial arrangements in protein-ligand interfaces and predicted binding sites and apply them to several use cases related to protein kinase research. Based on our earlier desktop application PELIKAN, GeoMine combines a flexible relational database with efficient search algorithms and a new easy-to-use web-based front-end for query generation and result browsing. This enables structural investigations on protein binding sites on-the-fly and offers a user-friendly environment for efficiently searching the binding site space. Several experiments demonstrate the performance and reliability of the search engine.

## RESULTS AND DISCUSSION

GeoMine offers a huge variety of possibilities to mine protein-ligand complexes and binding sites. A full description of GeoMine's comprehensive search capabilities can be found in the Experimental Section. The following applications of the tools toward protein kinase inhibitor design highlight the impact of these features for the structural investigations of protein kinase binding sites and the subsequent runtime analyses, showing that such analyses can be performed within seconds to minutes and offering a new way to work with large amounts of structural data in an interactive and easy-to-use manner. All queries described below are available in a machine-readable format in the Supporting Information.

**Exploiting Unusual Interactions for Selective Inhibitor Design.** During protein kinase inhibitor design, the question might arise as to whether previously unexplored unusual protein-ligand interactions play a major role in protein kinase selectivity. Based on a comprehensive study of such interactions,<sup>10</sup> we applied GeoMine to scan available protein-ligand complex structures for such interactions. We investigated the occurrence of interactions involving halogen atoms attached to aromatic rings with aromatic ring systems in protein side chains. To this end, we searched for appropriate geometric arrangements in all ligand-occupied pockets as stored in the PDB. As a template, we chose the structure with the PDB-ID 3q3k.<sup>11</sup> We defined three points: the aromatic ring center of the Tyr228 side chain, the location of the chlorine atom, and the aromatic ring center of the aromatic ring the halogen atom is attached to. All points were connected by distances with a tolerance of 2 Å for the distance between the halogen atom and the aromatic center of a protein residue's side chain (distance  $d_1$ ) and tolerances of 1 Å for the remaining two distances. To model the relative orientation of the aromatic ring systems, we define two angles  $\alpha$  and  $\beta$  with tolerances of 40° and 15°, respectively. Additionally, we reduce this search to pockets occupied by ligands with chlorine atoms. This search retrieved 695 3D matches in 629 pockets of 491 PDB structures (calculation time of 1.24 min). The results of the analysis are depicted in Figure 1 (top). A visual inspection of the results showed expected

Table 1. Overview of Existing Tools for 3D Searching in Structural Data Related to Query Variability and Precision<sup>a</sup>

	query variability	structure database
Relibase/Relibase+ (availability: not available)	atom-level precision, constraint ranges, nongeometric attributes	PDB, no information about adding own structure files, BS = 7 Å of the ligand atoms
PDBeMotif/MSDMotif (availability: public, web server)	atom-level precision only for ligands	PDB, based on user-defined selection of structure files, BS = 16 Å of the ligand atoms
PRDB (availability: not available)	distances between amino acids only between $\alpha$ carbons, nongeometric attributes	PDB, no information about adding own structure files, BS = 8 Å of the ligand atoms
PROLIX (availability: not available)	atom-level precision only for ligands, constraint range	PDB, Roche in-house X-ray structure database, no information about adding own structure files, BS = 4.5 Å of the ligand atoms
CSD-CrossMiner (availability: commercial, standalone)	only predefined feature types, pharmacophore query with user-defined tolerance spheres, nongeometric attributes, no angles	CSD, PDB (ligand-based binding sites), based on user-defined selection of structure files, BS = 6 Å of the ligand atoms
PELIKAN (availability: academic use, standalone)	atom-level precision, constraint ranges, nongeometric attributes	precompiled databases such as sCPDB 2013 and PDB (November 2016) available on the ZBH web site, structure files, BS = 6.5 Å of the ligand atoms
GSP4PDB (availability: public, web server)	ligands as three-letter codes or "any"; protein as one of the 20 natural amino acids or "undefined", "any", "polar", etc.; constraint ranges, gaps, and next in sequence; nongeometric attributes	PDB, BS = 7 Å of the ligand atoms
strucmotif-search (availability: public, web server (research collaboration for structural bioinformatics, RCSB), standalone)	amino acid defined by the distance between $\alpha$ -carbons as the side chain and $\beta$ -carbons as backbone representatives, nongeometric attributes using RCSB functions	PDB, structure files in standalone, distances between amino acid pairs 15 Å at maximum

<sup>a</sup>BS, binding site.

Table 2. Overview of GeoMines Capabilities in Contrast to Other Existing Tools for 3D Searching<sup>a</sup>

tool or GeoMine capabilities	user-defined angles	interaction detection	protonation	solvent exposure	atom-based queries	SSEs	ligand features	non-geometric filters	predicted sites	GUI
Relibase/Relibase+	x		x (ligand)		x		x	x	x	
PDBeMotif/ MSDmotif		x	x		x <sup>b</sup>		x			x
PRDB	x				x		x	x		x (ADOpt)
PROLIX		x								x
CSD-CrossMiner					x			x		x
PELIKAN	x	x	x		x		x	x		x
GSP4PDB(2)								x		x
strucmotif-search							x (via PDB)	x (via PDB)		x

<sup>a</sup>SSEs, secondary structure elements; GUI, graphical user interface. <sup>b</sup>Restricted to C $\alpha$  and the end of side chains.

interaction geometries between aromatic side chains and chlorine atoms attached to aromatic (hetero)cycles. We found that 51% of the ligands interact with tyrosine side chains and 26% interact with phenylalanine side chains, and we also found interactions with tryptophan (17%) and histidine side chains (5%). An analysis of the distance and angle ranges revealed broadly spread distributions of the distance  $d_1$  and the angle  $\alpha$ . Based on these findings, we can refine our query ( $d_1 = 4 \pm 1 \text{ \AA}$ ,  $\alpha = 25^\circ \pm 25^\circ$ ) and apply this query to search in the published structures assembled in the third version of the Kinase-Ligand Interaction Fingerprints and Structures (KLIFS) database<sup>12</sup> (<https://klifs.net/>)<sup>13</sup> to investigate the role of this interaction in protein kinases. The GeoMine search with the adjusted query resulted in 45 3D matches in 37 pockets of 34 PDB structures (0.42 min). Many of the identified interactions are located in binding sites in complex with well-known selective kinase inhibitors (Figure 1, mid), e.g., CDK2 in complex with the chemical probe 5,6-dichlorobenzimidazole-1- $\beta$ -D-ribofuranoside (PDB-ID 3my5<sup>14</sup>), but we could also identify small fragments that seem to undergo halogen-aromatic interactions, e.g., DYRK1A in complex with a chlorobenzothiazole fragment (PDB-ID 5a4q<sup>15</sup>). Concerning the type of residues undergoing these interactions, we find that approximately 51% involve phenylalanine and 27% involve tyrosine side chains as the most important interaction partners. Many of the hits indicate the usability of this interaction type for selective inhibition of protein kinases.

As a consequence, the question arises whether there already exist known inhibitors of a protein kinase of interest with an aromatic ring in the proximity of an aromatic side chain. Such an aromatic ring might be exploited as a selectivity anchor by adding chlorine atoms. To this end, the query had to be slightly adjusted, and instead of the chlorine atom we chose the attached carbon atom for the query and adjusted the distances according to the bond length of an aromatic carbon-chlorine bond of approximately 1.7  $\text{\AA}$  (see Figure 1, mid). We further restricted the character of the ligand carbon atom to an aromatic carbon atom with exactly two connections, excluding implicit hydrogen atoms (SMARTS pattern “[c;D2]”). This ensured that the aromatic carbon atoms of the retrieved ligands are not connected to halogen atoms and are instead connected to a hydrogen atom. A search with this query in all ligand-bound protein kinase structures retrieved 218 pockets of 178 PDB structures (1.23 min), highlighting that there are numerous kinase inhibitors with known binding modes that could be used as potential starting points to improve the compound selectivity.

For this showcase study, we picked cyclin-dependent-like kinase 5 (CDK5) as an example for a pharmacologically relevant

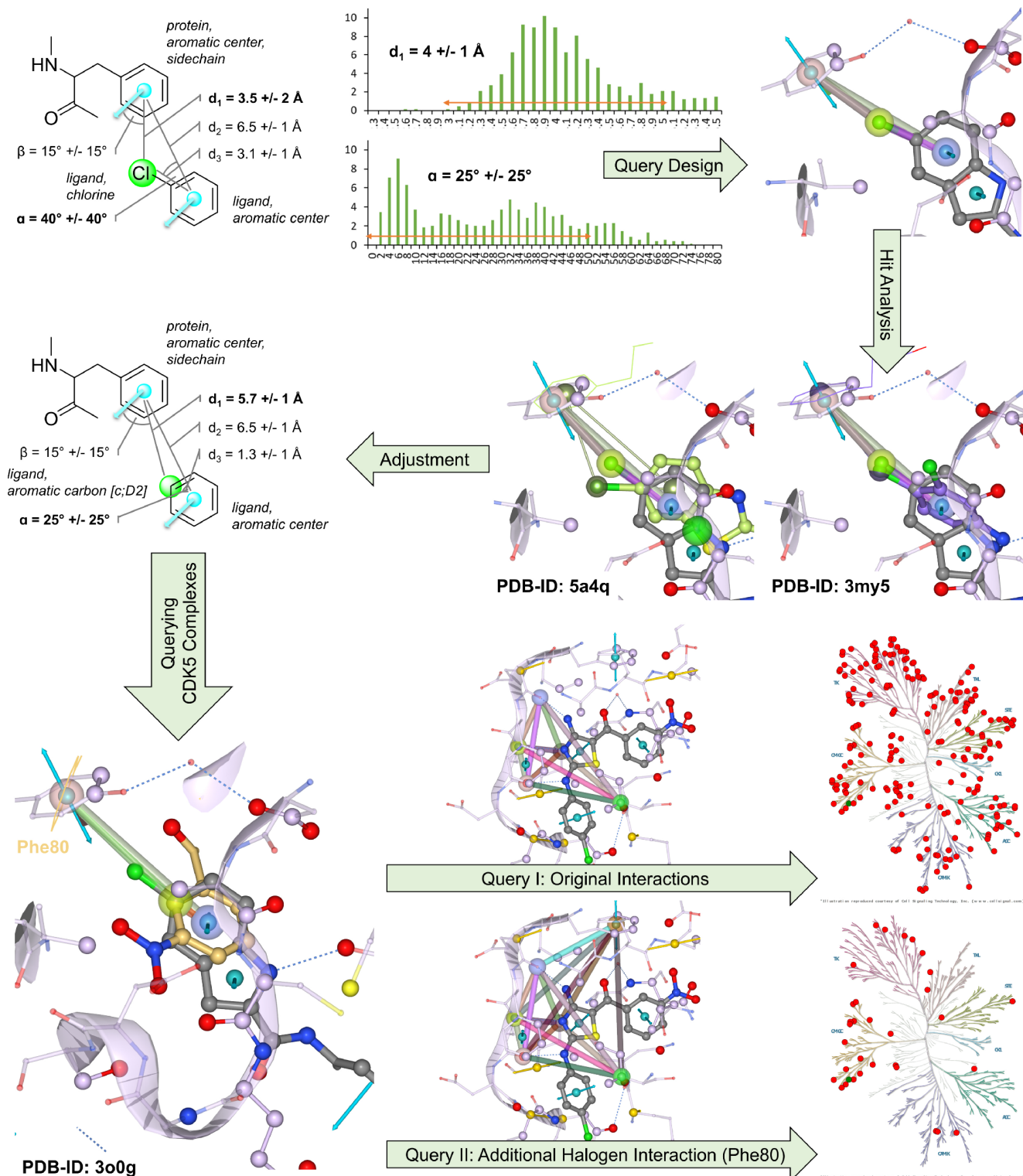
protein kinase target<sup>16</sup> and searched with the pattern against all available complex structures of this target. This search retrieved a structure of CDK5 in complex with an ATP analogue (PDB-ID 3o0g) with a K<sub>i</sub> value of 600 nM, which is likely not highly selective<sup>17</sup> but which might become more selective upon addition of a chlorine atom that interacts with Phe80 (Figure 1, bottom).

To support this hypothesis, we constructed a query for this site of CDK5 with the binding site properties, which are crucial for the interaction with this inhibitor (Figure 1, bottom). This time, we did not restrict our search to ligand-occupied pockets but searched for the pocket feature arrangement in all binding sites (predicted and ligand-based) of human protein kinases to identify as many potential off-targets as possible. This search retrieved 4801 3D matches in 3287 pockets of 2322 PDB structures (23.02 min, 227 distinct protein kinases have similar geometric arrangements of potentially interacting residues), highlighting the expected missing selectivity. However, the addition of the aromatic center of Phe80 as an additional interaction point for a potential halogen aromatic interaction reduced the number of hits by a factor of 8 (616 3D matches in 494 pockets of 392 PDB structures, 9.58 min, 33 distinct protein kinases are still similar based on the GeoMine query), pinpointing residue Phe80 as a potential selectivity anchor. Although there are still several similar protein kinase binding sites, they belong to similar kinase families, and additional ideas for further selectivity anchors can be derived from a visual inspection of the corresponding superimpositions. This example not only highlights the applicability of GeoMine for selectivity profiling but also the importance of enabling user-specific query design and database profiling in a large-scale manner.

Taken together, this showcase study exemplifies the capabilities of the GeoMine tool as an idea generator for protein kinase drug design. In the first step, typical and unusual interaction patterns can be explored and investigated. Subsequently, the queries can be adapted to find potential starting points for selective inhibitor design through the establishment of such interactions based on already known protein kinase complex structures. Moreover, GeoMine might also assist in improving our understanding of potential reasons for inhibitor selectivity, as discussed in the next example.

#### Searching for Selectivity Anchors in Protein Kinases.

GeoMine can also assist in the search for potential off-targets. This is not restricted to potential off-target structures with bound ligands, as empty predicted pockets are also included in the database. As an example, we picked two structures of epidermal growth factor receptor (EGFR) in complex with two well-characterized inhibitors. For the first structure of EGFR in



**Figure 1.** GeoMine as an idea generator for advancing protein kinase inhibitor design. In the presented workflow, the user will first search for a certain protein-ligand interaction pattern. In this case, we investigated the uncommon interaction between chlorine atoms and aromatic ring systems (top). Based on the derived geometric data, the user can adjust the query accordingly and search for this interaction type in a predefined set of ligand-bound structures (mid). After visual inspection, the user can subsequently try to find potential starting points for exploiting this interaction type to advance known inhibitors with respect to selectivity. In our showcase study, we picked the structure of CDK5 in complex with an ATP analogue (bottom). The interacting residues of the original compound are spread across the whole kinome. However, the inclusion of an interaction with Phe80 of CDK5 might lead to an improved selectivity profile, as highlighted by the GeoMine hits of the corresponding queries using KinMap.<sup>18</sup>

complex with the inhibitor gefitinib (PDB-ID 4wkq), we generated a query consisting of the protein's relevant interacting

residues, i.e., a backbone nitrogen atom as hydrogen bond donor in the hinge region (Met793), a hydrophobic side chain carbon

atom in the N-lobe (Ala743), a hydrophobic side chain carbon atom in the C-lobe (Leu844), and a hydrophobic side chain carbon atom of Lys745. This ligand-independent search query was used to screen all known protein kinases as stored in the KLIFS database but was restricted to structures from the organism *Homo sapiens*. The search resulted in 10 918 3D matches in 4451 pockets of 3049 PDB structures, pinpointing unselective inhibition (1.45 min). Subsequently, this initial query was extended by one side chain of residue Thr790 as a potential selectivity anchor as known from the structure of EGFR in complex with lapatinib (PDB-ID 1xkk<sup>19</sup>). Intriguingly, this search retrieved only 32 3D matches in 22 pockets of 20 PDB structures (1.16 min), highlighting the importance of Thr790 as a potential selectivity-introducing residue. This is in line with the finding that mutation T790 M leads to a significant increase in the IC<sub>50</sub> values for the highly selective inhibitor lapatinib (IC<sub>50</sub>(EGFR) = 4.9 nM, IC<sub>50</sub>(T790M) = 850 nM, and IC<sub>50</sub>(T790M/L858R) = 8500 nM).<sup>20</sup> A comparison of the hits with additional kinase profiling data<sup>20</sup> further underlines the validity of the result (Table 3).

**Table 3. Selectivity Profiling Results for Two Different EGFR GeoMine Queries<sup>a</sup>**

protein kinase	gefitinib		lapatinib	
	GeoMine result based on the query for 4wkq	IC <sub>50</sub> (nM) <sup>20</sup>	GeoMine result based on the query for 1xkk <sup>19</sup>	IC <sub>50</sub> (nM) <sup>20</sup>
EGFR	hit	0.51	hit	4.9
ERBB2 (ErbB2)	hit	3100	hit	9.8
ERBB4 (ErbB4)	hit	7.6	hit	24
LCK	hit	390	hit	n.d.
LYN	hit	350	hit	n.d.
DDR1	hit	37	not found	4400
DDR2	hit	570	not found	n.d.
EPHA5 (EphA5)	hit	740	not found	n.d.
EPHA7 (EphA7)	hit	990	not found	n.d.
EPHA8 (EphA8)	hit	730	not found	n.d.
EPHB2 (EphB2)	hit	890	not found	n.d.
EPHB4 (EphB4)	hit	420	not found	n.d.
FLT3	hit	730	not found	n.d.
MKMK1 (MNK1)	hit	130	not found	n.d.
MKMK2 (MNK2)	hit	150	not found	n.d.
PDGFRA (PDGFRa)	hit	600	not found	n.d.
PTK6 (BRK)	hit	860	not found	1100
SLK	hit	1300	not found	n.d.
STK10 (LOK)	hit	430	not found	n.d.
TNK2 (ACK)	hit	1100	not found	n.d.

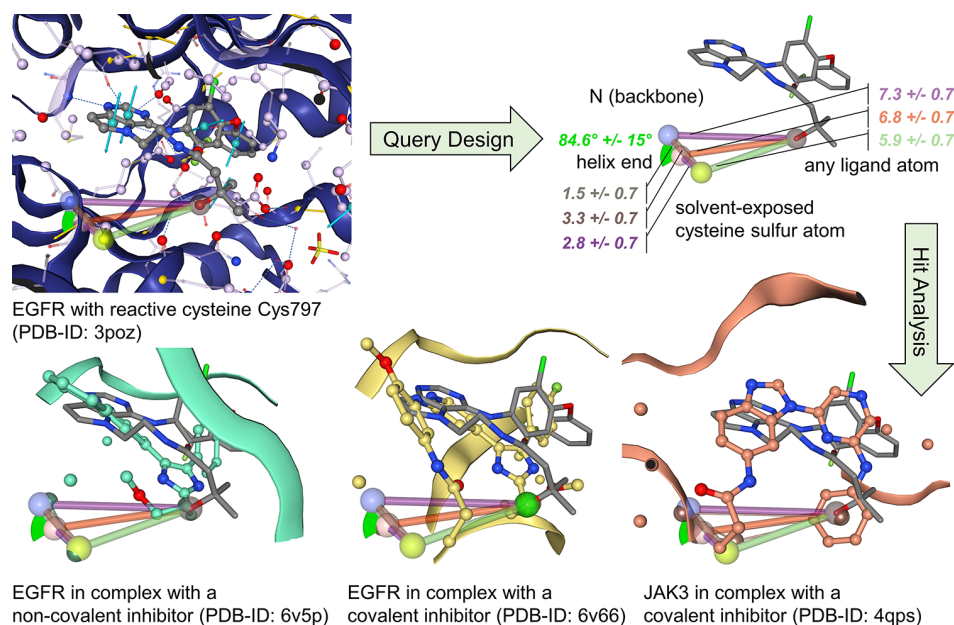
<sup>a</sup>The queries were based on the structure in complex with gefitinib (PDB-ID 4wkq; interacting residues Met793, Ala743, Leu844, and Lys745) and lapatinib (PDB-ID 1xkk; interacting residues Met793, Ala743, Leu844, Lys745, Thr790).

### Searching for Reactive Cysteines in Protein Kinases.

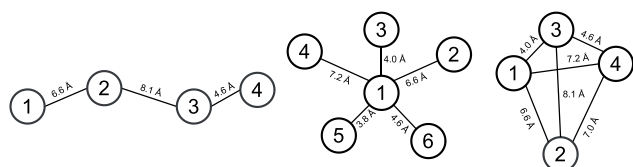
GeoMine enables the inclusion of secondary structure features in the search query. A very prominent example of how to use this feature is the search for compounds in the vicinity of reactive cysteine residues for the structure-based design of covalent inhibitors. Cysteine residues at the amino terminus of  $\alpha$ -helices are frequently characterized by a high nucleophilicity.<sup>21</sup> This fact is often exploited for the design of covalent protein kinase inhibitors.<sup>22</sup> In our last example, we demonstrate how the inclusion of secondary structure elements and solvent exposure enables the search for protein kinases with reactive cysteines in the neighborhood of known inhibitors. Based on the crystal structure of the protein kinase EGFR with reactive cysteine Cys797,<sup>23</sup> we constructed a query to search for known kinase inhibitors in the proximity of an amino-terminal solvent-exposed cysteine (PDB-ID 3poz<sup>24</sup>). This query consists of the solvent-exposed cysteine sulfur atom, the cysteine's backbone nitrogen atom, the amino-terminal helix end, and any ligand atom at well-defined distance intervals from these points (see Figure 2, top). Additionally, we used the angle between the helix vector and the segment between the helix terminus and the backbone nitrogen atom to further restrict the search. We used this query to search in a PDB subselection that contained all PDB-IDs of protein kinases in the KLIFS database.<sup>12</sup> The query resulted in 75 3D matches in 54 pockets of 36 PDB structures and took 54 s. The results not only reveal promising inhibitors that can be extended by suitable reactive groups, so-called covalent warheads,<sup>25</sup> to address the reactive cysteine in EGFR, but also hint at other protein kinases, e.g., janus kinase 3 (JAK3), that also harbor a reactive cysteine in this position and might be potential off-targets for covalent inhibitors to address this cysteine residue (Figure 2, bottom).

**Comparison to Other Methods.** None of the three exemplary analyses could be performed by any other tool listed in Table 2, hampering a comparison for the selected examples. The search for unusual protein-ligand interactions by other tools is prevented by the missing functionality to restrict the geometric search by angles. PRDB is the only database that enables the definition of angles in the search query. However, this database is no longer accessible. In our second example, we define a query with interaction features, e.g., a hydrophobic residue atom. PROLIX is the only other tool that enables the use of such features. However, the definition of queries is restricted to distinct residues such that matching is only possible between identical residues. The main limitation of similar methods for the analyses of reactive amino-terminal cysteine residues, as shown in our last example, is the lack of ability to include secondary structure information in the search queries. GeoMine combines the unique strengths of the individual tools in Table 2, thereby creating a versatile user-friendly method for multiple purposes.

**Query Computing Time.** The performance of GeoMine was tested with a standard PELIKAN benchmark<sup>7</sup> on all protein-ligand complexes in the scPDB<sup>28,29</sup> (2017 version). The database was constructed with only those PDB files that contained at least one reference ligand (16 561 PDB files), and unoccupied binding sites were excluded for comparability to PELIKAN. All queries were designed using the protein-ligand complex with the PDB-ID 1j7u<sup>30</sup> so that every query resulted in at least one match. There are the following three kinds of geometric queries: (1) four points that are linearly arranged, (2) six points in a star shape, and (3) four points in a tetrahedron shape (see Figure 3). The point-point constraints are used with a



**Figure 2.** Profiling for protein kinases with reactive cysteines at the amino terminus of helices. (Top) The query was generated based on the structure of EGFR with a small-molecule inhibitor (PDB-ID 3poz<sup>24</sup>). The query consists of four points, six distances, and one angle characteristic for reactive cysteine in the proximity of ligands. (Bottom) Selected hits are presented here. (Left) Structures of EGFR in complex with a noncovalent and covalent inhibitor (PDB-IDs 6v5p and 6v66, respectively). (Right) Structure of JAK3 harboring a reactive cysteine at the same location in complex with a covalent inhibitor as potential off-target (PDB-ID 4qps<sup>25</sup>).



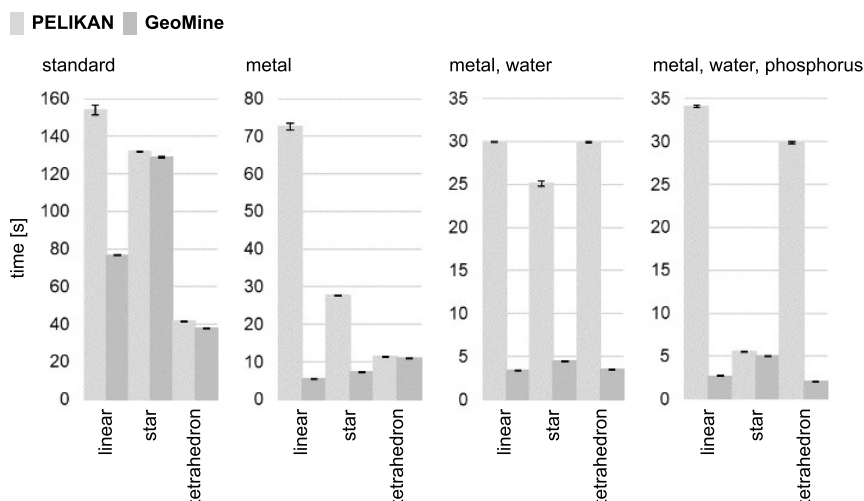
**Figure 3.** Geometric layout of the test queries for the runtime analysis, which are linear, star-shaped, or tetrahedral. Numbered points represent the search points with their IDs. Black lines describe the distance constraints. Queries were created using the PDB structure 1j7u as template structure.

tolerance of 0.5 Å. All three query types are used with different element types to assess the influence of specific attributes. The “standard” queries consist of oxygen and nitrogen atoms and, in case of the star-shaped queries, a carbon atom; the “metal” queries are those where one of the query points is changed to a magnesium ion; the “metal, water” queries are those with a magnesium ion and a water query point, and “metal, water, phosphorus” queries were those where another point is changed to a phosphorus atom. More detailed definitions of all test queries can be found in the [Supporting Information](#) (see Paragraph S2).

GeoMine and PELIKAN use different database technologies. While SQLite writes and reads data directly from disk, PostgreSQL has to establish a database connection first. In general, this network overhead is larger than that when reading from the solid-state drive. GeoMine was up to 10× faster than PELIKAN (see [Figure 4](#)) in our runtime tests. This is due to reimplemented key functionalities in the C++ code, adapted structure query language (SQL) queries, such as using JOINs on smaller parts of tables and more bundled database transactions, and making more use of the hardware by exploiting the advantages of the PostgreSQL database system.

In particular, queries with a linear topology are much faster, which is due to the lower number of distance constraints. In general, the distance constraint checks are the slowest in PELIKAN and GeoMine. Although the time required for this step was reduced with our query optimizations, it remains the most time-consuming one. The queries named “standard” are overall the slowest because they consist of carbon and nitrogen atoms, which are the most common elements in the database. Therefore, there are numerous possible points where all distance and angle constraints have to be checked. Overall, most of the query runtimes were significantly improved, and no query took longer using GeoMine. Using the solvent exposure of protein points reduces the runtimes even further.

**Upscaling.** In PELIKAN, databases had to be created by the user. When PELIKAN was developed, the PDB contained about 80 000 structures. Currently, there are more than twice as many. The focus of the application was on searching subsets of the PDB, e.g., the scPDB. With GeoMine, the entire PDB is searchable, and the database will be kept up-to-date in the future. As the PDB is continuously growing, GeoMine was designed to handle this upscaling. Since the number of available pockets and search points in the database increased by a factor of 3.5, retrieving a potential result point (PRP) list from the indexing structure results in substantially increased runtimes. This is not surprising, since the number of indexed bins defined by typical atom arrangements is constant. Therefore, the PELIKAN indexing structure is no longer used in GeoMine. To enable as many users as possible to search the database quickly, searches are performed on a server using up to 30 cores of a 2x Intel Xeon Gold 6248 processor (20 cores/2.5 GHz), 200 GB of main memory, and a Dell 1.6TB NVMe HHHH AIC PM1725b solid-state drive with an XFS file system. The parallelization of the queries is managed by PostgreSQL. This ensures that the hardware is used in the best possible way at each point in time and with a varying numbers of users. In addition, the use of



**Figure 4.** Average query runtimes of test queries. Each bar displays the mean value of five independent experiments. For each geometric query type, there is a “standard” query consisting of oxygen, nitrogen, and carbon atoms, a “metal” query where one of the query points is changed to a magnesium ion, a “metal, water” query with a magnesium ion and a water query point, and a “metal, water, phosphorus” query where a third point is also changed to a phosphorus atom. Calculations were performed on a PC equipped with an Intel i5-9500 (3.0 GHz) processor, 16 GB of main memory (6 GB usable by the PostgreSQL RDBMS), and a Toshiba BG4 PCIe solid-state drive (512 GB, model nvme) with a btrfs file system. The platform runs with a standard configuration of an openSUSE LEAP 15.0 with either PELIKAN (left, light gray) or GeoMine (right, dark gray).

PostgreSQL ensures that the method will keep working in the future even if there is an exponential growth of the number of structures in the PDB. Currently, the database is about 154 GB in size with the biggest table which stores all points of the pockets using about 72 GB of this data, while PostgreSQL reports a maximum table size of 32 TB.<sup>31</sup>

## CONCLUSIONS

Structure-based data mining has a huge potential in modern rational drug design, offering a large variety of data analytics. However, a database-driven method for efficient access to geometrical features in protein-ligand binding sites is required. For this task, we developed GeoMine and implemented it for practical validation. To our knowledge, it is the first method of its kind that enables searches in the entire PDB, including empty binding sites. Even purely ligand-based searches and template-free queries are possible. Our approach allows highly flexible definitions of search points that are not limited to predefined motifs. Point-point constraints and angle constraints allow the definition of geometrically precise and vague parts within a query. Textual and numerical properties of ligands, pockets, proteins, and complexes can be used to define a query in more detail and restrict the runtime.

GeoMine offers a comprehensive web interface. Searches with sufficiently specific queries are answered within seconds to minutes. The results can be displayed as superimpositions in an NGL viewer,<sup>32,33</sup> and statistics of the matched points, distances, interactions and angles can be downloaded. Through the extensive possibilities of query generation and GeoMine’s public availability as part of the *ProteinsPlus* web service, our method is highly useful for numerous applications.

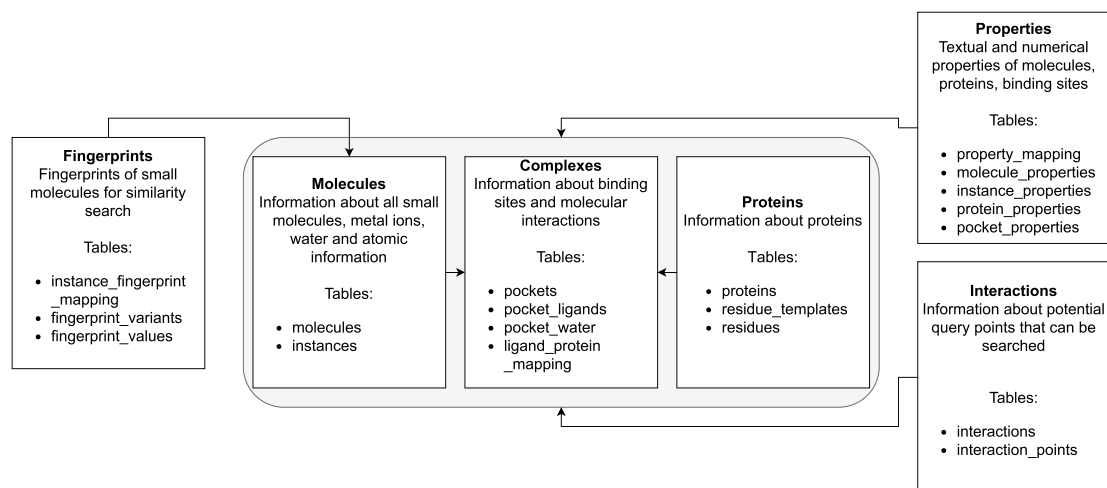
Particularly for protein kinase inhibitor design with its wealth of structural data, GeoMine enables a rational and data-driven molecular design approach. Similarities in binding sites can be used to design novel ligands. Selectivity patterns in protein kinases can be analyzed and investigated based on known binding sites even in the absence of ligands. Environments of functional groups can be analyzed to gain a better understanding

of interaction geometries. Geometric data mining enables the exploitation of specific interactions to selectively address protein kinase binding sites. Such target assessments and selectivity analyses are facilitated through flexible and time-efficient searches with intuitively generated template-based and template-free queries.

In the future, we plan to extend GeoMine by integrating protein-protein interfaces, 2D query design, and an automated query generation to extract commonalities and differences between given protein structure collections to formulate queries. The major benefits of GeoMine include the possibilities to design tailor-made queries, rendering it a versatile tool for multiple challenges in protein kinase inhibitor design, the inclusion of textual and numerical filters, its applicability to ligand-bound and predicted binding sites, and its short almost interactive computing time. These are unique features that enable a new way to deal with large amounts of structural data in drug design. Therefore, we hope that the tool will assist in numerous applications scenarios and will provide a unique means to explore and annotate protein kinases.

## EXPERIMENTAL SECTION

To make GeoMine accessible for many life scientists independent of their previous experience in software usage, the system is based on a database server with a web-based graphical user interface. As described in detail below, the database contains precalculated data on protein-ligand complexes and supports efficient access based on a highly flexible query engine. Briefly speaking, GeoMine is based on the PostgreSQL database management system due to its large SQL and full ACID (atomicity, consistency, isolation, and durability) compliance, good multiuser management, extensibility, and multitude of features such as the support for different security authentications. A back-end software written in C++ on top of the NAOMI library<sup>34,35</sup> preprocesses the PDB. The process is fully automated, including the calculation of binding sites<sup>36</sup> and the handling of protonation states and tautomerism.<sup>37</sup> Using the graphical user interface of the web service, a query is generated in extensible markup language (XML) and sent to the back-end server, which initiates the database search with an iterative approach. For computational chemists interested in automation, a representational



**Figure 5.** Illustration of the database schema. Tables are grouped as either molecule, protein, complex, property, fingerprint, or interaction. Arrows depict dependencies between groups.

state transfer application programming interface (REST API) is available to directly submit a query to the server.

**Data Preprocessing and Knowledge Extraction.** The process of building the GeoMine database consists of four subsequent phases. In the first phase, PDB files are read and converted into objects that represent the complex, its small molecules, water molecules, and metal ions. Among others, the PDB-ID, compound names, source organism, experimental method, and resolution are extracted from the header section. The second phase is dominated by data preparation. First, missing hydrogen atoms in the complex are identified and their coordinates are optimized using Protoss.<sup>37</sup> Protoss analyzes rotatable hydrogen atoms of terminal groups (e.g., hydroxy and amino groups), tautomers and protonation states of all chemical moieties (including ligand molecules), and flips of ambiguous residue side chain orientations (Asn, Gln, and His) and evaluates alternative orientations of water molecules. In addition, alternative conformations that might be annotated in the original protein structure are removed as they could hinder the analysis of molecular interactions. Second, all chains with at least 50% HETATM entries in the PDB file and more than 5 and less than 100 heavy atoms are converted to ligands. Third, the pocket detection algorithm DoGSite is applied.<sup>36</sup> DoGSite is a grid-based approach where each grid point is labeled depending on its spatial overlap with any protein atom. With a difference-of-Gaussian filter, small sphere-like cavities are identified, which are subsequently clustered to potential subpockets. Lastly, adjacent subpockets are merged into pockets. Fourth, all ligands that have at least six heavy atoms are associated with the detected pockets. This is done by finding the pocket that contains at least 20% of all the small molecule atoms. If there is no precalculated pocket for the small molecule, we calculate a new one using all heavy atoms within a 6.5 Å radius of any of the molecule atoms. The chosen radius represents a reasonable trade-off between specificity and runtime. We decided to use all atoms instead of the ligand's geometric center and the radius thereof to more accurately capture the binding site shape. The pockets that are not associated with a ligand are now filtered based on two criteria. First, the pocket volume has to exceed 100 Å<sup>3</sup>. Pocket volumes are calculated using the DoGSite pocket grids. The threshold is motivated by the fact that at least three water molecules should be accommodated by the pocket. Second, the largest *k* pockets are selected, where *k* is limited to two times the number of protein chains in the asymmetric unit.

Subsequently, several pocket characteristics,<sup>36</sup> secondary structure assignments from the PDB annotation or those calculated by an in-house version of DSSP<sup>38</sup> if the helix and sheet assignments are missing in the PDB file, and all protein-ligand interactions are calculated. Interactions include hydrogen bonds based on the predicted protonation or tautomerism patterns, aromatic interactions between rings, ionic interactions, metal interactions, and  $\pi$ - $\pi$  interactions. The

interactions are calculated according to Inhester et al.<sup>7</sup> A hydrogen bond is assigned if the distance between the corresponding donor and acceptor atoms is between 2 and 3.8 Å, the hydrogen-donor-acceptor angle is between  $-45^\circ$  and  $45^\circ$ , the donor-acceptor-lone pair angle is between  $-70^\circ$  and  $70^\circ$ , and the distance between the hydrogen atom and the lone pair is between 0 and 2 Å.  $\pi$ - $\pi$  interactions are assigned if the centroids of the interacting ring systems are between 2.5 and 5 Å apart. For  $\pi$ -cation interactions, the distance between the ring center and the cation has to be between 2 and 4 Å. Metal interactions are identified if the metal ion is between 1 and 3 Å away from any coordinating atom. Ionic hydrogen bonds are identified if the distance between the interaction partners corresponds to the sum of their corresponding van der Waals radii  $\pm 1$  Å. Further, the solvent-accessible surface area<sup>39,40</sup> of each pocket atom is calculated as described for the scoring function HYDE<sup>41–43</sup> so that buried atoms can be differentiated from solvent-exposed atoms in queries.

**Database Content and Design.** In GeoMine, there are multiple tables stored that can be divided into four groups (see Figure 5). This schema was already used in GeoMine's predecessor PELIKAN and was only slightly altered because some features that only exist in SQLite and not in PostgreSQL were used, e.g., the possibility of storing multiple data types in a single column. The first group stores general information about all small molecules, metal ions, water, proteins, and binding sites. Additionally, the molecular interactions and atomic information are stored there. With this information, the protein-ligand complex can be searched and also reconstructed when visualizing results. The second group stores fingerprints of all small molecules to enable similarity searching. In the third group, the textual and numerical properties of the ligands, proteins, and pockets such as the ligand names and resolution or the pocket volume are stored. Lastly, the fourth group contains all information and data about the potential query points, i.e., heavy atoms, ring centers, secondary structure points, and interactions. Each query point is stored with a unique ID and, among others, its coordinates, its chemical element, a foreign key, i.e., a key that links to the complexes table with the pocket properties, its solvent accessibility, its originating molecule (protein or ligand), and vectors in case of ring normals and secondary structure element end- or midpoints. Interactions are defined by the interaction type and the two query point IDs of the interacting atoms. Using the B-tree database indexes available in PostgreSQL enables fast access to this data and geometric queries.

**GeoMine Query Types.** GeoMine enables the search for properties of interest and geometric patterns in different ways. All query types can be combined.

**Textual and Numerical Searches.** The most basic search of GeoMine uses textual and numerical properties. This is helpful for a preselection or restriction of the search or the selection of a query template structure. A variety of filters for ligands, proteins, pockets, and

protein-ligand complexes are available. For ligands, an element count, i.e., the minimum and maximum number of specific chemical elements, a functional group count, and molecular properties such as the molecular weight or the number of hydrogen bond donors and acceptors can be defined. Proteins can be filtered by their UniProt ID,<sup>44</sup> EC number, or source organisms. A minimum and maximum count of specific amino acids the search space to relevant protein can be set to filter the pockets. Additionally, ligand names and pocket properties such as hydrophobicity, volume, or depth can be specified. Finally, the number of complexes can be reduced by title entries, resolution, organism, EC number, and experimental source. Handling and querying these data is straightforward in a relational database.

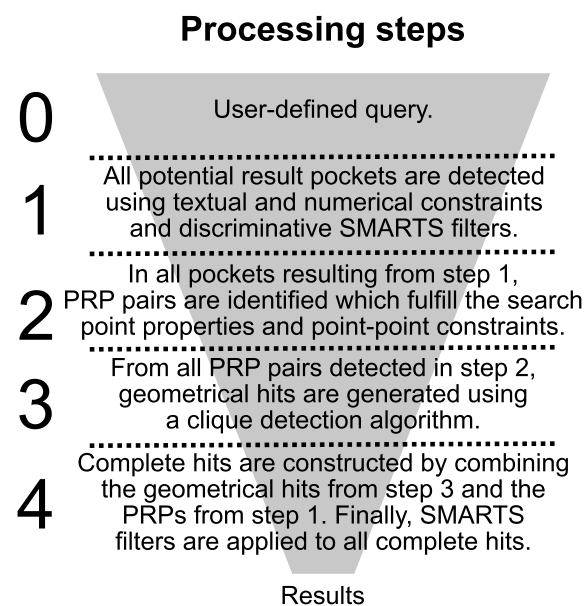
**Chemical Substructure Searches.** Substructure searches can be carried out using simplified molecular input line entry system (SMILES) arbitrary target specification (SMARTS) strings.<sup>45</sup> Large SMARTS patterns (patterns describing at least five atoms) and those containing rare elements are highly discriminative and can therefore be used to reduce the number of pockets in the search space early on. SMARTS patterns that do not fulfill these criteria will be evaluated at the end of the search algorithm because they will likely result in numerous matches, causing an increased runtime.

**Ligand Similarity Searches.** All ligands of the PDB structures are stored with their extended connectivity fingerprint (ECFP)-like<sup>46</sup> Morgan<sup>47</sup> fingerprints and CSFP (connected subgraph fingerprint) and tCSFP (topological connected subgraph fingerprint)<sup>48</sup> fingerprints in the database. The latter are new fingerprints that, in contrast to other methods, capture all connected subgraphs as structural features of a molecule. This property gives these fingerprints a complete feature space and a high adaptive potential. Especially in standard similarity-driven virtual screening settings, the CSFP has substantial advantages.<sup>48</sup> All fingerprints enable the user to define a certain similarity level to a given query ligand specified in SMILES,<sup>49</sup> which further helps to reduce the search space to relevant protein-ligand complexes. Furthermore, they can be used standalone to efficiently search small molecules in GeoMine.

**Geometry-Based Searches.** The unique key feature of GeoMine is its capability for precise geometric searching in binding sites. All atoms, helix and strand mid- and endpoints, and ring centers are possible query points in GeoMine. These points can either be selected using a template structure (loaded PDB file) or specified template-free. Queries based on template structures can be enriched by artificial search points, enabling the generation of hypothetical queries. Furthermore, there are many properties associated with query points that can either be automatically determined or manually adjusted. These properties are the chemical element, the interaction type, the molecule type (protein, ligand, metal, or water), the secondary structure class (helix, sheet, or none), whether they are in the protein backbone or a side chain, and the amino acid or amino acid type, e.g., polar. In the case of the secondary structure annotation, the user can also define a C $\alpha$  atom as an end- or midpoint of a helix or strand. Also, functional groups of ligands and the environment can be specified using SMARTS strings. GeoMine allows connecting atoms in SMARTS expressions with query points such that substructures with certain geometric orientations can be searched. Note that all property specifications are optional. All points that are at most 15 Å apart from each other can be connected by distance and interaction constraints. This value was chosen so that residue atoms on opposite sides of the pocket can be selected while the runtime of the search algorithm remains feasible. The distance constraints enable defined minimum and maximum ranges, while the interaction constraints are used to indicate specific interactions such as hydrogen bonds and ionic, metal,  $\pi$ -cation, and  $\pi$ - $\pi$  interactions. In addition, angles can be defined between two distances, interactions, or ring normals with minimum and maximum values to allow different constraint levels.

**Overview of the Search Process.** From the perspective of a user, it is important to get a rough idea of the search process so that the impact of query types and parameters on runtime and results can be estimated. An iterative search algorithm was developed to reduce the potential result set as early as possible while executing the fast search steps before the time-consuming ones. This algorithm consists of four

distinct steps, which are displayed in Figure 6. Note that the algorithm is exact, i.e., complexes and binding sites are retrieved if and only if they



**Figure 6.** Overview of the search process in GeoMine.

fulfill the specified user query. The algorithm differs from the one used in GeoMine's predecessor PELIKAN<sup>7</sup> in several aspects. In the latter, a tailor-made indexing structure was used to identify potential hits (named potential result points, PRPs) quickly. In GeoMine, accessing and checking matching properties of search points in the database are considerably improved with respect to runtime. Details about these improvements are given in the [Query Computing Time](#) section.

**Step 1.** All query features regarding textual and numerical properties, ligand similarity searches, and specific SMARTS-based substructure searches, which include more than four atoms or contain at least one non-carbon, non-nitrogen or non-oxygen atom, are performed. This step results in a list of potentially matching binding sites and is passed on to the next step.

**Step 2.** Point-point or distance and interaction constraints are processed sequentially. To reduce the runtime of this step, the processing order of these constraints is optimized beforehand. Here, the order is ascending with respect to the number of expected results, which is estimated by the product of the database count of different elements and interaction types for each search point. In each point-point constraint processing step, all PRPs that match all properties of the search points and are in the list of potential pockets are detected in the database. The interaction type is used if the point-point constraint is an interaction constraint. Set distance ranges are evaluated by calculating the Euclidean distance between the two PRPs. If no matching PRPs are found in a specific pocket for a point-point constraint, this pocket is removed from the list of potential pockets. The list of all potential PRPs is updated in each processing step. By doing this, the size of these lists of possible pockets and PRPs steadily decreases.

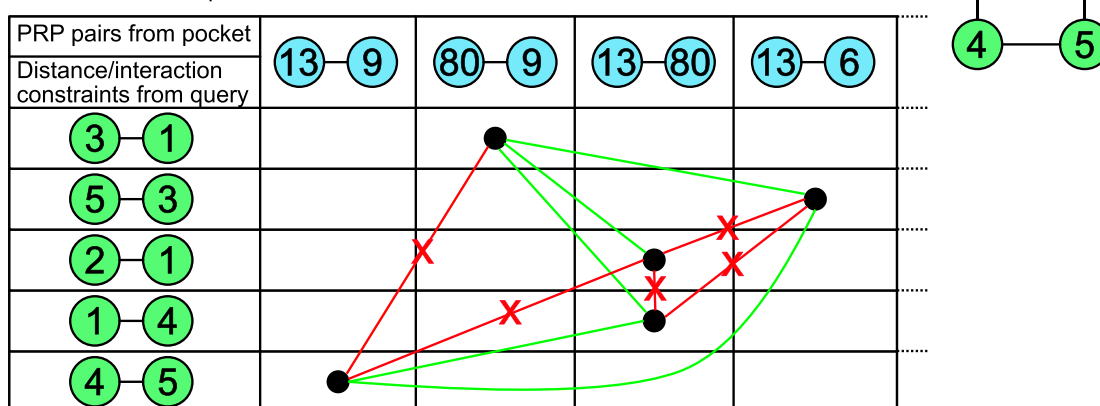
This step results in a list containing all possible PRPs that match all properties and point-point constraints for each search point. At this step of the algorithm, the possible PRPs are not yet assigned to pockets.

**Step 3.** The list of all detected PRPs from the previous step is used to construct a product graph for each pocket<sup>50</sup> (see Figure 7). Herein, each combination of a point-point constraint with a matching PRP pair generates a node. These nodes are connected if the PRP assignments do not contradict each other and the angle constraints are fulfilled. To find all matches for the whole query, the maximal cliques are calculated on the product graph.



For each pocket:

1. Build product graph using PRP pairs and checking the compatibility with distance/interaction constraints from the query
2. Calculate all cliques of size  $n$ , with  $n$  = number of distance/interaction constraints



**Figure 7.** Step 3 of GeoMine's search process. For each pocket, a product graph is built using all compatible results (black dots) of pocket PRP pairs (blue circles) to the distance and interaction constraints (green circles) of the user-defined query. Numbers in green and the blue circles represent PRP and search point IDs, respectively. Cliques are calculated for all compatible results. Green lines indicate that two results fulfill the angle constraints, and red lines that they do not. Adapted in part with permission from Inhester et al.<sup>7</sup> Copyright 2017, American Chemical Society.

**Step 4.** SMARTS patterns that were not processed in the first step are evaluated on the resulting PRPs. This includes, for example, SMARTS patterns that describe parts of an atom environment or chemical relations between search points. All these patterns are checked for each of the matching atoms of step 3.

**Database Statistics.** As mentioned before, GeoMine stores information about small molecules in the database. Conceptually, the topology and 3D coordinates of each inserted small molecule are separated. The two properties are stored in different tables so that the different topologies define unique ligands while the 3D coordinates define instances in exactly one binding site. Based on the RCSB PDB (<https://www.rcsb.org/>),<sup>51</sup> which contains 166 296 entries, there are 44 786 unique ligand topologies in the database. This number is a result of the ligand definition and the adjustment of HETATM annotations in the PDB files mentioned above. Water is the most abundant small molecule. The most frequently occurring small molecules are inorganic ions such as sulfate, zinc, and magnesium ions. The most commonly occurring organic small molecules comprise, for example, heme, N-acetylglucosamine, and glycerol. Of the 914 408 available pockets, 412 924 contain a ligand. This means that about 55% of all available pockets contain no reference ligand and should be considered as hypothetical binding sites. On average, the pockets have a volume of 270 Å<sup>3</sup> and comprise 200 heavy atoms with a solvent-accessible surface of about 325 Å<sup>2</sup>. There are about 17 solvent-accessible hydrogen bond acceptors and 12 hydrogen bond donors on average, and the ratio of hydrophobic pocket residues to all pocket residues is 0.376.

## SOFTWARE AND DATA AVAILABILITY

All data used were generated from the Protein Data Bank, which is freely available at PDB, <https://www.rcsb.org/>. GeoMine is available as a free web service that can be accessed using the link <https://proteins.plus>. All queries developed throughout this study are available in the [Supporting Information](#).

## ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jmedchem.1c01046>.

List of all properties of search points and their value ranges, list of all textual and numerical filters that can be used in GeoMine and their value ranges, exact query definitions of the application examples, and results

for the application example queries downloaded with the GeoMine GUI ([PDF](#))

List of all PDB-IDs in the GeoMine database as of May 28, 2021 and all query files in a machine-readable JavaScript object notation (JSON) format that can be imported in GeoMine to reproduce the queries and their results downloaded from GeoMine ([ZIP](#))

## AUTHOR INFORMATION

### Corresponding Author

Matthias Rarey – ZBH Center for Bioinformatics, Universität Hamburg, 20146 Hamburg, Germany; [orcid.org/0000-0002-9553-6531](https://orcid.org/0000-0002-9553-6531); Email: [matthias.rarey@uni-hamburg.de](mailto:matthias.rarey@uni-hamburg.de)

### Authors

Joel Graef – ZBH Center for Bioinformatics, Universität Hamburg, 20146 Hamburg, Germany; [orcid.org/0000-0001-8327-4936](https://orcid.org/0000-0001-8327-4936)

Christiane Ehrt – ZBH Center for Bioinformatics, Universität Hamburg, 20146 Hamburg, Germany; [orcid.org/0000-0003-1428-0042](https://orcid.org/0000-0003-1428-0042)

Konrad Diedrich – ZBH Center for Bioinformatics, Universität Hamburg, 20146 Hamburg, Germany; [orcid.org/0000-0001-8171-0888](https://orcid.org/0000-0001-8171-0888)

Martin Poppinga – ZBH Center for Bioinformatics, Universität Hamburg, 20146 Hamburg, Germany

Norbert Ritter – ZBH Center for Bioinformatics, Universität Hamburg, 20146 Hamburg, Germany

Complete contact information is available at:

<https://pubs.acs.org/10.1021/acs.jmedchem.1c01046>

### Author Contributions

J.G. implemented the C++ PostgreSQL interface of GeoMine, developed faster SQL queries and core functionalities, and integrated the ligand similarity search, empty binding sites, secondary structure points, and surface properties to the search capabilities. K.D. implemented the graphical user interface and the functions that provide the results to the GUI. C.E. designed and performed all exemplary protein kinase use cases and

analyzed the results. M.P. and N.R. participated in database design and query optimization. M.R. participated in the development of concepts and supervised the project. J.G., C.E., and M.R. wrote the manuscript.

## Notes

The authors declare the following competing financial interest(s): ProteinsPlus and the NAOMI ChemBioSuite use some methods that are jointly owned and/or licensed to BioSolveIT GmbH, Germany, M.R. is a shareholder of BioSolveIT GmbH.

## ACKNOWLEDGMENTS

The authors thank the whole development team of the NAOMI library for forming the basis of this work as well as the developers of PostgreSQL for making their database system available. This work was supported by the German Federal Ministry of Education and Research as part of CompLS and de.NBI (031L0172 and 031L0105). C.E. is funded by Data Science in Hamburg, Helmholtz Graduate School for the Structure of Matter (Grant HIDS-0002).

## ABBREVIATIONS USED

ACID, atomicity, consistency, isolation, and durability;; BS, binding site; CSFP, connected subgraph fingerprint; ECFP, extended connectivity fingerprint; GUI, graphical user interface; JSON, JavaScript object notation; KLIFS, Kinase-Ligand Interaction Fingerprints and Structures; PRP, potential result point; RCSB, research collaboration for structural bioinformatics; REST API, representational state transfer application programming interface; SMARTS, SMILES arbitrary target specification; SMILES, simplified molecular input line entry system; SQL, structure query language; SSE, secondary structure element; tCSFP, topological connected subgraph fingerprint; XML, extensible markup language

## REFERENCES

- (1) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.
- (2) Korb, O.; Kuhn, B.; Hert, J.; Taylor, N.; Cole, J.; Groom, C.; Stahl, M. Interactive and Versatile Navigation of Structural Databases. *J. Med. Chem.* **2016**, *59*, 4257–4266.
- (3) Mobilio, D.; Walker, G.; Brooijmans, N.; Nilakantan, R.; Denny, R. A.; Dejoannis, J.; Feyfant, E.; Kowtciwar, R. K.; Mankala, J.; Palli, S.; Punyamantula, S.; Tatipally, M.; John, R. K.; Humblet, C. A Protein Relational Database and Protein Family Knowledge Bases to Facilitate Structure-Based Design Analyses. *Chem. Biol. Drug Des.* **2010**, *76*, 142–153.
- (4) Weisel, M.; Bitter, H.-M.; Diederich, F.; So, W. V.; Kondru, R. PROLIX: Rapid Mining of Protein–Ligand Interactions in Large Crystal Structure Databases. *J. Chem. Inf. Model.* **2012**, *52*, 1450–1461.
- (5) Hendlich, M.; Bergner, A.; Günther, J.; Klebe, G. Relibase: Design and Development of a Database for Comprehensive Analysis of Protein–Ligand Interactions. *J. Mol. Biol.* **2003**, *326*, 607–620.
- (6) Golovin, A.; Henrick, K. MSDmotif: Exploring Protein Sites and Motifs. *BMC Bioinf.* **2008**, *9*, 312.
- (7) Inhester, T.; Bietz, S.; Hilbig, M.; Schmidt, R.; Rarey, M. Index-Based Searching of Interaction Patterns in Large Collections of Protein–Ligand Interfaces. *J. Chem. Inf. Model.* **2017**, *57*, 148–158.
- (8) Angles, R.; Arenas-Salinas, M.; García, R.; Reyes-Suarez, J. A.; Pohl, E. GSP4PDB: A Web Tool to Visualize, Search and Explore Protein-Ligand Structural Patterns. *BMC Bioinf.* **2020**, *21*, 85.
- (9) Bittrich, S.; Burley, S. K.; Rose, A. S. Real-Time Structural Motif Searching in Proteins Using an Inverted Index Strategy. *PLoS Comput. Biol.* **2020**, *16*, No. e1008502.

- (10) Kuhn, B.; Gilberg, E.; Taylor, R.; Cole, J.; Korb, O. How Significant Are Unusual Protein–Ligand Interactions? Insights from Database Mining. *J. Med. Chem.* **2019**, *62*, 10441–10455.
- (11) Yoshikawa, K.; Yoshino, T.; Yokomizo, Y.; Uoto, K.; Naito, H.; Kawakami, K.; Mochizuki, A.; Nagata, T.; Suzuki, M.; Kanno, H.; Takemura, M.; Ohta, T. Design, Synthesis and SAR of Novel Ethylenediamine and Phenylenediamine Derivatives as Factor Xa Inhibitors. *Bioorg. Med. Chem. Lett.* **2011**, *21*, 2133–2140.
- (12) Kanev, G. K.; de Graaf, C.; Westerman, B. A.; de Esch, I. J. P.; Kooistra, A. J. KLIFS: An Overhaul After the First 5 Years of Supporting Kinase Research. *Nucleic Acids Res.* **2021**, *49*, D562–D569.
- (13) KLIFS. <https://klifs.net/> (accessed 2021-03-24).
- (14) Baumli, S.; Endicott, J. A.; Johnson, L. N. Halogen Bonds Form the Basis for Selective P-TEFb Inhibition by DRB. *Chem. Biol.* **2010**, *17*, 931–936.
- (15) Rothweiler, U.; Stensen, W.; Brandsdal, B. O.; Isaksson, J.; Leeson, F. A.; Engh, R. A.; Svendsen, J. S. M. Probing the ATP-Binding Pocket of Protein Kinase DYRK1A with Benzothiazole Fragment Molecules. *J. Med. Chem.* **2016**, *59*, 9814–9824.
- (16) Do, P. A.; Lee, C. H. The Role of CDKs in Tumours and Tumour Microenvironments. *Cancers* **2021**, *13*, 101.
- (17) Ahn, J. S.; Radhakrishnan, M. L.; Mapelli, M.; Choi, S.; Tidor, B.; Cuny, G. D.; Musacchio, A.; Yeh, L.-A.; Kosik, K. S. Defining Cdk5 Ligand Chemical Space with Small Molecule Inhibitors of Tau Phosphorylation. *Chem. Biol.* **2005**, *12*, 811–823.
- (18) Eid, S.; Turk, S.; Volkamer, A.; Rippmann, F.; Fulle, S. KinMap: A Web-Based Tool for Interactive Navigation Through Human Kinome Data. *BMC Bioinf.* **2017**, *18*, 16.
- (19) Wood, E. R.; Truesdale, A. T.; McDonald, O. B.; Yuan, D.; Hassell, A.; Dickerson, S. H.; Ellis, B.; Pennisi, C.; Horne, E.; Lackey, K.; Alligood, K. J.; Rusnak, D. W.; Gilmer, T. M.; Shewchuk, L. A Unique Structure for Epidermal Growth Factor Receptor Bound to GW572016 (Lapatinib). *Cancer Res.* **2004**, *64*, 6652–6659.
- (20) Kitagawa, D.; Yokota, K.; Gouda, M.; Narumi, Y.; Ohmoto, H.; Nishiwaki, E.; Akita, K.; Kirii, Y. Activity-Based Kinase Profiling of Approved Tyrosine Kinase Inhibitors. *Genes to Cells* **2013**, *18*, 110–122.
- (21) Kortemme, T.; Creighton, T. E. Ionisation of Cysteine Residues at the Termini of Model  $\alpha$ -Helical Peptides. Relevance to Unusual Thiol pKa Values in Proteins of the Thioredoxin Family. *J. Mol. Biol.* **1995**, *253*, 799–812.
- (22) Liu, Q.; Sabnis, Y.; Zhao, Z.; Zhang, T.; Buhrlage, S.; Jones, L.; Gray, N. Developing Irreversible Inhibitors of the Protein Kinase Cysteinome. *Chem. Biol.* **2013**, *20*, 146–159.
- (23) do Amaral, D. N.; Lategahn, J.; Fokoue, H. H.; da Silva, E. M. B.; Sant’Anna, C. M. R.; Rauh, D.; Barreiro, E. J.; Laufer, S.; Lima, L. M. A Novel Scaffold for EGFR Inhibition: Introducing N-(3-(3-Phenylureido)Quinoxalin-6-yl) Acrylamide Derivatives. *Sci. Rep.* **2019**, *9*, 14.
- (24) Aertgeerts, K.; Skene, R.; Yano, J.; Sang, B.-C.; Zou, H.; Snell, G.; Jennings, A.; Iwamoto, K.; Habuka, N.; Hirokawa, A.; Ishikawa, T.; Tanaka, T.; Miki, H.; Ohta, Y.; Sogabe, S. Structural Analysis of the Mechanism of Inhibition and Allosteric Activation of the Kinase Domain of HER2 Protein. *J. Biol. Chem.* **2011**, *286*, 18756–18765.
- (25) Petri, L.; Egyed, A.; Bajusz, D.; Imre, T.; Hetényi, A.; Martinek, T.; Ábrányi Balogh, P.; Keseru, G. An Electrophilic Warhead Library for Mapping the Reactivity and Accessibility of Tractable Cysteines in Protein Kinases. *Eur. J. Med. Chem.* **2020**, *207*, 112836.
- (26) Heppner, D. E.; Günther, M.; Wittlinger, F.; Laufer, S. A.; Eck, M. J. Structural Basis for EGFR Mutant Inhibition by Trisubstituted Imidazole Inhibitors. *J. Med. Chem.* **2020**, *63*, 4293–4305.
- (27) Goedken, E. R.; Argiriadi, M. A.; Banach, D. L.; Fiamengo, B. A.; Foley, S. E.; Frank, K. E.; George, J. S.; Harris, C. M.; Hobson, A. D.; Ihle, D. C.; Marcotte, D.; Merta, P. J.; Michalak, M. E.; Murdock, S. E.; Tomlinson, M. J.; Voss, J. W. Tricyclic Covalent Inhibitors Selectively Target Jak3 through an Active Site Thiol. *J. Biol. Chem.* **2015**, *290*, 4573–4589.

- (28) Desaphy, J.; Bret, G.; Rognan, D.; Kellenberger, E. sc-PDB: A 3D-database of Ligandable Binding Sites—10 years on. *Nucleic Acids Res.* **2015**, *43*, D399–D404.
- (29) sc-PDB. <http://bioinfo-pharma.u-strasbg.fr/scPDB/> (accessed 2020-07-30).
- (30) Burk, D. L.; Hon, W. C.; Leung, A. K.-W.; Berghuis, A. M. Structural Analyses of Nucleotide Binding to an Aminoglycoside Phosphotransferase. *Biochemistry* **2001**, *40*, 8756–8764.
- (31) The PostgreSQL Global Development Group. Appendix K. PostgreSQL Limits. *PostgreSQL*. <https://www.postgresql.org/docs/12/limits.html> (accessed on 2021-03-19).
- (32) Rose, A. S.; Hildebrand, P. W. NGL Viewer: A Web Application for Molecular Visualization. *Nucleic Acids Res.* **2015**, *43*, W576–W579.
- (33) Rose, A. S.; Bradley, A. R.; Valasatava, Y.; Duarte, J. M.; Prlić, A.; Rose, P. W. NGL viewer: Web-Based Molecular Graphics for Large Complexes. *Bioinformatics* **2018**, *34*, 3755–3758.
- (34) Urbaczek, S.; Kolodzik, A.; Fischer, J. R.; Lippert, T.; Heuser, S.; Groth, I.; Schulz-Gasch, T.; Rarey, M. NAOMI: On the Almost Trivial Task of Reading Molecules from Different File formats. *J. Chem. Inf. Model.* **2011**, *51*, 3199–3207.
- (35) Bietz, S.; Inhester, T.; Lauck, F.; Sommer, K.; von Behren, M. M.; Fährrolfes, R.; Flachsenberg, F.; Meyder, A.; Nittinger, E.; Otto, T.; Hilbig, M.; Schomburg, K. T.; Volkamer, A.; Rarey, M. From Cheminformatics to Structure-Based Design: Web Services and Desktop Applications Based on the NAOMI Library. *J. Biotechnol.* **2017**, *261*, 207–214.
- (36) Volkamer, A.; Kuhn, D.; Grombacher, T.; Rippmann, F.; Rarey, M. Combining Global and Local Measures for Structure-Based Druggability Predictions. *J. Chem. Inf. Model.* **2012**, *52*, 360–372.
- (37) Bietz, S.; Urbaczek, S.; Schulz, B.; Rarey, M. Protoss: A Holistic Approach to Predict Tautomers and Protonation States in Protein-Ligand Complexes. *J. Cheminf.* **2014**, *6*, 12.
- (38) Kabsch, W.; Sander, C. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. *Biopolymers* **1983**, *22*, 2577–2637.
- (39) Lee, B.; Richards, F. The Interpretation of Protein Structures: Estimation of Static Accessibility. *J. Mol. Biol.* **1971**, *55*, 379–400.
- (40) Richards, F. M. Areas, Volumes, Packing, and Protein Structure. *Annu. Rev. Biophys. Bioeng.* **1977**, *6*, 151–176.
- (41) Reulecke, I.; Lange, G.; Albrecht, J.; Klein, R.; Rarey, M. Towards an Integrated Description of Hydrogen Bonding and Dehydration: Decreasing False Positives in Virtual Screening with the HYDE Scoring Function. *ChemMedChem* **2008**, *3*, 885–897.
- (42) Schneider, N.; Hindle, S.; Lange, G.; Klein, R.; Albrecht, J.; Briem, H.; Beyer, K.; Claußen, H.; Gastreich, M.; Lemmen, C.; Rarey, M. Substantial Improvements in Large-Scale Redocking and Screening Using the Novel HYDE Scoring Function. *J. Comput.-Aided Mol. Des.* **2012**, *26*, 701–723.
- (43) Schneider, N.; Lange, G.; Hindle, S.; Klein, R.; Rarey, M. A Consistent Description of HYdrogen Bond and DEhydration Energies in Protein-Ligand Complexes: Methods Behind the HYDE Scoring Function. *J. Comput.-Aided Mol. Des.* **2013**, *27*, 15–29.
- (44) The UniProt Consortium. UniProt: A Worldwide Hub of Protein Knowledge. *Nucleic Acids Res.* **2019**, *47*, D506–D515.
- (45) Daylight Chemical Information Systems, Inc. 4. SMARTS - A Language for Describing Molecular Patterns. *Daylight Theory Manual*. <http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html> (accessed on 2020-04-21).
- (46) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754.
- (47) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures—A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107–113.
- (48) Bellmann, L.; Penner, P.; Rarey, M. Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration. *J. Chem. Inf. Model.* **2019**, *59*, 4625–4635.
- (49) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model.* **1988**, *28*, 31–36.
- (50) Bron, C.; Kerbosch, J. Algorithm 457: Finding All Cliques of an Undirected Graph. *Commun. ACM* **1973**, *16*, 575–577.
- (51) RCSB PDB. <https://www.rcsb.org/> (accessed 2020-12-16).

# Supporting Information

## Searching Geometric Patterns in Protein Binding Sites and its Application to Data Mining in Protein Kinase Structures

Author Names: Joel Graef, Christiane Ehrt, Konrad Diedrich, Martin Poppinga, Norbert Ritter, Matthias Rarey\*

Author Address: Universität Hamburg, Center for Bioinformatics, Bundesstrasse 43, 20146 Hamburg, Germany

E-mail: matthias.rarey@uni-hamburg.de

### Table of Contents

Table S1.....	S1
Table S2.....	S2
Paragraph S1 .....	S3
Table S3.....	S3
Table S4.....	S4
Table S5.....	S5
Paragraph S2 .....	S7

**Table S1.** Properties which can be assigned to a search point in the geometrical query and their possible values.

Property	Possible choices
Original Molecule	Ligand, Metal, Protein, Water
Element	Alpha Carbon, Boron, Bromine, Calcium, Carbon, Chlorine, Cobalt, Copper, Fluorine, Iodine, Iron, Magnesium, Manganese, Nickel, Nitrogen, Oxygen, Phosphorus, Sulfur, Zinc

S1

Interaction type		Acceptor, Anion, AromaticRingCenter, Cation, Donor, Hydrophobic, Metal
If Original Molecule = Protein	Amino acid	Ala, Arg, Asn, Asp, Cso, Cys, Glu, Gln, Gly, His, Ile, Leu, Lys, Met, Phe, Pro, Ser, Thr, Trp, Tyr, Val, Hydrophobic, Polar, Aromatic, Acidic, Basic, Neutral
	Location in amino acid	Backbone, Sidechain
	Secondary structure	Helix, Sheet, Helix End, Helix Mid, Sheet End, Sheet Mid, no secondary structure
If Original Molecule = Ligand	Functional group	Alcohol, Aldehyde, Amide, Amidine, Amine, Azide, Ester, Ether, Furane, Guanidine, Ketone, Nitrile, Phenyl, Pyridine, Pyrrole, Thiophene
If Original Molecule = Ligand or Protein	Atom description	SMARTS
	Minimal surface	All floating-point numbers $\geq 0 \text{ \AA}^2$

**Table S2.** Textual and numerical properties which can be added to a query and their possible values.

Category	Property	Possible choices
Ligand filter	Element	Boron, Bromine, Carbon, Chlorine, Fluorine, Iodine, Nitrogen, Oxygen, Phosphorus, Sulfur, and a count (min, max)
	Functional group	Alcohol, Aldehyde, Amide, Amidine, Amine, Azide, Ester, Ether, Furane, Guanidine, Ketone, Nitrile, Phenyl, Pyridine, Pyrrole, Thiophene, and a count (min, max)
	Molecule property	Acceptors, aromatic atoms, aromatic rings, aromatic ringsystems, charge, cyclomatic number, donors, halogens, heavy atoms, hetero atoms, inorganic atoms, Lipinski acceptors, logP, molecular weight (MW), Max continuous path of rotatable bonds, max cyclomatic number, max ring size, max ringsystem size, rings, ringsystems, rotatable bonds, stereo bonds (E/Z), stereo centers (R/S), topological polar surface area, unique ring families (URFs), volume, and a count (min, max)

	Similarity	CSFP, tCSFP, ECFPlike, a similarity percentage, the pocket name to which other ligands should be similar to and a min and max for CSFP and tCSFP or radius for ECFPlike Morgan fingerprint
	SMARTS	SMARTS string
Protein filter	Uniprot ID	Free text
	Organism	Free text
	EC number	All four numbers can be set individually
Pocket filter	Ligand name	Free text
	Amino acids	Ala, Arg, Asn, Asp, Cso, Cys, Glu, Gln, Gly, His, Ile, Leu, Lys, Met, Phe, Pro, Ser, Thr, Trp, Tyr, Val, and a count (min, max)
	Property	Acceptors, depth, donors, heavy atoms, hydrophobicity, metal, DoGSite simple score, surface, surface-volume-ratio, volume, and a value range (min, max)
	Has ligand	No parameter. If filter is set only pockets containing a ligand are searched.
P-L-Complex-filter	PDB title entry	Free text
	Resolution	Value range (min, max)
	Experimental source	Unknown, electron crystallography, electron microscopy, fiber diffraction, NMR solid state, NMR solution, neutron diffraction, solution scattering, X-ray

### Paragraph S1. Additional filters

In addition to the filters above there exists a PDB subselection option. The user can provide a list of PDB IDs to limit the search to specific PDBs.

**Table S3.** Overview of existing tools for 3D searching in structural data related to data processing and storage.

	Dataprocessing and storage
Relibase/Relibase+	- C++-based self-written database system with objects stored in multiple archives in a B-tree data structure

	- binding site definition: 7 Å of the ligand atoms
PDBeMotif/MSDmotif	- PostgreSQL - binding site definition: 16 Å of the ligand atoms
PRDB	- information about database type not provided in the publication - binding site definition: 8 Å of the ligand atoms
PROLIX	- information about database type not provided in the publication - binding site definition: 4.5 Å of the ligand atoms
CSD-CrossMiner	- SQLite - HET groups with less than 5 or more than 100 atoms are removed - binding site definition: 6 Å of the ligand atoms
PELIKAN	- SQLite - binding site definition: 6.5 Å of the ligand atoms
GSP4PDB	- PostgreSQL - HET entries as ligands - binding site definition: 7.0 Å of the ligand atoms
strucmotif-search	- inverted index implemented by a file system-based approach - no database - motifs of at least 3 and at most 10 amino acids - distances between amino acid pairs of 15 Å at maximum - all ligands are removed

**Table S4.** Overview of existing tools for 3D searching in structural data related to search capabilities and algorithms, and result presentation.

	Search capabilities and algorithms	Result presentation
Relibase/Relibase+	- incremental - start with fingerprints	- superimposition based on sequence - list of results and statistics
PDBeMotif/MSDmotif	- no information in publication other than the search being based on C $\alpha$ coordinates or end of sidechain coordinates	- superimpose similar proteins (only available for sequence-based searches) - list of results and statistics
PRDB	- conversion from search query to SQL query or directly as database SQL	- list of results

S4

	query without using the interface	
PROLIX	- incremental - start with ligand fingerprints	- list of results and statistics
CSD-CrossMiner	- incremental search which starts with fingerprints	- superimpositions based on the geometric query - list of results
PELIKAN	- incremental - start with points from index structure using environment as triangles - exact (including symmetric matches)	- superimpose results based on the geometric query - list of results and statistics
GSP4PDB	- graph-based structural pattern query is transformed into an SQL query	- 2D and 3D alignment - list of results
strucmotif-search	- motifs are split into residue pairs, similar occurrences are retrieved for geometric descriptors via an inverted index lookup, checking candidates for resemblance to query motif	- Alignment of motifs to query as well as complete structure that contains query or motif according to publication <sup>1</sup> - list of results with a score based on geometric properties of residues in the query and the matched structure pair

**Table S5.** Queries and results of application examples in paper

Files for all queries in the paper are given in the supporting information. JSON files are queries which were exported in GeoMine and can be used with with the Import function (Import button next to the search button in GeoMine on ProteinsPlus. Results of the queries downloaded from GeoMine are in the correspondingly named ZIP files. Below are the names and a short description of the queries.

<b>Filename</b>	<b>Description</b>
Exploiting_Unusual_Interactions_pdb_search	<u>Application example:</u> Exploiting Unusual Interactions for Selective Inhibitor Design <u>Query:</u> Search for appropriate geometric arrangements of halogen atoms attached to aromatic rings with aromatic ring systems in protein sidechains in all ligand-occupied pockets as stored in the PDB.
Exploiting_Unusual_Interactions_KLIFS	<u>Application example:</u> Exploiting Unusual Interactions for Selective Inhibitor Design

<sup>1</sup> Bittrich, S.; Burley, S. K.; Rose, A. S. Real-time structural motif searching in proteins using an inverted index strategy. bioRxiv 2020



	<p><u>Query</u>: Investigation of interaction geometries between aromatic sidechains and chlorine atoms attached to aromatic (hetero-)cycles in the published structures assembled in the third version of the Kinase–Ligand Interaction Fingerprints and Structures (KLIFS) database (<a href="https://klifs.net/">https://klifs.net/</a>, accessed on 03/24/2021).</p>
Exploiting_Unusual_Interactions Adapated_query_KLIFS	<p><u>Application example</u>: Exploiting Unusual Interactions for Selective Inhibitor Design <u>Query</u>: Adaption of the KLIFS query in all ligand-bound protein kinase structures highlighting that there are numerous kinase inhibitors with known binding modes that could be used as potential starting points to improve compound selectivity.</p>
Exploiting_Unusual_Interactions _CDK5	<p><u>Application example</u>: Exploiting Unusual Interactions for Selective Inhibitor Design <u>Query</u>: There are numerous kinase inhibitors with known binding modes that could be used as potential starting points to improve compound selectivity issues. Cyclin-dependent-like kinase 5 (CDK5) is used as an example for a pharmacologically relevant protein kinase target. This query continues the search in all PDB-IDs that result from the ‘Adapated query KLIFS’.</p>
Exploiting_Unusual_Interactions _3o0g_selectivity_permissive	<p><u>Application example</u>: Exploiting Unusual Interactions for Selective Inhibitor Design <u>Query</u>: Investigation if CDK5 in complex with an ATP analogue (PDB-ID 3o0g) might become more selective upon addition of a chlorine atom which interacts with Phe80. No restriction to ligand-occupied pockets but all predicted and ligand-based to identify as many potential off-targets as possible.</p>
Exploiting_Unusual_Interactions _3o0g_selectivity_restrictive	<p><u>Application example</u>: Exploiting Unusual Interactions for Selective Inhibitor Design <u>Query</u>: Addition of the aromatic center of Phe80 as additional interaction point for a potential halogen aromatic interaction.</p>
Searching_for_Selectivity_Anchors _in_Protein_Kinases_4wkq	<p><u>Application example</u>: Searching for Selectivity Anchors in Protein Kinases <u>Query</u>: Search for potential off-targets in empty predicted binding sites by using EGFR in complex with inhibitor gefitinib (PDB-ID 4wkq) as template and screening the protein kinases stored in the KLIFS database.</p>

Searching_for_Selectivity_Anchors_in_Protein_Kinases_1xkk	<u>Application example:</u> Searching for Selectivity Anchors in Protein Kinases <u>Query:</u> Extension of the application examples query in 4wkq by one sidechain of residue Thr790 as potential selectivity anchor as known from the structure of EGFR in complex with lapatinib (PDB-ID 1xkk).
Searching_for_Reactive_Cysteins_in_Protein_Kinases_3poz	<u>Application example:</u> Searching for Reactive Cysteines in Protein Kinases <u>Query:</u> Demonstration of how the inclusion of secondary structure elements and solvent exposure enables the search for protein kinases with reactive cysteines in the neighborhood of known inhibitors based on the crystal structure of the protein kinase EGFR (PDB-ID 3poz).

**Paragraph S2.** Query definitions of runtime analysis and comparison between PELIKAN and GeoMine

*Linear - standard*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Reference ligand, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Protein, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å

Distance constraint between search points 2-3: min = 7.6 Å, max = 8.6 Å

Distance constraint between search points 3-4: min = 4.1 Å, max = 5.1 Å

*Linear – metal*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

S7

Search point 4: Original Molecule = Protein, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å  
Distance constraint between search points 2-3: min = 9 Å, max = 10 Å  
Distance constraint between search points 3-4: min = 4.1 Å, max = 5.1 Å

#### *Linear – metal, water*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Search point 4: Original Molecule = Water, Element = Undefined, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å  
Distance constraint between search points 2-3: min = 9 Å, max = 10 Å  
Distance constraint between search points 3-4: min = 6 Å, max = 7 Å

#### *Linear – metal, water, phosphorus*

Search point 1: Original Molecule = Reference ligand, Element = Phosphorus, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Search point 4: Original Molecule = Water, Element = Undefined, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Distance constraint between search points 1-2: min = 6.7 Å, max = 7.7 Å  
Distance constraint between search points 2-3: min = 9 Å, max = 10 Å  
Distance constraint between search points 3-4: min = 6 Å, max = 7 Å

#### *Star - standard*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any  
Search point 3: Original Molecule = Reference ligand, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Protein, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 5: Original Molecule = Reference ligand, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 6: Original Molecule = Protein, Element = Carbon, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å

Distance constraint between search points 1-3: min = 3.5 Å, max = 4.5 Å

Distance constraint between search points 1-4: min = 6.7 Å, max = 7.7 Å

Distance constraint between search points 1-5: min = 3.3 Å, max = 4.3 Å

Distance constraint between search points 1-6: min = 4.1 Å, max = 5.1 Å

#### *Star – metal*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Protein, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 5: Original Molecule = Reference ligand, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 6: Original Molecule = Protein, Element = Carbon, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å

Distance constraint between search points 1-3: min = 4 Å, max = 5 Å

Distance constraint between search points 1-4: min = 6.7 Å, max = 7.7 Å

Distance constraint between search points 1-5: min = 3.3 Å, max = 4.3 Å

Distance constraint between search points 1-6: min = 4.1 Å, max = 5.1 Å

#### *Star – metal, water*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Water, Element = Undefined, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 5: Original Molecule = Reference ligand, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 6: Original Molecule = Protein, Element = Carbon, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å

Distance constraint between search points 1-3: min = 4 Å, max = 5 Å

Distance constraint between search points 1-4: min = 5.3 Å, max = 6.3 Å

Distance constraint between search points 1-5: min = 3.3 Å, max = 4.3 Å

Distance constraint between search points 1-6: min = 4.1 Å, max = 5.1 Å

#### *Star – metal, water, phosphorus*

Search point 1: Original Molecule = Reference ligand, Element = Phosphorus, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Water, Element = Undefined, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 5: Original Molecule = Reference ligand, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 6: Original Molecule = Protein, Element = Carbon, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 1-2: min = 6.7 Å, max = 7.7 Å

Distance constraint between search points 1-3: min = 3 Å, max = 4 Å

Distance constraint between search points 1-4: min = 5.2 Å, max = 6.2 Å

Distance constraint between search points 1-5: min = 4.5 Å, max = 5.5 Å

Distance constraint between search points 1-6: min = 4.2 Å, max = 5.2 Å

#### *Tetrahedron – standard*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Reference ligand, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Protein, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å

Distance constraint between search points 1-3: min = 3.5 Å, max = 4.5 Å

Distance constraint between search points 1-4: min = 6.7 Å, max = 7.7 Å

Distance constraint between search points 2-3: min = 7.6 Å, max = 8.6 Å

Distance constraint between search points 2-4: min = 6.5 Å, max = 7.5 Å

Distance constraint between search points 3-4: min = 4.1 Å, max = 5.1 Å

#### *Tetrahedron – metal*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Protein, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å

Distance constraint between search points 1-3: min = 5.7 Å, max = 6.7 Å

Distance constraint between search points 1-4: min = 6.7 Å, max = 7.7 Å

Distance constraint between search points 2-4: min = 6.5 Å, max = 7.5 Å

Distance constraint between search points 2-3: min = 9 Å, max = 10 Å

Distance constraint between search points 4-3: min = 4.2 Å, max = 5.2 Å

#### *Tetrahedron – metal, water*

Search point 1: Original Molecule = Reference ligand, Element = Oxygen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Water, Element = Undefined, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 1-2: min = 6.1 Å, max = 7.1 Å

Distance constraint between search points 1-3: min = 5.7 Å, max = 6.7 Å

Distance constraint between search points 1-4: min = 5.3 Å, max = 6.3 Å

Distance constraint between search points 2-3: min = 9 Å, max = 10 Å

Distance constraint between search points 2-4: min = 4.2 Å, max = 5.2 Å

Distance constraint between search points 3-4: min = 6 Å, max = 7 Å

*Tetrahedron – metal, water, phosphorus*

Search point 1: Original Molecule = Reference ligand, Element = Phosphorus, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 2: Original Molecule = Protein, Element = Nitrogen, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 3: Original Molecule = Metal, Element = Magnesium, Interaction type = Metal, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Search point 4: Original Molecule = Water, Element = Undefined, Interaction type = Undefined, Functional group = any, Amino acid = Any, Atom name = Any, Sec. structure = Any, Amino acid location = Any

Distance constraint between search points 2-3: min = 9 Å, max = 10 Å

Distance constraint between search points 2-4: min = 4.2 Å, max = 5.2 Å

Distance constraint between search points 1-2: min = 6.7 Å, max = 7.7 Å

Distance constraint between search points 3-4: min = 6 Å, max = 7 Å

Distance constraint between search points 1-3: min = 4.2 Å, max = 5.2 Å

Distance constraint between search points 1-4: min = 5.2 Å, max = 6.2 Å





### F.3 GeoMine: interactive pattern mining of protein–ligand interfaces in the Protein Data Bank

- [D3] Diedrich, K., **Graef, J.**, Schöning-Stierand, K., Rarey, M., GeoMine: interactive pattern mining of protein–ligand interfaces in the Protein Data Bank. *Bioinformatics* 37.3 (2020), S. 424–425. DOI: [10.1093/bioinformatics/btaa693](https://doi.org/10.1093/bioinformatics/btaa693).

Nachdruck mit Erlaubnis von [D3] und der Oxford University Press.

Die *Supporting Information* ist unter [D3] verfügbar. Diese beinhaltet einzig Statistiken der durchgeführten GeoMine-Anfragen in Form einer Auflistung aller gefundenen Distanzen, Aminosäuren, Sekundärstrukturen etc.

Structural bioinformatics

# GeoMine: interactive pattern mining of protein–ligand interfaces in the Protein Data Bank

Konrad Diedrich , Joel Graef , Katrin Schöning-Stierand and Matthias Rarey \*

Universität Hamburg, ZBH – Center for Bioinformatics, 20146 Hamburg, Germany

\*To whom correspondence should be addressed.

Associate Editor: Arne Elofsson

Received on May 29, 2020; revised on July 12, 2020; editorial decision on July 22, 2020; accepted on July 24, 2020

## Abstract

**Summary:** The searching of user-defined 3D queries in molecular interfaces is a computationally challenging problem that is not satisfactorily solved so far. Most of the few existing tools focused on that purpose are desktop based and not openly available. Besides that, they show a lack of query versatility, search efficiency and user-friendliness. We address this issue with GeoMine, a publicly available web application that provides textual, numerical and geometrical search functionality for protein–ligand binding sites derived from structural data contained in the Protein Data Bank (PDB). The query generation is supported by a 3D representation of a start structure that provides interactively selectable elements like atoms, bonds and interactions. GeoMine gives full control over geometric variability in the query while performing a deterministic, precise search. Reasonably selective queries are processed on the entire set of protein–ligand complexes in the PDB within a few minutes. GeoMine offers an interactive and iterative search process of successive result analyses and query adaptations. From the numerous potential applications, we picked two from the field of side-effect analyze showcasing the usefulness of GeoMine.

**Availability and implementation:** GeoMine is part of the ProteinsPlus web application suite and freely available at <https://proteins.plus>.

**Contact:** [rarey@zbh.uni-hamburg.de](mailto:rarey@zbh.uni-hamburg.de)

**Supplementary information:** [Supplementary data](#) are available at *Bioinformatics* online.

## 1 Introduction

The understanding, manipulation and modulation of protein function require substantial structural knowledge of the protein binding sites. One of the main sources for structural data is the Protein Data Bank (PDB) (Berman, 2000). Despite its substantial growth, improvement of data quality and potential as a knowledge source, there is only a small number of tools featuring 3D geometric searching for protein–ligand interfaces based on user-defined queries (Angles *et al.*, 2020; Hendlich *et al.*, 2003; Korb *et al.*, 2016; Mobilio *et al.*, 2010; Weisel *et al.*, 2012). The first has been Relibase which was suspended in 2018. To our knowledge, Relibase was the only tool supporting atomic precision for both protein and ligand parts within the query. Besides GSP4PDB, the tools are desktop applications and not freely available. Reasonably short runtimes can be observed in the case of Prolix and CrossMiner due to their use of fingerprint techniques. The query versatility of all tools is however limited. In CrossMiner, a query consists of pharmacophore spheres which represent predefined features. Prolix and PRDB do not support an atom-level precision for protein parts of the query. GSP4PDB lacks atomic query precision. In Prolix and GSP4PDB, the user can design a query using a 2D sketcher. PRDB requires a query in SQL format. Only CrossMiner provides the possibility to construct queries in a 3D representation of a protein–ligand

complex. Regarding the lack of existing solutions, we developed GeoMine, which is based on an enhanced version of PELIKAN (Inhester *et al.*, 2017).

GeoMine is publicly available via a web-interface and part of the ProteinsPlus (Fährrolfes *et al.*, 2017; Schöning-Stierand *et al.*, 2020) server (<https://proteins.plus>). A search of geometrical, textual and numerical queries can be easily performed on protein–ligand interfaces derived from complexes contained in the PDB in a reasonably short time. In the following, we will illustrate the features of GeoMine by different use cases. Detailed descriptions of the methods are available in the PELIKAN (Inhester *et al.*, 2017) publication.

## 2 Usage and output

According to the ProteinsPlus workflow, GeoMine is started with only a PDB structure as input. 3D query design is guided by a precalculated pocket with selectable features in the embedded NGL viewer (Rose *et al.*, 2018). The query generation process allows the user to select amongst others atoms and aromatic ring centers or to place such points at unoccupied positions. Point–point constraints, i.e. interatomic distance ranges or interactions, and angle constraints between any pair of connected point–point constraints or aromatic ring normals allow the definition of any geometric arrangement.

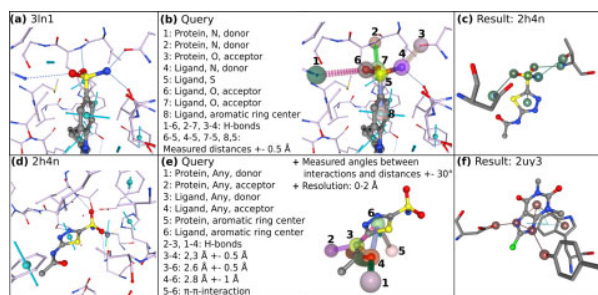


Fig 1. Query construction for the search of celecoxib (a, b) and acetazolamide (d, e) using GeoMine. (c, f) Example results from the searches of the queries defined in (b) and (e), respectively. The structures matching the query are highlighted in the results.

The resulting query is shown in the NGL viewer and simultaneously in tables for further modification by a variety of geometrical constraints, e.g. the range of a distance, and chemical properties. Main properties like the molecule type of an atom are automatically set while more discriminative ones, like the functional group of a ligand atom, can be explicitly defined by the user. Additionally, it is possible to query textual and numerical properties of the protein–ligand complex and its components, i.e. the depth of a pocket or the EC number. All search types can be used separately or together either on a PDB subselection or on the complete dataset. The first 150 matches are listed in a table and can be superimposed for visual analysis onto the 3D query in the viewer. An extensive statistics report, which allows a more sophisticated analysis of all results, as well as the pockets of the first 150 matches can be downloaded. The complete result set can be filtered continuing to search in it with the current query as a new starting point.

### 3 Applications

Since GeoMine is able to find structural similarities between binding sites of unrelated proteins, it is a valuable tool for off-target studies, e.g. with the aim of lead optimization, drug repurposing or explaining side effects. In the following, we will describe two different off-target searches showcasing the comprehensiveness of GeoMine results. Additional application examples are available in the PELIKAN (Inhester *et al.*, 2017) paper. The GeoMine database searches are performed using up to 30 cores of a 2 × Intel Xeon Gold 6248 processor (20 cores/2.5 GHz), 200 GB of main memory and a Dell 1.6TB NVMe HHL AIC PM1725b solid state drive with an xfs file system.

The identification of protein–ligand complexes with similar interaction patterns like a given query complex can generate ideas about potential off-target proteins. For our first example application, we choose the COX-2 selective inhibitor celecoxib (PDB code: 3LN1; Fig. 1a). In the precalculated pocket, the unsubstituted aryl-sulfonamide moiety of celecoxib interacts with the protein environment via 4 hydrogen bonds (Fig. 1a). A query describing partially this interacting moiety (Fig. 1b) took 45 s and resulted in 43 matches (see statistics report in Supplementary Material S1). A variety of different protein classes emerged by this search, for example carbonic anhydrase (CA II) complexed with the inhibitor acetazolamide (PDB code: 2H4N; Fig. 1c). According to studies, CA II is an off-target for unsubstituted sulfonamides like celecoxib (Weber *et al.*, 2004) implicating the enzyme in celecoxib side effects. Validation results for this query are 3LN1, 5JW1 (celecoxib cocrystallized with COX-2) and 1OQ5 (celecoxib cocrystallized with CA II). 1OQ5 was found removing one hydrogen bond from the query.

For the illustration of the second off-target search, we choose the previously found CA II complexed with acetazolamide (PDB code: 2H4N; Fig. 1d). Dependent on the arrangement of available

functional groups in a protein pocket, ligands may form different interaction patterns. To find similar geometric arrangements, we constructed a query from the complex 2H4N with parts of the ligand and hypothetical alternative interactions. It includes the ligands' thiadiazole ring center and the donor and acceptor of its acetamide fragment (Fig. 1e). Potential interaction directions are defined by angles. Geometrical flexibility is achieved by relatively large tolerance values for angles and distances. Keeping the atom elements unspecified by describing only their interaction and molecule types ensures chemical fuzziness. Low-quality results are prevented by a numerical filter. The search took 52 s and resulted in 57 matches (see statistics report in Supplementary Material S2), for instance, chitinase that binds the inhibitor theophylline (PDB code: 2UY3; Fig. 1f). According to a study, chitinase is an off-target for acetazolamide, which results as a promising lead for antifungal drug development (Schüttelkopf *et al.*, 2010). A validation result can be found in PDB file 2UY4 (acetazolamide cocrystallized with chitinase).

### 4 Conclusions

GeoMine addresses the computational challenge of efficient geometrical data mining of protein–ligand binding sites. Reasonable queries can be answered by GeoMine within seconds up to a few minutes. All structures that match the query are found and presented in a comprehensive manner. The search infrastructure of GeoMine is easy to use and publicly available as part of the ProteinsPlus web service.

### Funding

This work was supported by the German Federal Ministry of Education and Research as part of the German Network for Bioinformatics Infrastructure – de.NBI [031L0172, 031L0105].

*Conflict of Interest:* ProteinsPlus and in the NAOMI ChemBio Suite use some methods that are jointly owned and/or licensed to BioSolveIT GmbH, Germany, M.R. is a shareholder of BioSolveIT GmbH.

### References

- Angles, R. *et al.* (2020) GSP4PDB: a web tool to visualize, search and explore protein–ligand structural patterns. *BMC Bioinformatics*, **21**, 85.
- Berman, H.M. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
- Fährrolfes, R. *et al.* (2017) ProteinsPlus: a web portal for structure analysis of macromolecules. *Nucleic Acids Res.*, **45**, 337–343.
- Hendlich, M. *et al.* (2003) Relibase: design and Development of a database for comprehensive analysis of protein–ligand interactions. *J. Mol. Biol.*, **326**, 607–620.
- Inhester, T. *et al.* (2017) Index-based searching of interaction patterns in large collections of protein–ligand interfaces. *J. Chem. Inf. Model.*, **57**, 148–158.
- Korb, O. *et al.* (2016) Interactive and versatile navigation of structural databases. *J. Med. Chem.*, **59**, 4257–4266.
- Mobilio, D. *et al.* (2010) A protein relational database and protein family knowledge bases to facilitate structure-based design analyses. *Chem. Biol. Drug Des.*, **76**, 142–153.
- Rose, A.S. *et al.* (2018) NGL viewer: web-based molecular graphics for large complexes. *Bioinformatics*, **34**, 3755–3758.
- Schöning-Stierand, K. *et al.* (2020) ProteinsPlus: interactive analysis of protein–ligand binding interfaces. *Nucleic Acids Res.*, **48**, 48–53.
- Schüttelkopf, A.W. *et al.* (2010) Acetazolamide-based fungal chitinase inhibitors. *Bioorg. Med. Chem.*, **18**, 8334–8340.
- Weber, A. *et al.* (2004) Unexpected nanomolar inhibition of carbonic anhydrase by COX-2-selective celecoxib: new pharmacological opportunities due to related binding site recognition. *J. Med. Chem.*, **47**, 550–557.
- Weisel, M. *et al.* (2012) PROLIX: rapid mining of protein–ligand interactions in large crystal structure databases. *J. Chem. Inf. Model.*, **52**, 1450–1461.



## F.4 ProteinsPlus: a comprehensive collection of web-based molecular modeling tools

- [D4] Schöning-Stierand, K., Diedrich, K., Ehrt, C., Flachsenberg, F., **Graef, J.**, Sieg, J., Penner, P., Poppinga, M., Ungethüm, A., Rarey, M., ProteinsPlus: a comprehensive collection of web-based molecular modeling tools. *Nucleic Acids Res.* 50.W1 (2022), W611–W615. DOI: 10.1093/nar/gkac305.

Nachdruck mit Erlaubnis von [D4] und der Oxford University Press.

# ProteinsPlus: a comprehensive collection of web-based molecular modeling tools

Katrin Schöning-Stierand<sup>1,†</sup>, Konrad Diedrich<sup>1,†</sup>, Christiane Ehrh<sup>1,†</sup>,  
Florian Flachsenberg<sup>1,†</sup>, Joel Graef<sup>1,†</sup>, Jochen Sieg<sup>1,†</sup>, Patrick Penner<sup>1</sup>,  
Martin Poppinga<sup>1,2</sup>, Annett Ungethüm<sup>3</sup> and Matthias Rarey<sup>1,\*</sup>

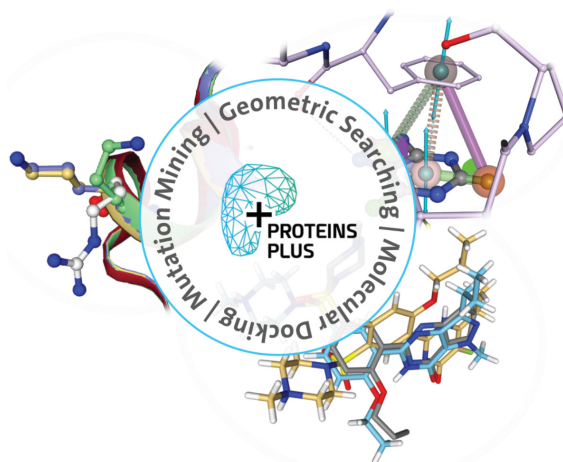
<sup>1</sup>Universität Hamburg, ZBH - Center for Bioinformatics, Bundesstraße 43, 20146 Hamburg, Germany, <sup>2</sup>Universität Hamburg, Department of Informatics, Vogt-Kölln-Straße 30, 22527 Hamburg, Germany and <sup>3</sup>Universität Hamburg, Center for Data and Computing in Natural Sciences (CDCS), Notkestraße 11, 22607 Hamburg, Germany

Received February 26, 2022; Revised April 05, 2022; Editorial Decision April 10, 2022; Accepted April 19, 2022

## ABSTRACT

Upon the ever-increasing number of publicly available experimentally determined and predicted protein and nucleic acid structures, the demand for easy-to-use tools to investigate these structural models is higher than ever before. The ProteinsPlus web server (<https://proteins.plus>) comprises a growing collection of molecular modeling tools focusing on protein–ligand interactions. It enables quick access to structural investigations ranging from structure analytics and search methods to molecular docking. It is by now well-established in the community and constantly extended. The server gives easy access not only to experts but also to students and occasional users from the field of life sciences. Here, we describe its recently added new features and tools, beyond them a novel method for on-the-fly molecular docking and a search method for single-residue substitutions in local regions of a protein structure throughout the whole Protein Data Bank. Finally, we provide a glimpse into new avenues for the annotation of AlphaFold structures which are directly accessible via a RESTful service on the ProteinsPlus web server.

## GRAPHICAL ABSTRACT



## INTRODUCTION

The ProteinsPlus (1,2) web server, openly available at <https://proteins.plus>, offers molecular modeling support for all protein structures that are publicly available as PDB files in the Protein Data Bank (PDB) (3). Usually, workflows for structure-based design necessitate a comprehensive user knowledge of different molecular modeling tools. For example, predicting potential binding sites, finding similar binding sites for ensemble docking, and molecular docking of small molecules of interest into a binding site requires access to and knowledge of a high number of tools with a multitude of parameters. Furthermore, researchers must rely on their computational resources. With the ProteinsPlus server, these shortcomings are overcome by enabling users to perform all these steps via one unique and easily accessible interface. The server is under constant development including

\*To whom correspondence should be addressed. Tel: +49 40 428387350; Fax: +49 40 428387352; Email: [matthias.rarey@uni-hamburg.de](mailto:matthias.rarey@uni-hamburg.de)

†The authors wish it to be known that, in their opinion, the first six authors should be regarded as Joint First Authors.

Present address: Florian Flachsenberg, BioSolveIT GmbH, An der Ziegelei 79, 53757 St. Augustin, Germany.

© The Author(s) 2022. Published by Oxford University Press on behalf of Nucleic Acids Research.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License

(<http://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact [journals.permissions@oup.com](mailto:journals.permissions@oup.com)

fine-tuning, feature extensions, and the integration of additional modeling tools.

Here, we offer insights into feature extensions for the structural multi-purpose comparison tool GeoMine, the newly integrated molecular docking tool JAMDA and MicroMiner - a method that can be used to screen for single-residue substitutions in local protein environments in the whole PDB.

Finally, the artificial intelligence-based protein structure predictions by AlphaFold (currently predicted by AlphaFold Monomer v2.0) enable unprecedented access to high-quality models of proteins of yet unknown structure (4). These models are now readily accessible via the AlphaFold Protein Structure Database (<https://alphafold.ebi.ac.uk/>) and can be directly imported via the provided REST API.

## MATERIAL AND METHODS: EXTENSIONS AND NOVEL TOOLS

### GeoMine

From the analysis of binding sites to investigations of geometric preferences for interactions, the ever-increasing number of molecular structures in the PDB offers a multitude of possibilities for in-depth studies of binding sites, their properties and their similarities. This requires comprehensive search capabilities. With GeoMine (5,6), we have developed a search engine that allows for the generation of and the search for atom-based geometric query patterns and an extensive textual and numerical filtering of the PDB. The query atoms can be described manually or automatically with varying degrees of detail, from major properties like the corresponding molecule type, i.e. nucleic acid, protein, ligand, water, or metal, to more restrictive ones, e.g. the molecular surface contribution of a protein or nucleic acid atom. Further feature points like aromatic ring centers can be added to the query and described equally. Distance ranges or hydrogen bond, pi-pi, pi stacking, pi-cation, metal and ionic interactions between atoms and feature points can be introduced into the query, and angle ranges between those can be specified. With the combination of all these features, almost any 3D pattern can be designed and searched in the entire PDB.

In the *ProteinsPlus* user interface, the query can be created in a 3D viewer from scratch by the placement of new atoms and feature points or by selecting those in a visualized binding site of a PDB/AlphaFold structure or any uploaded structure file. For this structure, GeoMine predicts binding pockets with interactions and hydrogen atoms using the tools DoGSiteScorer (7) and Protoss (8,9), respectively. If a ligand is present but no pocket has been calculated, a pocket is defined using a radius of 6.5 Å of any ligand atom. The computing times for the iterative search of over one million preprocessed bindings sites depends on the specificity of the query. Most requests can be processed in the range of minutes. For each detected hit, the root-mean-square deviation (RMSD) between the query and the part of the site matching the query is calculated enabling a ranking of the results by geometric fit. The 150 best results are listed in a table and can be visually inspected superimposed to the query in the NGL viewer. Different visualization options are available,

for example, choice of residues (complete pocket or only of the residues that match the query). The 150 best-matching pockets can be downloaded in PDB format together with a report containing the statistical overview of all results. The statistics report lists the PDB IDs and ligand names of all found pockets, the distributions of the RMSD values, and the properties of all matched atoms, feature points, distances, interactions, and angles of the query, e.g. the functional group distribution for a matched ligand atom. The user interface with a query history allows a continuous refinement of the results providing an interactive workflow of query modification and subsequent searching in the results. With this tool, protein function or ligand off-targets can be discovered by searching similar binding site properties in 3D space. GeoMine has recently been applied for a detailed analysis of structural features in protein kinase structures (5).

### JAMDA

Protein-ligand docking is one of the core tasks in structure-based drug design. With JAMDA, we aimed for the implementation of a fully-automated docking workflow in the *ProteinsPlus* server that does not only provide the actual docking algorithm but also encompasses all necessary preprocessing steps, including protonation state assignment and calculation of hydrogen coordinates for the protein (8), prediction of protonation and tautomeric states of the molecules to be docked (10), as well as the generation of 3D coordinates/conformations (11). While a certain degree of manual intervention is possible, our goal was to provide a fully automated workflow with optimized default parameters. This enables even less experienced users to derive potential binding modes of small molecules in the binding site of interest. From the analysis of structure-activity relationships to the test of new binding hypotheses, the established pipeline offers unlimited access to predicted binding modes.

JAMDA docking combines the TriX docking algorithm (12,13) for initial pose generation with the JAMDA scoring function (14), and our novel LSL-BFGS optimization algorithm (14,15) for scoring and pose optimization. Initially, conformers for the molecule to be docked are generated with the Conformer (11). The raw poses are subjected to a scoring and optimization cascade using the JAMDA scoring function to refine and rank the docking poses.

On *ProteinsPlus*, JAMDA allows for a fully automated docking: Only the protein, the binding site, and the molecules to be docked must be provided by the user. The binding site can be defined based on a known ligand or selected from the pocket definitions in *ProteinsPlus* (1) (e.g. predicted by DoGSiteScorer (16)). To enable the user to manually adjust the binding sites, all ligand-based and predicted binding sites which do not originate from GeoMine are editable by the user in the pockets tab by clicking on the pencil symbol of the pocket of interest in the upper right corner. Neither the protein nor the molecules to be docked must be manually prepared by the user because this is an integral part of the JAMDA docking workflow: The protein is prepared by assigning likely protonation states using Protoss (8). Furthermore, only structurally relevant water molecules and small molecules that are common cofac-

tors are kept. The molecules to be docked can be provided by picking a ligand from the NGL viewer for redocking studies or by uploading molecules in any common molecular file format (including SMILES without coordinates). Their predominant protonation and tautomeric states are predicted with UNICON (10) prior to docking. Most of these preprocessing steps can optionally be customized by the user.

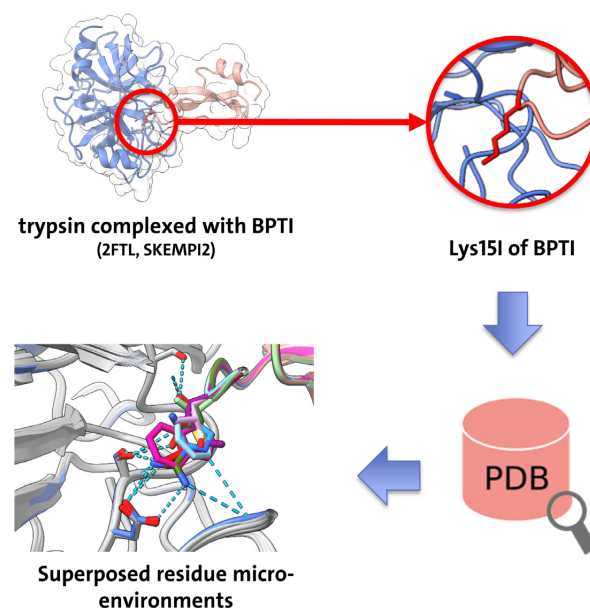
The preprocessing and docking are performed on the server and, currently, up to five molecules may be docked simultaneously. In the *ProteinsPlus* web interface, the resulting docking poses are shown in a table (with JAMDA score and the RMSD if a redocking was performed) and visualized in the NGL viewer panel for interactive analyses. They can also be downloaded for alternative visualizations and further processing. In consequence, JAMDA offers a pipeline for molecular docking that provides reliable results even in the absence of substantial knowledge regarding molecular modeling tools.

### MicroMiner

MicroMiner searches for mutations in protein structure databases. On *ProteinsPlus*, it screens for single-residue substitutions in the experimental structures of the entire PDB. Retrieved mutant structures can be easily analyzed and compared to the wildtype through automatically generated superpositions in the NGL viewer. The tool focuses on the local 3D micro-environment of single residues in a query protein. It searches the protein structure database for similar local environments with a mutated central residue. For reasonably large wildtype protein structures it is feasible to search for substitutions of all residues in the query at once. In this way, a user can comprehensively explore the wealth of experimental protein structures that exemplify the local effects of mutations through the interactive web interface.

MicroMiner originates from the ASCONA (17) and SIENA (18) technology for binding site similarity search and ensemble compilation. However, instead of focusing on the protein environment of ligands, MicroMiner uses the local 3D micro-environment of any individual residue as the query to search for residues embedded in similar local arrangements. A database search starts by selecting a query residue from which the local 3D protein neighborhood within a distance cutoff (default 6.5 Å) represents the query micro-environment. The connected sequence fragments of this environment are used to identify candidate protein structures with similar sequence fragments in the database. Second, all potential matches are identified by residue-wise sequence alignments. A subsequent fuzzy geometric filter based on the C $\alpha$  atom orientation and distances of the matching sequence fragments ensures a reasonably similar structural arrangement while tolerating structural changes upon mutation. Thus, we identify local micro-environments with a high sequence and structural similarity. Figure 1 shows the MicroMiner workflow.

Within the *ProteinsPlus* server, the user can select single residues of interest or all residues in the input structure to be searched against the PDB. Searching for all residues is feasible within one minute or less on average, depending on the size of the input protein and the number of similar



**Figure 1.** MicroMiner workflow. With the local 3D micro-environment of a selected query residue, the PDB is searched. Structures from the database containing a similar micro-environment identical in sequence except for the query residue position are retrieved and superposed for analysis. In this way, MicroMiner yields structure ensembles exemplifying the local effects of mutations.

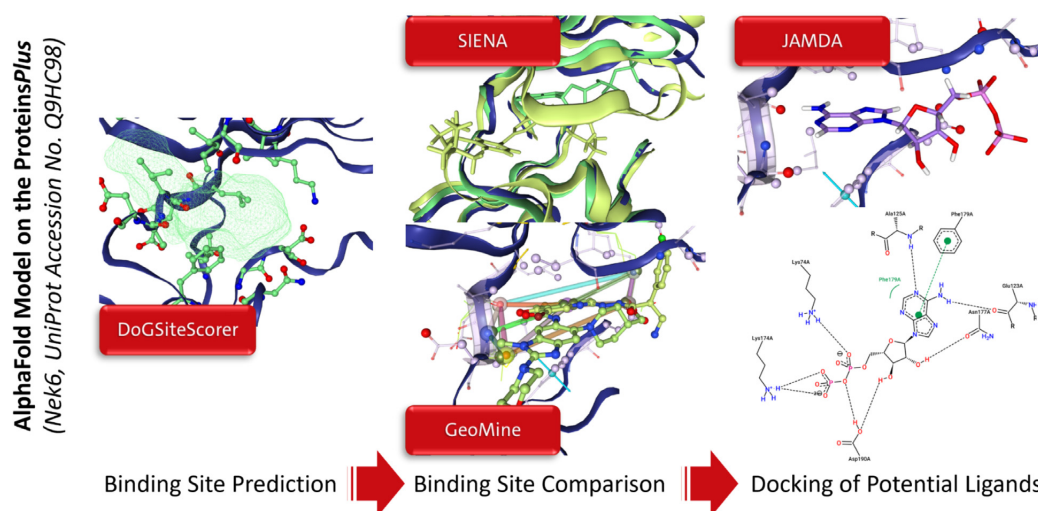
micro-environments in the PDB. The protein structures of retrieved micro-environments can be explored interactively as a structure ensemble in the 3D viewer and sorted by properties of interest, for example, the RMSD of the local environments to investigate the structural effects of mutations. Further applications are the search for highly conserved regions in protein structures, comparisons of the impact of conservative and radical substitutions, or the investigation of structural effects upon substitution for evaluating the reliability and accuracy of computationally generated models of single-residue substitutions.

### Integration of AlphaFold structures

The inclusion of AlphaFold protein structure models (4) (<https://alphafold.ebi.ac.uk/>) in the *ProteinsPlus* web server enables easy access to machine learning-based predictions of previously unknown structures. The models are accessible on our web server by entering the UniProt Accession Number on the landing page or uploading a preprocessed structure. The user can analyze these structures in the same way publicly available PDB structures can be analyzed by making use of all applicable capabilities of the *ProteinsPlus* tools.

Besides the structural uncertainty of AlphaFold structures (19), the missing ligand annotations are a major drawback. This led to the development of the database AlphaFill (20) which annotates the 3D models with cofactors and metal ions and transfers them into the structure assisting in the functional annotation of the models. However, this annotation procedure was only followed for structures that show an identity of at least 35% to known 3D structures





**Figure 2.** This workflow shows exemplary results for structural investigations of the AlphaFold model for the Nek6 (UniProt Accession Number Q9HC98). First, the user can detect druggable binding sites with DoGSiteScorer. Pocket ‘P\_2’ which was predicted as druggable is depicted in green on the right. Next, the pocket can be used for a SIENA search for similar binding sites. Shown are two matches from this analysis with Nek7 structures: 2WQN with ADP and 6S73 in complex with the ligand with the ID F9N in the PDB. GeoMine can be applied for more specific user-defined searches in the binding sites of the PDB. Using a geometric query annotating solvent-exposed potentially interacting atoms and their distances, we found 116 pockets with a similar geometry in the PDB (e.g. cAMP-dependent protein kinase A with the PDB ID 7BAQ, PDB ligand ID T82 or interleukin-1 receptor-associated kinase 4 with the PDB ID 6O94, PDB ligand ID LRS). The corresponding query can be found in the Supplementary Data for upload to the GeoMine tool on the ProteinsPlus for this structure. Interesting small molecules from the identified similar sites can be downloaded and subsequently be used for molecular docking with JAMDA. The figures on the right show the second highest-scoring predicted binding mode for ADP in the binding site of Nek6 and its 2D interaction visualization with PoseView (21).

stored in the PDB and restricted to common cofactors and ions with potentially functional roles. For researchers interested in the structural annotation of structures that have no known homologs in the PDB, the ProteinsPlus web service comes in handy. It enables on-the-fly prediction of binding sites with DoGSiteScorer, retrieval of similar binding sites with SIENA, the identification of further potentially interesting ligands by user-defined GeoMine queries, and the molecular docking of these ligands into the AlphaFold model with JAMDA, see Figure 2.

### Ligand annotation for AlphaFold models

Given a protein of interest, e.g. the human protein kinase NIMA-related kinase 6 (Nek6), we can start our ProteinsPlus investigations by providing its UniProt Accession Number Q9HC98 and entering the structural analysis mode of the web service. Next, we can predict potential binding sites using DoGSiteScorer. These predicted sites can be used to search for potential ligands with SIENA. By selecting, for example, the pocket named ‘P\_2’ and performing a SIENA search for this predicted binding site, we can retrieve similar sites in complex with various ligands. Besides ADP (the annotation which was also found by AlphaFill), we find similar kinase binding sites in complex with further ligands, in this case, the inhibitor with the PDB ligand ID F9N in complex with Nek2 and Nek7. The active site sequence identity is 94%. The retrieved aligned complexes can be downloaded, together with the corresponding ligand SDF files. The results also enable the exploration of structural flexibility of similar binding sites that can be used, e.g. for the generation of other conformational states that are not covered

in the AlphaFold database by homology modeling based on the identified structures.

The ligands retrieved from the SIENA run can either be transferred into the binding site based on the resulting alignment or using the on-the-fly docking tool JAMDA. It can be applied to find whether the found ligands from similar sites can be accommodated in the model’s binding site. However, care should be taken regarding the model quality of the binding site residues as this can have a huge impact on the docking performance. Some preprocessing steps of the original AlphaFold structure might be necessary to obtain reliable ligand binding modes (22).

The search for similar binding sites using the ProteinsPlus, however, is not restricted to binding sites with a high sequence identity. GeoMine can be applied to generate user-defined queries that search for geometric patterns of interacting binding site residues in nearly one million binding sites (predicted or ligand-annotated) in the PDB. For our example protein kinase, additional GeoMine queries result in the identification of further protein kinases in complex with inhibitors which can be used as idea generators for *in silico* drug design.

### SUMMARY AND OUTLOOK

The ProteinsPlus web server offers a unique access point to protein structure and protein–ligand complex data processing on the worldwide web. Current developments with only conservative extensions of the user interface enable even broader access to molecular modeling tools which usually require comprehensive user knowledge. Furthermore, steady improvements and feature extensions based

on suggestions of users render it a lively and well-kept platform. To support users in getting started with the web server, we offer comprehensive documentation of the provided services (<https://proteins.plus/help/index>) and hands-on tutorials (<https://proteins.plus/help/tutorial>). As with all computational modeling approaches, the tools behind ProteinsPlus have their limitations. All users are asked to consult the corresponding methods' publication for more details on the respective restrictions and application domains.

Besides the introduction of new features for GeoMine and the integration of the novel methods JAMDA and MicroMiner, we are in a constant process of elaborating the web server, its tool base, and its potential use cases. The first inclusion of AlphaFold structures in the web server opens new avenues for structural explorations that have not yet been fully explored. With numerous extensions in mind, including 2D and automated query generation in GeoMine or multiple mutations search in MicroMiner, we hope to create a steadily growing, easy-to-use modeling infrastructure for the life science community.

## DATA AVAILABILITY

ProteinsPlus is a publicly available web-based protein structure analysis service, available at <https://proteins.plus>.

## SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

## FUNDING

Development of ProteinsPlus was supported by de.NBI (in part); German Federal Ministry of Education and Research (BMBF) [031L0105 to K.S. and J.S.]; Development of GeoMine was supported by the German Federal Ministry of Education and Research as part of CompLS and de.NBI [031L0172, 031L0105 to K.D. and J.G.]; Development of MicroMiner was supported by the German Federal Ministry of Education and Research (BMBF) as part of protPSI [031B0405B to J.S.]; DASHH: Data Science in Hamburg - Helmholtz Graduate School for the Structure of Matter, funded by the Helmholtz Association [HIDSS-0002 to C.E.]; Center for Data and Computing in Natural Sciences (CDCS), funded by Authority for Science, Research and Equality of the Free and Hanseatic City of Hamburg (BWFG) [LFF-HHX-03 to A.U.]. Funding for open access charge: Internal university funds.

**Conflict of interest statement.** ProteinsPlus uses some methods that are jointly owned and/or licensed to BioSolveIT GmbH, Germany, M.R. is a shareholder of BioSolveIT GmbH.

## REFERENCES

1. Schönig-Stierand, K., Diedrich, K., Fährrolfes, R., Flachsenberg, F., Meyder, A., Nittinger, E., Steinegger, R. and Rarey, M. (2020) ProteinsPlus: interactive analysis of protein–ligand binding interfaces. *Nucleic Acids Res.*, **48**, W48–W53.

2. Fährrolfes, R., Bietz, S., Flachsenberg, F., Meyder, A., Nittinger, E., Otto, T., Volkamer, A. and Rarey, M. (2017) Proteins plus: a web portal for structure analysis of macromolecules. *Nucleic Acids Res.*, **45**, W337–W343.
3. Bertram, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H. and Shindyalov, I.N. (2000) The protein data bank ([www.rcsb.org](http://www.rcsb.org)). *Nucleic Acids Res.*, **28**, 235–242.
4. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Židek, A., Potapenko, A. *et al.* (2021) Highly accurate protein structure prediction with alphafold. *Nature*, **596**, 583–589.
5. Graef, J., Ehrt, C., Diedrich, K., Poppinga, M., Ritter, N. and Rarey, M. (2022) Searching geometric patterns in protein binding sites and their application to data mining in protein kinase structures. *J. Med. Chem.*, **65**, 1384–1395.
6. Diedrich, K., Graef, J., Schönig-Stierand, K. and Rarey, M. (2021) GeoMine: interactive pattern mining of protein–ligand interfaces in the protein data bank. *Bioinformatics*, **37**, 424–425.
7. Volkamer, A., Kuhn, D., Rippmann, F. and Rarey, M. (2012) DoGSiteScorer: a web server for automatic binding site prediction, analysis and druggability assessment. *Bioinformatics*, **28**, 2074–2075.
8. Bietz, S., Urbaczek, S., Schulz, B. and Rarey, M. (2014) Protoss: a holistic approach to predict tautomers and protonation states in protein–ligand complexes. *J. Cheminform.*, **6**, 12.
9. Lippert, T. and Rarey, M. (2009) Fast automated placement of polar hydrogen atoms in protein–ligand complexes. *J. Cheminform.*, **1**, 13.
10. Sommer, K., Friedrich, N.-O., Bietz, S., Hilbig, M., Inhester, T. and Rarey, M. (2016) UNICON: a powerful and Easy-to-Use compound library converter. *J. Chem. Inf. Model.*, **56**, 1105–1111.
11. Friedrich, N.-O., Flachsenberg, F., Meyder, A., Sommer, K., Kirchmair, J. and Rarey, M. (2019) Conformer: a novel method for the generation of conformer ensembles. *J. Chem. Inf. Model.*, **59**, 731–742.
12. Schlosser, J. and Rarey, M. (2009) Beyond the virtual screening paradigm: structure-based searching for new lead compounds. *J. Chem. Inf. Model.*, **49**, 800–809.
13. Henzler, A.M., Urbaczek, S., Hilbig, M. and Rarey, M. (2014) An integrated approach to knowledge-driven structure-based virtual screening. *J. Comput. Aided. Mol. Des.*, **28**, 927–939.
14. Flachsenberg, F., Meyder, A., Sommer, K., Penner, P. and Rarey, M. (2020) A consistent scheme for gradient-based optimization of protein–ligand poses. *J. Chem. Inf. Model.*, **60**, 6502–6522.
15. Flachsenberg, F. and Rarey, M. (2021) LSOpt: an open-source implementation of the step-length controlled LSL-BFGS algorithm. *J. Comput. Chem.*, **42**, 1095–1100.
16. Volkamer, A., Griewel, A., Grombacher, T. and Rarey, M. (2010) Analyzing the topology of active sites: on the prediction of pockets and subpockets. *J. Chem. Inf. Model.*, **50**, 2041–2052.
17. Bietz, S. and Rarey, M. (2015) ASCONA: rapid detection and alignment of protein binding site conformations. *J. Chem. Inf. Model.*, **55**, 1747–1756.
18. Bietz, S. and Rarey, M. (2016) SIENA: efficient compilation of selective protein binding site ensembles. *J. Chem. Inf. Model.*, **56**, 248–259.
19. Perrakis, A. and Sixma, T.K. (2021) AI revolutions in biology. *EMBO Rep.*, **22**, e54046.
20. Hekkelman, M.L., de Vries, I., Joosten, R.P. and Perrakis, A. (2021) AlphaFill: enriching the alphafold models with ligands and co-factors. bioRxiv doi: <https://doi.org/10.1101/2021.11.26.470110>, 27 November 2021, preprint: not peer reviewed.
21. Stierand, K., Maass, P.C. and Rarey, M. (2006) Molecular complexes at a glance: automated generation of two-dimensional complex diagrams. *Bioinformatics*, **22**, 1710–1716.
22. Skolnick, J., Gao, M., Zhou, H. and Singh, S. (2021) AlphaFold 2: why it works and its implications for understanding the relationships of protein sequence, structure, and function. *J. Chem. Inf. Model.*, **61**, 4827–4831.

## F.5 Database and Workflow Optimizations for Spatial-Geometric Queries in GeoMine

- [D5] Poppinga, M., **Graef, J.**, Diedrich, K., Rarey, M., Ritter, N., „Database and Workflow Optimizations for Spatial-Geometric Queries in GeoMine“. In: *Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings*. Bd. 3630. CEUR Workshop Proceedings. Maarburg, Deutschland: CEUR-WS.org, 2023, S. 86–97. URL: <https://ceur-ws.org/Vol-3630/LWDA2023-paper8.pdf>.

Nachdruck mit Erlaubnis von [D5] und den CEUR Workshop Proceedings.

# Database and Workflow Optimizations for Spatial-Geometric Queries in GeoMine

Martin Poppinga<sup>1,2</sup>, Joel Graef<sup>2</sup>, Konrad Diedrich<sup>2</sup>, Matthias Rarey<sup>2</sup> and Norbert Ritter<sup>1</sup>

<sup>1</sup>Universität Hamburg, Fachbereich Informatik, 22527 Hamburg, Germany

<sup>2</sup>Universität Hamburg, ZBH – Center for Bioinformatics, 20146 Hamburg, Germany

## Abstract

Addressing computational problems in science often involves customized algorithmic approaches, which can lead to overlooking well-established solutions in data management and storage. When scientific datasets grow, these customized approaches may struggle to query data efficiently. Effective data management is essential for ensuring accurate and fast analysis of scientific data. Describing changes in the *GeoMine* software, this paper highlights the potential for improvements in data-driven science.

*GeoMine* enables spatial-geometric searches in three-dimensional molecular space, facilitating tasks such as pharmaceutical drug discovery by finding similar geometric patterns in protein-ligand complexes. The original *GeoMine* application utilized a relational database solely for fundamental data storage and combined it with a tailored algorithmic pattern-matching strategy, leaving room for improvements. This work presents a technical overview of database and workflow optimizations in *GeoMine* to handle the increasing data size. Our improvements focus on moving the main computational tasks from the application level to the database system and optimizing the database utilization. A new query design, better utilization of indexes, and optimizations in textual queries led to a 15x speedup in our experiments, reducing the mean runtime of queries to under 8 seconds.

The presented improvements are essential for *GeoMine* to be offered as a service-oriented web application. The success of these improvements highlights the significance of database optimization in science, demonstrating the potential and necessity of proper data management.

## Keywords

Database optimization, query optimization, data management, databases for bioinformatics

## 1. Introduction

Mining huge datasets is a central task in research. Analyzing molecular interactions between proteins and small organic molecules is essential for understanding disease treatments and advancing medical research. This includes searching for spatial similarities and geometric arrangements, which can provide vital insights into the functional aspects of proteins. Results can be used for further research, for example, in pharmaceutical drug discovery or biotechnology [1]. With the growth of accessible datasets, searching for patterns in this data becomes

LWDA'23: Lernen, Wissen, Daten, Analysen. October 09–11, 2023, Marburg, Germany

✉ martin.poppinga@uni-hamburg.de (M. Poppinga); graef@zbh.uni-hamburg.de (J. Graef);

diedrich@zbh.uni-hamburg.de (K. Diedrich); rarey@zbh.uni-hamburg.de (M. Rarey);

norbert.ritter@uni-hamburg.de (N. Ritter)

🆔 0000-0001-8529-8376 (M. Poppinga); 0000-0001-8327-4936 (J. Graef); 0000-0001-8171-0888 (K. Diedrich);

0000-0002-9553-6531 (M. Rarey); 0000-0002-1502-1395 (N. Ritter)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

increasingly challenging [2, 3]. Besides the continuous growth of available experimental data, machine-learning-based structure predictions add millions of new structural models [4].

*GeoMine* [3] is an application enabling a visual-guided geometric pattern search of molecular data in three-dimensional space. It is embedded in the *proteins.plus*<sup>1</sup> server [5], a collection of different web-based tools for various tasks in protein-based research. The server is a free service based on publicly available datasets handling over half a million page requests per year. The back end of *GeoMine* was derived in prior work from the *PELIKAN* application developed in the same group [6], which was utilizing a custom algorithmic approach for query processing. With the Protein Data Bank (PDB) [7] as a fast-growing dataset underlying *GeoMine* and the shift from a desktop application to a server-based approach, *GeoMine* required an overhaul of the original query workflow to maintain the ability to provide results in a fast manner.

With this work, we investigate the potential of adopting a database-driven architecture, focusing on the database as the main part of query execution and reducing application-side processing. We were able to reduce the mean runtime in our experiments from about 2 minutes per query to less than 8 seconds, utilizing changes in the workflow and database optimizations. As we present in this work, a substantial performance enhancement has been achieved by shifting to a more database-centric method.

The paper is organized as follows: Section 2 provides an overview of the field of work, the data structure, and the query design; Section 3 details the improvements made to the query workflow and database optimizations; Section 4 presents and discusses the experimental results; Section 5 concludes the paper and outlines future work.

## 2. Background and Related Work

### 2.1. Data Management and Storage

Data management in scientific research involves the systematic collection, organization, storage, and sharing of data to facilitate its reusability and ensure the reproducibility of research findings. In the context of our work, which focuses on querying structured data sets, the storage aspect is particularly important. In the scientific domain, many existing applications are designed for single-user usage, often locally storing data in various formats or utilizing object stores with limited retrieval possibilities [8, 9]. For structured data, *Relational Database Management Systems* (RDBMS) are the most commonly used systems, providing robust and efficient solutions. Commonly, embedded systems are used, such as SQLite [10] for applications with smaller or medium-sized data sizes or DuckDB [9] for analytical workloads. For *Online Transaction Processing* (OLTP) workloads which require fast query performance and regular updates, server-based RDBMS are a popular choice. Large analytical queries are often served by designated *Online Analytical Processing* (OLAP) systems such as data warehouses, which are often proprietary solutions. For handling large-scale semi-structured datasets, NoSQL systems are frequently used, with columnar and graph databases being popular for analytical queries. The choice of data management and storage solutions is crucial to ensure efficient processing, reduced resource consumption, and accurate and fast analysis of scientific data.

---

<sup>1</sup><https://proteins.plus>

**PostgreSQL** GeoMine utilizes PostgreSQL [11], a robust and widely accessible open-source database management system. As multiple users can access a web-based application such as GeoMine at the same time, the ability of a client-server-based database system to handle multiple queries efficiently in parallel is required. PostgreSQL's widespread adoption [12] enables cloud-agnostic hosting on every major platform since most cloud platforms offer PostgreSQL solutions or other PostgreSQL-compatible scalable databases. Additionally, setting up on-premise or local instances is straightforward. PostgreSQL is suited for OLTP and also OLAP workloads [13]. The required workloads here can be depicted in the area of OLAP, given the potential complexity of the designed queries. However, given the use case of an interactive search mask for a web service, fast responses are a requirement. PostgreSQL's efficient query planning and extensibility for additional approaches (e.g., PostGIS [14] for spatial data or Citus [15] for distributed and columnar storage) make it a suitable foundation for GeoMine's use case.

## 2.2. Protein-Ligand Interactions and Binding Pockets

Protein-ligand interactions are of particular interest in biomolecular and pharmaceutical research. Ligands are small molecules that can interact and bind to the generally much larger proteins. Protein complexes can contain multiple pockets of varying sizes, partly containing ligands. Drug molecules used as pharmaceuticals are generally designed to target specific proteins. Researchers can gain valuable insights by investigating specific three-dimensional structures and searching for potential candidates to bind with these proteins.

**Protein Data Bank** The PDB [7, 2], established in 1971, is a comprehensive repository of 3D structural data of proteins and nucleic acids. The structural information is primarily obtained through experimental methods, predominantly X-ray crystallography, from research facilities worldwide [2]. As a freely available resource, the PDB has become vital for research in various fields by providing atomic-scale structural insights for drug design and understanding biological processes, containing more than 200,000 structures as of April 2023. Further, with the advantage of *Computed Structure Models*, which are protein structure predictions, for example, by *AlphaFold2* [4], additional datasets with about 1,000,000 structures are available now [2].

## 2.3. GeoMine

Discovering similar structures across distinct complexes or finding molecules that bind to a specific pocket of interest is a major task in medical research. GeoMine is able to construct comprehensive databases derived from the PDB and supports exploring these databases with a web-based search interface. [3]

The preprocessing and database creation procedures employ components of the NAOMI library [16]. For example, pockets are classified in a complex preprocessing pipeline when constructing the database [3]. Central components are the DoGSite algorithm [17], which identifies empty binding pockets within protein structures, and the calculation of interactions [18].

The central part of the search and unique key feature is the ability to specify geometric properties, for instance, distances and angles between any points, such as atoms. Further, point properties can be specified, such as an atom's chemical element and interactions between

points. This way, precise structural motifs (structural patterns) in protein-ligand complexes can be searched. While GeoMine’s predecessor *PELIKAN* was a single-user application based on an integrated *SQLite* [10] database, the *GeoMine* back end is aimed at a server-focused architecture. In the initial development of GeoMine [3], the query execution capabilities of *PELIKAN* were extended for new functionality but were not changed in structure to adapt to the new architecture.

**Database Design** For our experiments in Section 4, we used a PostgreSQL15 database created with the PDB dataset from October 2022. For querying the dataset, the database can be considered read-only. The database requires approximately 165GB of disk space.

For the geometric search, we focus on two tables. The first table, the *point table*, comprises all atoms and other definable points, such as the center of aromatic rings. It contains 340,716,693 searchable entries. These points are distributed across 1,382,853 distinct pockets, which serve as containers for groups of points. The largest pocket identified in our dataset contains 20,306 points, while the smallest pocket only holds 9 points. Each entry in the point table has a unique identifier, references the containing pocket, and contains various other fields with properties per point. Some properties, such as the accessible surface area of an atom, are floating point numbers. Other attributes, such as the chemical element, contain only a few distinct values, represented as integers or short strings.

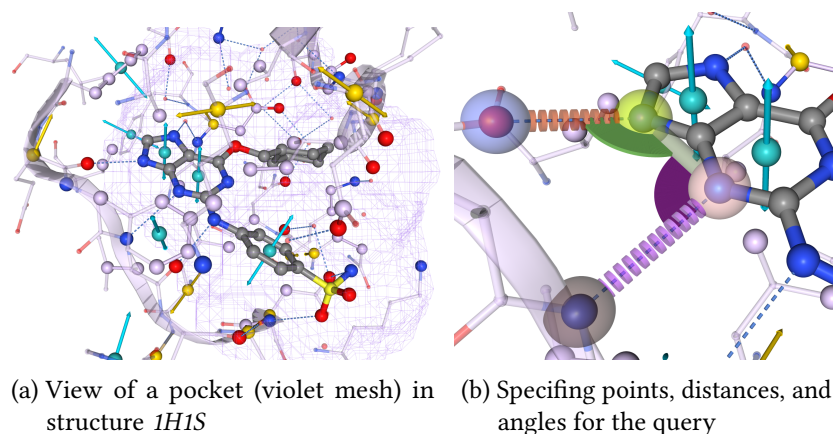
The second table, the *interaction table*, stores pre-calculated interactions [18]. These interactions represent noteworthy connections between two points, for example, hydrogen bonds. 13,018,225 point pairs are stored here.

**Query Creation** When creating a query, users can specify multiple constraints. The most fundamental categories encompass *Textual and Numerical Searches*, wherein metadata filters at the protein structure or pocket level can be defined. Users can directly pre-select several structures or create various filters, such as the minimum number of particular chemical elements or a certain molecular weight range for the ligand. It also enables filtering using patterns that describe a local environment using the chemical substructure language SMARTS strings [19].

The central search element and origin of GeoMine’s name are geometry-based searches. To build the query, users may interactively select points in the web front end [21] (see Figure 1), utilizing an arbitrary PDB file as a template structure or define them without a template.

Users may select an arbitrary number of points, which can be filtered based on different properties. Moreover, the specification of distance ranges between two points and angles between specified distances is possible. Further, interactions between points, as stored in the interaction table, can be added to the query. Together they resemble an atomic substructure, which will be searched for. Each pocket can be examined individually as the interactions between one ligand and an individual pocket in a protein are of interest.

**Query Execution** The initial approach for query execution was first described for the predecessor tool *PELIKAN* by Inhester et al. [6]. The most significant enhancement for the runtime in developing the original GeoMine approach – utilizing a PostgreSQL database instead of SQLite – did not change the workflow of the searching process. The approach remained mostly



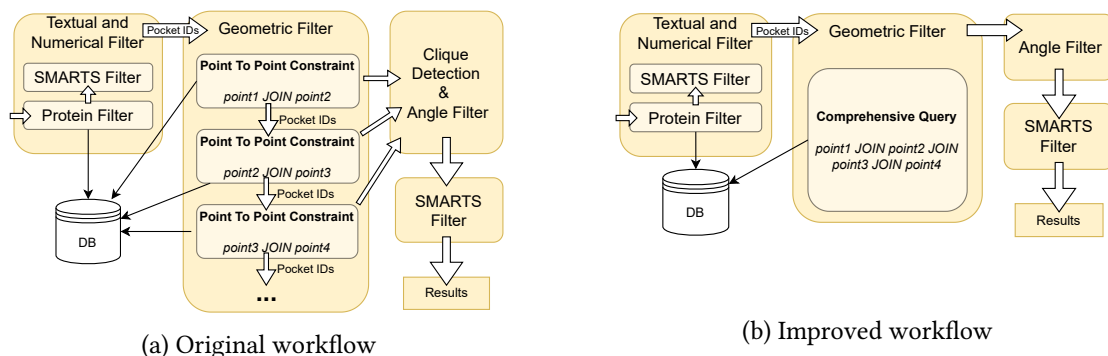
**Figure 1:** GeoMine’s three-dimensional view of a binding pocket based on the *NGL viewer* [20]. Users can interactively select atoms and other points and specify distances and interactions between them to generate the query. Here, a pocket around a ligand (bold bonds) is shown, together with the surrounding atoms of the protein.

algorithmic focused, with all major computational steps performed within the application (see Figure 2a), as the original PELIKAN software was designed to be a standalone desktop application. In the original approach of GeoMine [3], four major steps were performed strictly sequentially for each query to filter the potential results:

1. Textual and Numerical Constraints - A filter eliminates all proteins and pockets that do not meet specified properties or do not correspond to a given restrictive SMARTS filter. This step yields a list of all matching proteins and their pockets.
2. Obtaining all point pairs - For each point pair in the query, all possible results are returned, and distances, as well as interaction constraints, are checked.
3. Clique detection - An algorithm reconstructs the coherent component graph for all obtained point pairs and checks all defined angle constraints.
4. Less restrictive SMARTS filters for points were applied to the now-generated results.

Steps two and three of the query processing presented particular challenges. All point and point pair constraints were queried individually in the database. Since a single constraint for a point pair is often not very specific, it leads to big intermediate results. Only by chaining several constraints the number of points is sufficiently reduced. The need to cross-verify each point with all matching points in its pocket demanded significant computational resources, especially if the filter for the points were unspecific. The list of potential pockets needed to be recreated for each pair, as only pockets which contained results in prior pair subqueries remained in the search space. This caused the search to be strictly sequential and required the serialization and deserialization of long pocket-ID lists for the SQL WHERE clauses. As the application and database system are separate processes or running on separate servers, the required repeated transfer of these lists also affected the performance. Because some point-to-point constraints were specific (less frequent in the dataset) and others were unspecific (frequent in the dataset), a hand-crafted scoring function was utilized to estimate the best ordering of queries, starting





**Figure 2:** The original and the improved processing workflow of GeoMine (Simplified) for a given search

with the most specific queries to reduce the search space early [6]. Although this improved the join order in many cases, it had the disadvantage of preventing the database system from executing classical optimizations, such as parallelism and join order optimization.

Further, an additional algorithm was required since the results from the preceding steps consisted only of point pairs. The *Bron-Kerbosch algorithm* [22], a graph-based backtracking algorithm for clique detection, was used. This algorithm recursively verified whether all discovered point pairs constituted a complete graph and checked for angle constraints. This demanded substantial computational effort, taking several hours on large potential result sets.

### 3. Optimizations

This research aims to achieve optimal performance and ease of setup across various environments. Alongside the contributions of this work, the application has transitioned to a containerized setup for cloud environments. The optimizations presented in this work are essential for facilitating the deployment of a scalable application. In this section, we will distinguish between the *original approach* in GeoMine [3] and the *improved approach* we present in this work. The yielded results for each query remained identical.

#### 3.1. Optimizing SQL Queries

The most significant change from the original approach was the redesign of the SQL query generation. Sequential processing of each constraint within a query led to severely limited query-level parallelism and long processing times as described in Section 2.3. Therefore, all SQL queries are now designed to make use of PostgreSQL’s internal planning and optimization. In contrast to the original approach, where each point-to-point constraint was queried separately, a single comprehensive query containing all attributes and constraints for geometrical patterns is now constructed, see Figure 2b. This reduces overhead by eliminating the need to repeatedly serialize extensive lists of pocket IDs or create temporary tables. To achieve this, the point table joins itself as often as points were specified in the query, usually 5-15 times. As a match occurs inside a single pocket, we only need to join points within the same pocket. With information about the distribution of properties like the chemical element, the RDBMS can estimate which

part of the query restricts the search space the most and improve the join order. The original approach required running the checks on all points within all remaining pockets, not being able to skip points that were not matched in earlier subqueries. Intermediate results now remain within the database system and do not require serialization for application transfer. Additionally, merging all constraints (points, distances, and interactions) into one query eliminates the need for clique detection, as the output of the RDBMS is a connected and valid result.

Among all the geometric properties, only the angle checking between point pairs remains a separate step in the application, as this increases the complexity of the query without showing the benefits of an early reduced search space in our tests. Textual and numerical filters remain in a separate query to allow prior filtering, as SMARTS patterns require in-application processing. Allowing the RDBMS to determine the join order and the parallel execution resulted in a significant speedup of benchmark queries. The results are detailed in Section 4.

### 3.2. Enhanced Utilization of PostgreSQL Indexes

In the original approach, a single extensive index structure was created, covering 15 out of 17 table columns. Although PostgreSQL allows for the construction of multi-column indexes with a large number of attributes, these structures are only effective in certain situations due to their size and depending on the used attributes. However, using multiple single-column indexes and allowing PostgreSQL to combine them as recommended in the documentation [23] did not achieve the desired performance improvement.

Only the combination of several attributes could substantially reduce the number of yielded points. The best-found solution for our workload was a balanced compromise between index size and utilization, including only the most frequently used columns in a multi-column index. We identified two separate cases for index usage. Firstly, the earliest scheduled subquery focused solely on the attributes, disregarding their pocket, in cases without textual and numerical filters. Secondly, an index for subsequent subqueries was needed to filter for pocket IDs required for the join. In almost all instances, the optimizer determined to filter for the pocket ID in the second subquery. In some instances, a parallel index scan was performed. Filtering by the pocket ID reduced the search space best in these cases since the most restrictive subquery had already been executed as the first scheduled subquery. Therefore, we introduced a second index with the pocket identifier positioned first in the index. For both structures, we utilized PostgreSQL's default *B-Tree index* as other index structures seemed not beneficial in our tests. As pockets usually contain only a few hundred points, spatial indexes, like r-trees provided by PostGIS [14], did not provide the desired benefits. Filtering points and calculating all distances performed better in our tests than spatial operations due to the overhead of utilizing a spatial column. Index creation only needed a few minutes, but additional indexes for specific queries would no longer fit into the filesystem read cache and reduce performance.

### 3.3. Improving Text Search

The initial step of the workflow involves filtering structures based on textual and numerical attributes. These filters target various properties, the most important being the PDB identifiers used to select a pre-defined or user-defined subset of protein structures. A short alphanumeric

**Table 1**

Experiment overview. Showing enabled improvements between baseline *ex01* and all improvements *ex06*

Improvement	<b>ex01</b>	ex02	ex03	ex04	ex05	<b>ex06</b>	ex07	ex08	ex09	ex10
Index Improvement		x				x	x	x	x	
No Wildcards			x			x	x	x		x
New Query Design				x		x	x		x	x
No ILIKE					x	x		x	x	x

code identifies each structure.

Previously, an SQL *ILIKE* (case insensitive match) statement with a wildcard match at the beginning and end of the string was executed to check for the desired properties. For the PDB codes, we could make two changes. We could discard the wildcards in the query unless explicitly desired, which enables the utilization of a search index. And as the codes are not case-sensitive, we can replace the *ILIKE* with a *LIKE*, allowing for a case-sensitive search and resulting in a substantial speedup, as demonstrated in Section 4.

## 4. Evaluation and Discussion

### 4.1. Methods

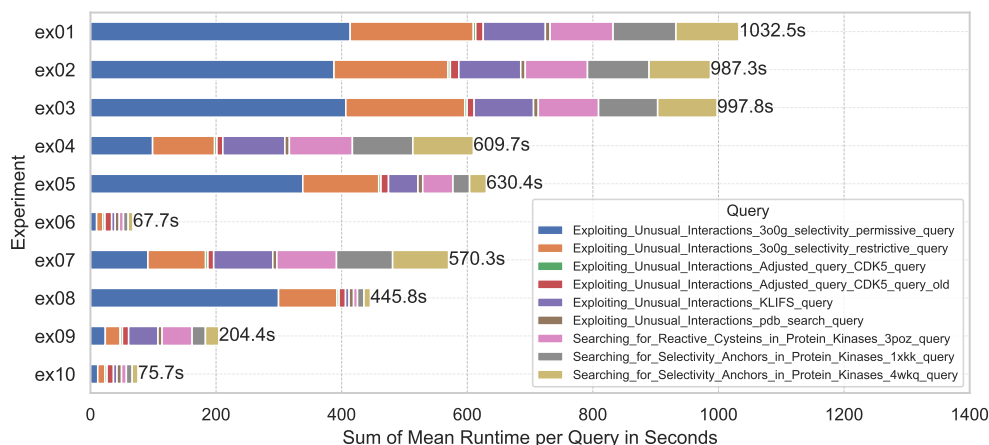
To evaluate the impact of each modification suggested for GeoMine, several experiments were derived from the original GeoMine approach *ex01* (see Table 1). Experiments *ex02* to *ex05* each contain only one of the improvements, *ex06* contains all improvements, while experiments *ex07* to *ex10* contain all except one. This way, we show which change impacts the performance most, as different improvements benefit from each other.

For evaluating the performance across different workloads, we used a set of nine queries already used in previous work [3], designed to highlight available features, show examples for common applications and estimate the runtime of different patterns common in GeoMine practical applications. They emitted between 2 and 7117 results.

We used a PostgreSQL15 database system. All data was stored on an SSD. Unless otherwise specified, a dedicated server with 400GiB RAM and 80 Cores was used (PostgreSQL 128GB *sharedbuffers*, 16 *parallel workers*). Podman [24] was used to deploy the system. Each experiment was repeated five times. The GeoMine application was executed on the same node as the PostgreSQL database. We configured PostgreSQL to utilize less memory than available, as GeoMine required a high amount of working memory for some workloads. Additionally, we conducted tests on commodity systems by employing two setups (*small/medium*) using virtual servers. Both setups stored data on SSDs and were equipped with 12 cores and 24GB RAM, resp. 18 cores and 48GB RAM.

### 4.2. Results

Figure 3 shows the mean runtime of the nine test queries for each experiment as depicted in Table 1. Each change led to better performance, with the highest performance gain occurring



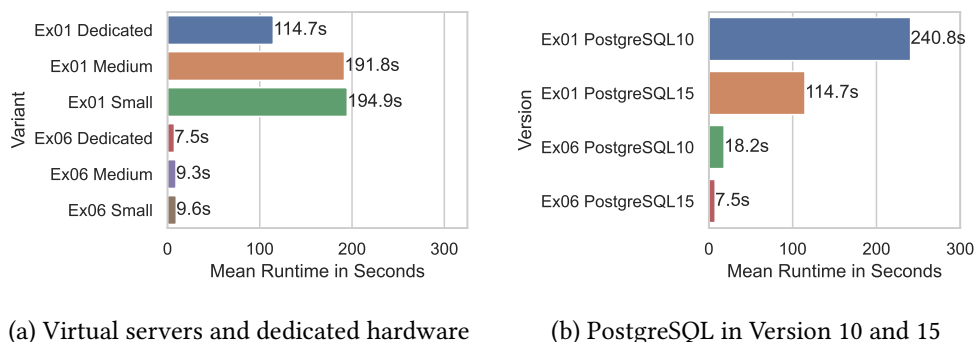
**Figure 3:** The sum of mean runtimes in seconds for each experiment as described in Section 4.1. Each color represents one distinct query

when all changes were applied together. The required time for performing all nine queries decreased from 1033sec of the original approach (*ex01*) to 68sec with all improvements (*ex06*).

The new query design (*ex04*) had the most substantial impact on performance, particularly visible in the long-running queries. Also, the transition from the ILIKE to the LIKE statement notably reduced runtime. The performance gain is most noticeable on the medium-running queries containing a long list of PDB IDs for a preselection. The experiments 02 and 03, the *new index* and *no wildcards* in the PDB ID selection showed only a small improvement. However, experiment 09, which contains all changes except the wildcard improvement, shows that it has an impact on the overall runtime, presumably benefiting from the switch to the LIKE statement. The changes in index structures showed less impact than expected, demonstrating that PostgreSQL can handle indexes with an inflated number of columns. However, the performance was drastically worse if no index was used or index structures did not combine multiple attributes. For instance, combining one index per attribute led to an increase of the sum of the mean runtimes from 68sec (*ex06*) to 134sec.

**Unspecific Queries** Some of the used queries include a protein filter to reduce the number of searched pockets. When removing these filters and searching the whole dataset, the original approach reached its set limits (needing more than 100GB RAM or 1h time) on some of these and other queries with less restrictive geometric filters. With the improved approach, some queries with extensive intermediate results could now be computed for the first time, often within minutes.

**Alternative Setups** As large database instances are not always accessible, for example, due to cost constraints in cloud environments, we also conducted our experiment on two smaller virtual servers. As shown in Figure 4a, the performance gains were also visible on these smaller server instances. These tests were performed on shared hardware, so they can only show a general trend rather than precise comparative data. However, they demonstrate the feasibility



**Figure 4:** Mean runtimes in alternative configurations of experiment 01 and 06

of processing on shared virtual servers. Additionally, we observed a substantial speedup while transitioning from PostgreSQL10 to PostgreSQL15 as displayed in Figure 4b. Combined with our improvements, we achieved a speedup factor of 32.

## 5. Conclusion and Future Work

GeoMine is a unique application for geometric searches in large collections of protein-ligand complexes with high relevance for life-science research. We showed that it was possible to achieve a large speedup on our query processing by moving major parts of the processing from a custom-written logic inside the software to a PostgreSQL database system. Additionally, different approaches in database optimization contributed to further performance gain. Overall, these achievements are critical for the practical use of the system handling the growing dataset. Some queries could be executed for the first time on our setup due to these changes. In this work, we focused on optimizations of the database and query design. We demonstrated the substantial benefits of database optimizations in scientific applications, achieving a fifteen-fold speedup in GeoMine. Coupled with a halving of the runtime through the use of a newer PostgreSQL version, we managed to reduce the average runtime from minutes to seconds.

Looking ahead, we plan to explore additional database paradigms, such as distributed or column-based systems, and establish schema changes for further optimizations. The caching of intermediate results, as well as determining the join order by extended statistics or by utilizing machine learning, may potentially provide additional benefits. This way, we aim to achieve even better performance for searching scientific data with a service-oriented web service.

## Acknowledgments

This work was supported by the German Federal Ministry of Education and Research as part of CompLS and de.NBI (031L0172 and 031L0105).

## References

- [1] T. Inhester, M. Rarey, Protein–ligand interaction databases: advanced tools to mine activity data and interactions on a structural level, *WIREs Computational Molecular Science* 4 (2014) 562–575. doi:10.1002/wcms.1192.
- [2] S. K. Burley, C. Bhikadiya, C. Bi, S. Bittrich, H. Chao, L. Chen, e. a. Craig, RCSB Protein Data Bank (RCSB.org): delivery of experimentally-determined PDB structures alongside one million computed structure models of proteins from artificial intelligence/machine learning, *Nucleic Acids Research* 51 (2022).
- [3] J. Graef, C. Ehrt, K. Diedrich, M. Poppinga, N. Ritter, M. Rarey, Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures, *Journal of Medicinal Chemistry* 65 (2022) 1384–1395. doi:10.1021/acs.jmedchem.1c01046.
- [4] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al., Highly accurate protein structure prediction with alphafold, *Nature* 596 (2021) 583–589. doi:10.1038/s41586-021-03819-2.
- [5] K. Schöning-Stierand, K. Diedrich, R. Fährrolfes, F. Flachsenberg, A. Meyder, E. Nittinger, R. Steinegger, M. Rarey, Proteins plus: interactive analysis of protein–ligand binding interfaces, *Nucleic acids research* 48 (2020) W48–W53. doi:10.1093/nar/gkaa235.
- [6] T. Inhester, Mining of Interaction Geometries in Collections of Protein Structures, Ph.D. thesis, Universität Hamburg, 2017.
- [7] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne, The Protein Data Bank, *Nucleic Acids Research* 28 (2000) 235–242. doi:10.1093/nar/28.1.235.
- [8] C. Tenopir, N. M. Rice, S. Allard, L. Baird, J. Borycz, L. Christian, B. Grant, R. Olendorf, R. J. Sandusky, Data sharing, management, use, and reuse: Practices and perceptions of scientists worldwide, *PloS one* 15 (2020) e0229003.
- [9] M. Raasveldt, H. Mühleisen, Data management for data science-towards embedded analytics., in: *CIDR*, 2020.
- [10] R. D. Hipp, SQLite, 2020. URL: <https://www.sqlite.org/>.
- [11] The PostgreSQL Global Development Group, PostgreSQL: The world’s most advanced open source relational database, 2023. URL: <https://www.postgresql.org>.
- [12] solid IT gmbh, Db-engines ranking, 2023. URL: <https://db-engines.com/en/ranking>.
- [13] A. Conrad, Database of the year: Postgres, *IEEE Software* 38 (2021) 130–132. doi:10.1109/MS.2021.3089730.
- [14] The PostGIS Development Group, Postgis, 2023. URL: <https://postgis.net>.
- [15] U. Cubukcu, O. Erdogan, S. Pathak, S. Sannakkayala, M. Slot, Citus: Distributed postgresql for data-intensive applications, in: *Proceedings of the 2021 International Conference on Management of Data, SIGMOD ’21*, 2021, p. 2490–2502. doi:10.1145/3448016.3457551.
- [16] S. Urbaczek, A. Kolodzik, J. R. Fischer, T. Lippert, S. Heuser, I. Groth, T. Schulz-Gasch, M. Rarey, Naomi: On the almost trivial task of reading molecules from different file formats, *Journal of Chemical Information and Modeling* 51 (2011) 3199–3207. doi:10.1021/ci200324e.
- [17] J. Graef, C. Ehrt, M. Rarey, Binding site detection remastered: Enabling fast, robust,

- and reliable binding site detection and descriptor calculation with dogsite3, *Journal of Chemical Information and Modeling* 63 (2023) 3128–3137. doi:10.1021/acs.jcim.3c00336, PMID: 37130052.
- [18] T. Inhester, S. Bietz, M. Hilbig, R. Schmidt, M. Rarey, Index-based searching of interaction patterns in large collections of protein–ligand interfaces, *Journal of Chemical Information and Modeling* 57 (2017) 148–158.
- [19] I. Daylight Chemical Information Systems, Smarts-a language for describing molecular patterns, 2007.
- [20] A. S. Rose, A. R. Bradley, Y. Valasatava, J. M. Duarte, A. Prlić, P. W. Rose, NGL viewer: web-based molecular graphics for large complexes, *Bioinformatics* 34 (2018) 3755–3758. doi:10.1093/bioinformatics/bty419.
- [21] K. Diedrich, J. Graef, K. Schöning-Stierand, M. Rarey, GeoMine: interactive pattern mining of protein–ligand interfaces in the Protein Data Bank, *Bioinformatics* 37 (2020) 424–425. doi:10.1093/bioinformatics/btaa693.
- [22] C. Bron, J. Kerbosch, Algorithm 457: finding all cliques of an undirected graph, *Communications of the ACM* 16 (1973) 575–577.
- [23] PostgreSQL 15, Documentation, 2023. URL: <https://www.postgresql.org/docs/15/>.
- [24] Containers, podman, 2023. URL: <https://podman.io/>.





## F.6 Database-Driven Identification of Structurally Similar Protein-Protein Interfaces

- [D6] **Graef, J.**, Ehrt, C., Reim, T., Rarey, M., Database-Driven Identification of Structurally Similar Protein-Protein Interfaces. *J. Chem. Inf. Model.* (2024). Akzeptiert.

Reprinted with permission from [D6]. Copyright 2024 American Chemical Society.

# Database-Driven Identification of Structurally Similar Protein-Protein Interfaces

Joel Graef, Christiane Ehrh,\* Thorben Reim, and Matthias Rarey\*



Cite This: <https://doi.org/10.1021/acs.jcim.3c01462>



Read Online

ACCESS |



Metrics & More

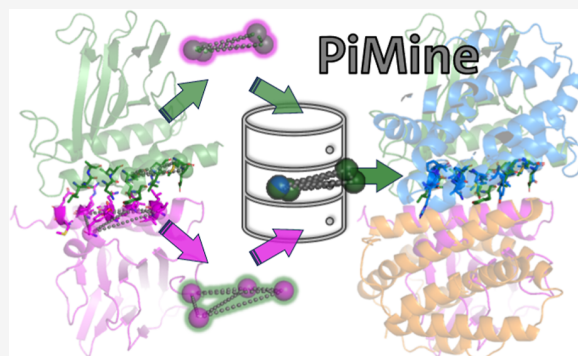


Article Recommendations



Supporting Information

**ABSTRACT:** Analyzing the similarity of protein interfaces in protein-protein interactions gives new insights into protein function and assists in discovering new drugs. Usually, tools that assess the similarity focus on the interactions between two protein interfaces, while sometimes we only have one predicted interface. Herein, we present PiMine, a database-driven protein interface similarity search. It compares interface residues of one or two interacting chains by calculating and searching tetrahedral geometric patterns of  $\alpha$ -carbon atoms and calculating physicochemical and shape-based similarity. On a dedicated, tailor-made dataset, we show that PiMine outperforms commonly used comparison tools in terms of early enrichment when considering interfaces of sequentially and structurally unrelated proteins. In an application example, we demonstrate its usability for protein interaction partner prediction by comparing predicted interfaces to known protein-protein interfaces.



## INTRODUCTION

Protein-protein interactions (PPIs) are integral for many cellular processes and are associated with various diseases.<sup>1</sup> Targeting PPIs represents a major challenge with high potential for drug discovery.<sup>2</sup> An interesting illustration of how known protein-protein complexes can be used for the rational design of drugs is Venetoclax, which was approved for the therapy of chronic lymphocytic leukemia in 2016.<sup>3</sup> Its predecessor molecules, ABT-737<sup>4</sup> and ABT-263 (Navitoclax),<sup>5</sup> were designed based on a known complex structure of its antiapoptotic target BCL-X<sub>L</sub> and a peptide derived from the pro-apoptotic protein Bak. The investigation of the structures of these inhibitors in complex with BCL-2 and a re-engineering strategy led to the development of Venetoclax,<sup>6</sup> impressively illustrating the impact of our knowledge of PPIs for drug design. However, in contrast to the exploitation of the comparatively small protein binding pockets, there are still few computational methods for targeting protein-protein interfaces, which are called interfaces for simplicity. This lack of in silico tools can be attributed to multiple reasons, such as interfaces being large with 1500–3000 Å<sup>2</sup>, very hydrophobic, and flat without deep cavities. Therefore, designing small-molecule binders is difficult.<sup>7</sup> Although permanent and transient complexes differ in these properties,<sup>8</sup> both types of PPIs could be successfully addressed by small molecule binders in the past.<sup>9</sup> Therefore, we do not distinguish between these PPI types, which are both challenging to address. Nevertheless, the analysis of interfaces and the interactions can lead to insights into, e.g., the affinity of interactions, the biological function, or potential side effects of PPI-based drugs due to

interface similarities. Although we have abundant information on biologically relevant PPIs based on experimental data,<sup>10</sup> their protein-protein complex structure is often unknown. Moreover, proteins of unknown function might harbor interfaces for PPI, whose knowledge might help unravel their biological impact.<sup>11</sup> Also, known small molecule inhibitors of similar PPIs can be explored by researchers to study their interfaces.<sup>12</sup> Comparing interfaces and looking for similarities is one approach to broadening our structural knowledge of interfaces by analyzing protein sequences and their structures. While numerous sequences are known, the generated alignments of sequence-dependent methods always obey the sequence order. As many functionally similar binding interfaces are sequence-independent,<sup>13</sup> the latter methods fail to detect their similarities. Also, chains with similar interfaces might be structurally close but sequentially remote. In these cases, structure-dependent methods are the solution of choice, provided a protein structure model is available. The prediction of interfaces and a comparison to already known biologically relevant interfaces can help in understanding the structural details of PPIs. Therefore, interface comparison methods should enable users to screen databases based on predicted

**Received:** September 11, 2023

**Revised:** February 26, 2024

**Accepted:** February 26, 2024

Table 1. Exemplary Selection of Tools for Protein-Protein Interface Similarity Calculations

method	availability	citation count	algorithmic approach
I2I-SiteEngine <sup>16</sup>	not available	21 (92 <sup>a</sup> )	interface definition: 4 Å; representation of interface as surface points that describe physicochemical properties and local surface curvature; triangles of surface points are hashed and used for matching
CMAPi <sup>22</sup>	not available	22	interface definition: 10 Å; interfaces as contact map matrices of residues; uses 2D dynamic programming to optimize alignment score using the Smith-Waterman <sup>23</sup> algorithm
iAlign <sup>18</sup>	standalone	75	interface definition: 4.5 Å; three initial alignments are calculated using gapless threading, secondary structure, and fragment assembly; alignments are refined using dynamic programming
PCalign <sup>24</sup>	standalone	15	interface definition: 4.5 Å; $C\alpha$ atoms are applied to geometric hashing where the $C\alpha$ atoms are assigned with chemical types of the residue and an alignment of those point clouds is searched
PROSTA-inter <sup>25</sup>	standalone <sup>b</sup>	15	interface definition: 6 Å; selects $C\alpha$ atoms and calculates alignments based on local and remote fragments; alignments are clustered and further refined for final results; also supports nucleic acids ( $C3'$ atoms)
InterComp <sup>26</sup>	standalone	13	interface definition: 5 Å; represents interface residues as points in space using $C\alpha$ atoms; uses simulated annealing and compares distance maps; sequence-dependent
PatchBag <sup>27</sup>	standalone	3	bag-of-words approach which represents the protein surface or interfaces as vectors of counts of geometrical types of surface patches; patches are defined for a residue by its $C\alpha$ atom and 4 neighboring $C\alpha$ atoms

<sup>a</sup>SiteEngine<sup>17</sup> citation count. <sup>b</sup>Web service not available anymore.

interfaces for single-chain protein structures.<sup>14</sup> This possibility is often not implemented in current comparison approaches. Furthermore, programs for similarity assessment differ in the interface region definition, the algorithm, the applied scoring functions, and the datasets on which the methods are parametrized and tested.

Tools for structure-based calculation of protein-protein interface similarities are usually developed as standalone programs, performing pairwise comparisons. In contrast, structurally comparing one interface to, e.g., the complete Protein Data Bank (PDB),<sup>15</sup> is considerably more time-consuming than sequence-based approaches. To our knowledge, the tool I2I-SiteEngine<sup>16,17</sup> was the only protein-protein interface similarity search still available online as a web server. However, the analysis was restricted to pairwise comparisons. Unfortunately, the standalone tool and web server are no longer accessible at the time of submission.

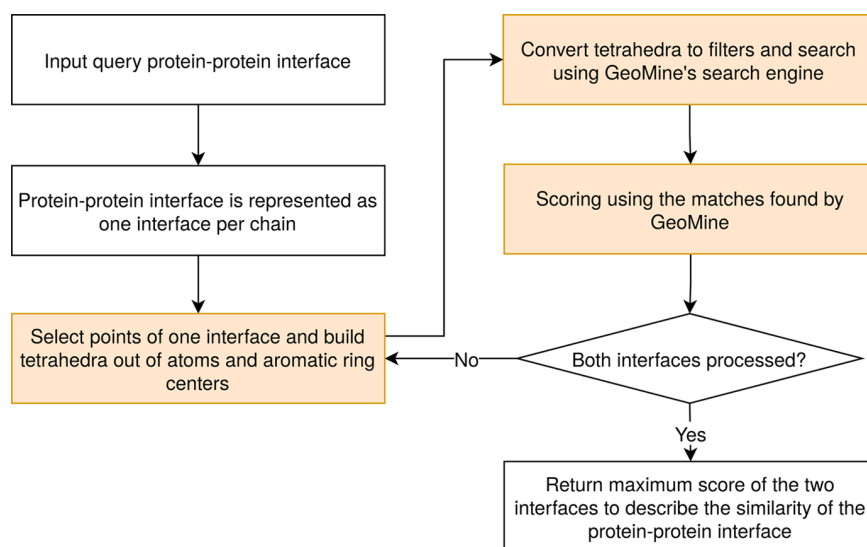
Table 1 gives an exemplary selection of protein-protein interface similarity calculation methods. iAlign<sup>18</sup> shares a similar concept to the protein structure alignment method TM-align.<sup>19</sup> It defines protein-protein interfaces by the heavy atom distances between protein residues. Thus, a residue is part of an interface if at least one of its heavy atoms is within a maximum distance of 4.5 Å from any heavy atom of the other protein chain. Nevertheless, the interface comparisons use complete protein sequences. Two scoring functions are available: the TM-score and the IS-score. Both estimate the similarity based on the  $C\alpha$  atoms of the aligned residue pairs. In addition to the pairwise distances, the IS-score includes a so-called contact overlap factor. It describes the conservation of interfacial contact patterns. The alignment algorithm calculates three initial alignments: gapless threading, secondary structure, and fragment assembly alignments. These alignments result in a scoring matrix for iterative refinements. The algorithm stops after 30 iterations or when the alignment converges. A drawback of this method is that the alignments are still-dependent on the quality of the sequence alignments, as it relies on gapless alignments in the initial steps.

I2I-SiteEngine<sup>16</sup> defines interfaces in the same way as iAlign, but with a maximum distance of 4 Å. The interfaces are described by surface points and are annotated by the physicochemical properties of the functional groups of the residues. Surface points are then grouped into surface patches. In addition, a shape function based on solid angles describes the average curvature of each surface patch.<sup>16,20,21</sup> Then, the

centers of the patches are combined as multiple triangles and retained if they have complementary physicochemical properties to a patch in the second protein interface, with which they build up the protein-protein interface. The triangles are hashed and can be searched for in other protein complexes. Two scoring functions that use the physicochemical properties and the solid angles assess the interface similarity. While the initial score is applied to a low-resolution representation to reduce the search space, the second score uses a higher resolution level. The match list is then enumerated by calculating the maximum weight match using a bipartite graph. In this process, two so-called 1:1 correspondence scores are calculated. Finally, all scores are summed to get the total similarity score. The alignments of I2I-SiteEngine are fully sequence-independent, but the method has a much longer runtime than iAlign.<sup>18</sup>

In this study, we present PiMine, a tool for the alignment and similarity assessment of protein-protein interfaces based on structural features. Its algorithm assumes that if one protein chain or interface of a PPI is similar to one of another PPI, both interfaces are similar. In contrast to other interface-comparison tools, PiMine calculates the similarity for interfaces between two interacting protein chains as well as for one-sided interface regions, as defined by a user or predicted by a third-party program. The latter is a major benefit compared to other interface comparison methods. PiMine reports three similarity scores based on the shape, the physicochemical properties, and a combination thereof. In addition, the tool provides the aligned protein structures in the PDB format.

Here, we benchmark PiMine against the currently most cited ones for interface comparison based on both existing well-known datasets and newly developed datasets to evaluate screening performance, alignment quality, and runtime. Given the high early enrichment of PiMine for state-of-the-art datasets, we could confirm that our assumption that the similarity between two protein chains often also reflects the similarity of the interfaces of the two interacting chains holds for known similar interface pairs. Based on a novel, highly unbiased benchmark dataset, we demonstrate that PiMine performs superior to frequently cited tools in correctly detecting remote similarities between sequentially unrelated interfaces. Moreover, we show that a comparison based on the interfaces of single chains with PiMine is robust and leads to a convincing early enrichment of similar interface pairs. Finally, we illustrate how PiMine performs well for known application examples of interface comparison methods and how it can be



**Figure 1.** Overview of the PiMine algorithm steps illustrated as boxes. The TetraScan algorithm is highlighted in orange.

applied to identify novel potential interaction partners of protein chains.

## METHODS

**Interface Modeling.** PiMine, like many other methods, determines the interface regions of PPIs by the proximity of protein chains. Protein heavy atoms within 4.5 Å of any heavy atom of the interacting chain constitute the interface.

**Alignment Algorithm.** We developed the new algorithmic concept of TetraScan for binding site<sup>28</sup> and protein-protein interface comparisons. Here, we describe the method in the context of interface similarity. TetraScan creates database queries for protein site atoms in the form of tetrahedrons. It uses a database based on the GeoMine technology<sup>29</sup> comprising all interface atoms of protein–ligand or protein-protein complexes to compare. During database creation, reasonable protonation states<sup>30</sup> are determined, hydrogen atom positions are predicted, and the interface atoms are stored along with their spatial coordinates and physicochemical properties.

Figure 1 provides an overview of the search algorithm. It starts using a PiMine database created for complex structures in the PDB or mmCIF file format and a query protein-protein interface whose similarity to other protein-protein interfaces should be assessed. This query protein-protein interface consists of a subset of protein atoms of two nearby chains or predefined interface atoms of a single chain. The interface specification options are described in more detail in the [Supporting Information](#) (see Paragraph S1).

The interface atoms can optionally be reduced to only the C $\alpha$  atoms of all residues (default) or to so-called “restricted” points, which are all heteroatoms (oxygen, nitrogen, and sulfur), all aromatic ring centers (His, Phe, Trp, and Tyr), and all hydrophobic side chain carbon atoms of Ala, Ile, Leu, Lys, Met, Pro, and Val. When using the restricted points or all points, the interface atoms are reduced to solvent-exposed atoms of the unbound structure. Aromatic ring centers are used irrespective of the solvent exposure of the corresponding ring atoms. Then, tetrahedra are generated for all interface atoms of a single chain. The distance between two atoms or corners of a tetrahedron must be in a specified range,

optionally defined by the user (default:  $minDist = 1 \text{ \AA}$ ,  $maxDist = 14 \text{ \AA}$ ). The set of generated tetrahedra is sorted in descending order according to the sum of their edge lengths. From this set, the tetrahedrons are selected. The selection ensures that each interface atom is part of at least one tetrahedron. If all interface atoms are represented and more tetrahedrons than a defined value (default:  $noffilters = 30$ ) are calculated, only the  $noffilters$  largest tetrahedrons are selected. The resulting tetrahedra are converted to filters for querying the database with PiMine. All edges are annotated by the distances between the atoms or aromatic centers and an adjustable tolerance (default:  $distTolerance = 1.0 \text{ \AA}$ ). For the atoms, different properties such as residue name, residue class, backbone/side chain, solvent exposure, aromatic center, chemical element, and atom interaction type are annotated if available. The properties used in the search depend on the selected points and the chosen hierarchy level. Four hierarchy levels are available: (1) atom interaction type only, (2) chemical element and atom interaction type, (3) residue type, element, and backbone/side chain, and (4) residue name, chemical element, and backbone/side chain. By default, PiMine uses the second hierarchy level and handles C $\alpha$  atoms as a unique atom type differentiated from other carbons (adjustable using the parameter  $filterHierarchyLevel$ ).

The search returns a series of atom mappings for the query tetrahedrons. One hit represents a tetrahedron pattern from the query interface also occurring in the target interface, where all hierarchy-specific properties of the atoms and defined distance criteria match. The respective tetrahedron filter can match several times in the same target interface, matching even the same atoms. Symmetrical atom mappings are not filtered because they lead to different alignments and therefore different scores. For each hit, a superposition of the query and the match can be calculated with the Kabsch–Umeyama algorithm.<sup>31–33</sup> We assess whether a hit provides a good superposition and thus has a high degree of similarity in two steps. First, a prefiltering step checks the shape score based on the C $\alpha$  atoms and a radius of 6 Å. This radius definition represents the average residue diameter of about 10.6 Å<sup>34</sup> and a small tolerance of 1.4 Å, leading to 12 Å. Second, the best  $x$  hits per matching target interface are selected, where  $x$  is

C

Table 2. Scoring Schema for Two Atoms Regarding Their Pharmacophore-Based Similarity<sup>a</sup>

	Acc/Don	Acc	Don	Aro	HyPhob	Ca	Pos/Don	Neg/Acc
Acc/Don	1	0.6	0.6	0	0	0	0.6	0.6
Acc		1	0	0	0	0	0	0.8
Don			1	0	0	0	0.8	0
Aro				1	0.8	0	0	0
HyPhob					1	0	0	0
Ca						1	0	0
Pos/Don							1	0
Neg/Acc								1

<sup>a</sup>Acc/Don: hydrogen bond acceptor and donor, Acc: hydrogen bond acceptor, Don: hydrogen bond donor, Aro: aromatic ring atom, HyPhob: hydrophobic atom, Ca:  $\alpha$ -carbon atom, Pos/Don: hydrogen bond donor and positively charged, and Neg/Acc: hydrogen bond acceptor and negatively charged.

defined as the square root of the number of total hits in the respective target interface. After this prefiltering step, the similarity scores for all remaining hits are calculated. These atom-wise scores are considering all neighboring atoms in a predefined radius (default:  $scoringRadius = 1.5 \text{ \AA}$ ). For each target interface, the highest-scoring alignment is selected at the end.

Since the similarity determination may depend on the particular protein chain under consideration, the query interface should be chosen appropriately. A PPI is defined by one interface per protein chain. Thus, there are four possible chain pairs for the similarity calculation. For the query interface chains A and B and the target interface chains A' and B', four chain-based alignments are possible: (1) A versus A', (2) A versus B', (3) B versus A', and (4) B versus B'. Therefore, we calculate all four alignments and select the highest-scoring alignment for transforming the target match to the query structure. Note that, by default, we report the maximum of the four calculated interface similarity scores. Optionally, PiMine can also consider the similarities between both interfaces if the parameter *twoSidedScoring* is enabled.

Finally, the matches are returned with their corresponding scores and the transformed target interfaces in the PDB format for ranking, visualizing, and investigating the detected similarity if required.

**Similarity Measure.** There are three similarity scores in PiMine: shape-based (shape score), pharmacophore-based (pharma score), and their equally weighted sum called the SP-score. The scores are calculated for all solvent-exposed points using a linear search or the nanoflann implementation of the  $k$ -dimensional tree,<sup>35</sup> depending on how large the interface is, to reduce runtime. A linear search will be performed if there are fewer than 100 solvent-exposed interface points. For each query interface point, the closest point of the matching target interface is searched within a defined scoring radius. If at least one point is found, the counter for the shape score is incremented by one, and the pharma score is incremented based on pharmacophore properties using a knowledge-driven scoring matrix (Table 2). After all query interface points have been processed, both scores are normalized by the number of surface points of the larger single-chain interface, providing the shape and pharmacophore-based score for each hit. At the end of this process, the best alignments are selected based on the highest SP-score of all hits for the target interface.

**Datasets.** We used five datasets in our experiments (Table 3).

The first dataset, the *ParamOptSet*, encompasses complex structures for the scoring quality assessment in protein-protein

docking studies<sup>36</sup> to classify correctly and incorrectly predicted protein-protein complex structures. We downloaded the complete set of predicted complex structures<sup>37</sup> and preprocessed the datasets as follows: the PDB structures of the native complexes were downloaded and used as a query. Matches with all predicted complexes of high- and medium-quality ( $(0.3 \leq f_{nat} < 0.5)$  and  $(LRMSD \leq 5.0 \text{ \AA}$  or  $IRMSD \leq 2 \text{ \AA})$  or ( $f_{nat} \geq 0.5)$  and  $(LRMSD > 1.0 \text{ \AA}$  and  $IRMSD > 1.0 \text{ \AA})$ ) were handled as similar interfaces (actives). Their scores should be higher than those of matches with predicted complexes of acceptable quality or incorrectly predicted complexes (inactives). Here, LRMSD (originally *L\_rms*) denotes the root-mean-square deviation (*rmsd*) of the smaller protein compared with the native pose, IRMSD (originally *I\_rms*) is the backbone *rmsd* of the interface residues, and  $f_{nat}$  is "defined as the number of native (correct) residue-residue contacts in the predicted complex divided by the number of contacts in the target complex".<sup>38</sup> To ensure a more realistic ratio of inactives to actives, we used only structures with a ratio of at least 20 (inactives/actives). Furthermore, we excluded complexes with interfaces between more than two chains. The final dataset contains predicted structures for 18 native protein-protein complexes (18 groups, 5678 structures) for parameter optimization.

Details regarding the second (*DimerS97*)<sup>18</sup> and third (*Keskin*)<sup>39</sup> sets, two published datasets from the literature, can be found in Paragraph S2 in the [Supporting Information](#). Pairwise comparisons of the chains constituting the datasets' interfaces were performed with TM-align.<sup>18</sup>

To test whether PiMine detects similarities between single-chain interfaces of sequentially and structurally dissimilar chains, we designed another dataset called *PiMineSet*. We downloaded all structures from the PDB as of March 15th, 2022. Next, we applied the standalone version of EPPIC<sup>43</sup> to find biological interfaces in all asymmetric units of these structures, as downloaded from <https://github.com/eppic-team/eppic> on September 9th, 2021. We retained only PDB structures with a resolution of at most 2  $\text{\AA}$  and a free *R*-factor of at most 0.25 from this set of biological interfaces and ignored structures without these annotations. The remaining structures were sequence-culled using Linclust<sup>44</sup> in the slower but more sensitive "cluster" mode, with a minimum sequence identity of 25%. Apart from that, default settings were used. Next, we compared the sequence-culled protein chains against all protein chains of biological interfaces in our PDB subset using Foldseek<sup>45</sup> with default settings and a TM-score threshold of 0. To find interfaces with two structurally related chains and two structurally unrelated chains as similar pairs of

Table 3. Overview of the Composition and Purpose of the Five Datasets Used in This Publication<sup>a</sup>

name	biological interfaces	similarity criterion	#Act	#Inact	date	purpose
<i>ParamOpISet</i>	manually curated <sup>40</sup>	similarity to native protein-protein complex; <sup>36</sup> actives: high- and medium-quality complexes; inactives: acceptable and incorrect complexes	129	5549	2015	parameter optimization
<i>Dimer597</i> <sup>18</sup>	interfacial energy according to Lu et al. <sup>41</sup> below -12 <sup>42</sup>	same SCOP assignment of at least one structural similarity according to TM-align <sup>18</sup> ; minimum contact overlap ratio of 0.3	373	176,875	2010	enrichment assessment (sequentially and structurally related chains)
<i>Keskin</i> <sup>39</sup>	≥10 interface residues	geometric similarity of the position of C $\alpha$ atoms; percent residue identity in the match; size similarity of the interfaces	4876	176,627	2004	enrichment assessment (structurally related interfaces)
<i>PiMineSet</i>	EPPIC <sup>43</sup> predictions for ASU	sequential and structural similarity between two chains of both interfaces; sequential and structural dissimilarity between the other two chains of both interfaces; residue overlap of 0.6 and 0.8 of the aligned interface chains	77	2718	2022	enrichment and alignment assessment
<i>RunTimeSet</i>	EPPIC <sup>43</sup> predictions for ASU	n/a	n/a	n/a	2022	runtime analyses and applications

<sup>a</sup>ASU, asymmetric unit; #Act, number of similar interfaces; #Inact, number of dissimilar interfaces.

our new dataset, we checked each interface chain in the sequence-culled set of biological interfaces for its similarities to other biologically relevant interfaces. An interface pair was retained for further analyses if it fulfilled all the following conditions: (1) one chain of the query interface had a TM-score of at least 0.5 to another chain of the target interface, (2) this chain had a TM-score below 0.5 to the other chain of the target interface, and (3) the other chains of both interfaces had a TM-score below 0.5. These preselected pairs of interfaces were processed with UCSF Chimera.<sup>46</sup> Both proteins were loaded, and the chains with a TM-score of at least 0.5 were aligned using MatchMaker.<sup>47</sup> All residues of the similar query chain within a 4 Å environment of the corresponding partner chains were selected to determine the percentage of overlapping residues of the query chain in both interfaces relative to all interface residues of the interfaces of both complexes (residue overlap). We visually inspected all interfaces with an overlap (residue intersection) of at least 60% relative to the number of residues of one interface and 80% relative to the number of residues of the other one to extract interesting similar interface pairs. Also, a more detailed analysis with TM-align<sup>19</sup> was performed to compare all interface chains. The TM-score for two chains of each pair had to be higher than 0.75, while the score for the other two chains of the interfaces had to be below 0.5. Interface pairs fulfilling the TM-score criteria applied for the Foldseek analysis with a relative overlap of at most 5% constitute the dissimilar interfaces (inactives) in this dataset. The TM-scores and relative residue overlaps of the finally chosen similar and dissimilar interface pairs can be found in the corresponding repository.<sup>48</sup> Figure S1 in the Supporting Information shows exemplary similar and dissimilar interfaces in this dataset. An additional benefit of this dataset is the availability of the corresponding alignments. Therefore, we also used this set to benchmark the alignment performance of PiMine compared with other commonly used protein-protein complex alignment methods.

To analyze the runtime, we built a fifth dataset, named *RunTimeSet*, by applying the EPPIC software<sup>43</sup> on all protein structures in the PDB on October 28th, 2022, to predict biological protein-protein interfaces in the asymmetric unit of the PDB entries. This dataset contains 169,944 interfaces in 59,928 structures, which is suitable for screening with known and predicted protein-protein interfaces, e.g., to identify potential protein interaction partners. For the run-time analyses, we compared the interface between chains A and B of the randomly selected PDB entry 3t4m against all dataset interfaces.

**Usage of External Tools.** I2I-SiteEngine was downloaded (<http://bioinfo3d.cs.tau.ac.il/cgi-bin/pdownload/progdownload.pl/?pname=I2ISiteEngine>, last access: September 27th, 2022) and installed using the Perl script "install\_I2ISiteEngine.pl". I2I-SiteEngine was run with the default parameters. From all calculated scores (low-resolution score, overall surface score, 1:1 correspondence curvature and distance score, and total score), we assessed the performance using the highest total score. The executable "pdb\_trans\_all\_atoms.Linux" for applying the transformation matrices of the alignments to the target structures did not work. Therefore, we applied the reported transformation matrix to the target PDB structures using the tool *pdbsset* of the CCP4 Software Suite<sup>49</sup> (version 7.1). For the screening dataset, 169,769 of 169,944 interfaces could be correctly prepared (99.9%). Subsequently, 169,714 interfaces were compared, while only 164,321

E

<https://doi.org/10.1021/acs.jcim.3c01462>  
J. Chem. Inf. Model. XXXX, XXX, XXX–XXX

**Table 4.** First Eight Results of the Optimization of the Maximum Distance (*maxDist*), the Minimum Distance (*minDist*), the Distance Tolerance (*distTolerance*), and the Scoring Radius (*scoringRadius*)<sup>a</sup>

entry	<i>minDist</i>	<i>maxDist</i>	<i>scoringRadius</i>	<i>distTolerance</i>	EF (1%, 2%, 5%, 10%, 20%)	AUC	time [s]
1	3	14	1.5	1.5	10.22, 6.62, 3.73, 2.64, 2.17	0.702	8825
2	2	14	1.5	1.5	10.22, 6.62, 3.73, 2.64, 2.17	0.701	5016
3	1	14	1.5	1.5	10.22, 6.62, 3.73, 2.64, 2.17	0.701	5589
4	1.5	14	1.5	1.5	10.22, 6.62, 3.73, 2.64, 2.17	0.701	5651
5	2.5	14	1.5	1.5	10.22, 6.62, 3.73, 2.64, 2.17	0.701	8860
6	3	14	1.5	1.0	10.22, 6.62, 3.73, 2.64, 2.13	0.686	7757
7	1	14	1.5	1.0	10.22, 6.62, 3.73, 2.64, 2.13	0.686	4761
8	2	14	1.5	1.0	10.22, 6.62, 3.73, 2.64, 2.13	0.686	4964

<sup>a</sup>Results are sorted by their enrichment factors (EFs), area under the receiver operating characteristics curve (AUC), and runtime.

interfaces were aligned. All interfaces of the *RunTimeSet* were prepared beforehand, and we only measured the time for the comparisons. iAlign (version 1.1) was downloaded (<https://sites.gatech.edu/cssb/ialign/>) as a precompiled executable. The method was executed with the default parameters. However, the minimum number of residues for a protein–peptide interfaces that are part of the numerous datasets analyzed herein. Also, we tested both available scoring functions in this work: the IS-score (default, iAlign-IS) and the TM-score (iAlign-TM). For the *RunTimeSet*, 6628 interfaces were not correctly detected due to lowercase chain identifiers that iAlign cannot process (3.9%). For the remaining set, 163,274 interfaces were successfully parsed. Besides the missing interfaces between chains with lower-case IDs, some interfaces, e.g., the interface between chains B and C of PDB entry 8eav (structures with unknown sequences, i.e., containing residues with three-letter code UNK), could not be found. Altogether, 163,254 alignments were obtained. For the run-time analyses, all PDB files were prepared beforehand, and the interfaces of the *RunTimeSet* were extracted. Only the runtime for the comparisons was measured.

**Parameter Optimization.** The parameters of PiMine were optimized on the *ParamOptSet* introduced above. Based on this dataset, we evaluated the parameters maximum distance (*maxDist*), minimum distance (*minDist*), distance tolerance (*distTolerance*), and scoring radius (*scoringRadius*) (see section [Alignment Algorithm](#)). The maximum distance was varied between 11 and 14 Å in 0.5 Å steps. This range ensures that filters are generated for all interfaces in the *RunTimeSet*. To evaluate this, we varied the maximum distance and looked at the number of generated filters. For a maximum distance below 11 Å, no filters were generated for some interfaces. The minimum distance was increased from 1 to 3 Å in steps of 0.5 Å, while the distance tolerance values were tested from 0.5 to 1.5 Å in 0.5 Å steps. The scoring radius was varied between 1.25 and 2.0 Å in 0.25 Å steps. Besides that, we decided to use  $\alpha$ -carbon atoms for filter generation as the default instead of also looking at other atoms to improve the runtime. Most comparison tools, e.g., InterComp<sup>26</sup> and PatchBag,<sup>27</sup> showed good performance although relying solely on  $\alpha$ -carbon atoms. Also, we set the default number of generated filters to 30. It is used to select at least as many filters as are required to cover the complete modeled query protein interface. The second filter hierarchy level is used by default, which describes the selected atoms' chemical elements and properties (e.g., hydrogen bond donor or hydrogen bond acceptor). Consequently, we investigated  $7 \times 5 \times 3 \times 4 = 420$  parameter combinations.

The parameter combination results were first sorted according to their non-normalized enrichment factor (EF) values at 1, 2, 5, 10, and 20%, second to the area under the receiver operating characteristics (ROC) curve (AUC), and third to the runtime. Considering all of the results, the change in the minimum distance has no impact. On the other hand, the performance increases with increasing maximum distance, distance tolerance, and scoring radius. The results for the best eight parameter combination sets are listed in [Table 4](#). All further parameter sets lead to values of 9.43 or lower for the EF at 1%.

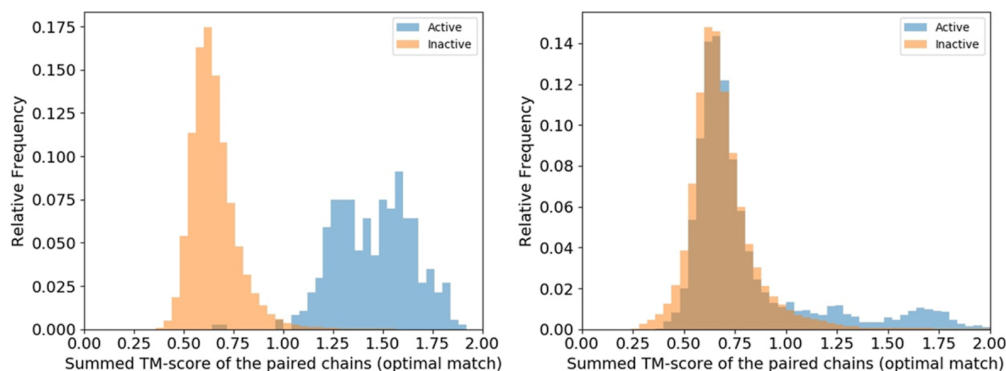
The parameter sets of 1 to 5 lead to the same EFs and only show a slight AUC difference of 0.001. Entries 6 to 8 lead to a slight decrease of the EF at 20% (2.17 to 2.13) and a lower AUC. Results 7 and 8 had the shortest runtimes of the displayed sets. Because the AUC of parameter set 2 only differs in the third decimal place from parameter set 1 but requires 43% less runtime, it is preferred over set 1 and represents the “accuracy-optimized” setting. As parameter set 7 is the fastest parameter set with nonetheless high AUC and EFs, we selected it as the “runtime-optimized” setting.

**Runtime Analysis.** Runtime calculations were performed for the *RunTimeSet* on a PC equipped with an Intel i5-9500 (3.0 GHz) processor, 32 GB of main memory, and both a Toshiba KBG40ZNS512G solid-state drive (SSD, 512 GB, model NVMe) and a Hitachi HUA722020ALA330 hard-disk drive (HDD, 2 TB) with an xfs file system. The PostgreSQL database used for PiMine was initialized and run on the same computer. PostgreSQL was initialized on either the HDD or SSD. The PostgreSQL parameters are listed in the [Supporting Information](#) (Tables S1 and S2). Runtimes were measured using the Linux command line tool “/usr/bin/time” (wall clock time).

We do not list any runtimes of the SQLite database because these are about three times longer. We do not recommend using this database type for large databases, as SQLite performs searching using only one thread, while the PostgreSQL database potentially uses multiple available threads of the employed processor. Also, while SQLite works out of the box for PiMine, a PostgreSQL database is also easy to set up and use.

## RESULTS AND DISCUSSION

**Current Datasets for the Evaluation of Protein–Protein Interface Comparison Methods and an Unbiased Alternative.** The definition of binding site similarity and dissimilarity depends on the model used.<sup>50</sup> The model heavily influences the development of corresponding similarity measures, e.g., shape- versus pharmacophore- versus com-



**Figure 2.** Global similarity of the interface chains in the similar and dissimilar protein-protein interface pairs of the *Dimer597* (left) and the *Keskin* (right) set. TM-align<sup>18</sup> was applied to compare the chains of the protein-protein complex pairs. For the chains A and B of the first protein-protein complex and the chains A' and B' of the second protein-protein complex, we can find two potential pairings (AA' and BB' or AB' and BA'). The chain pairing with the highest sum of TM-scores for both chains was used to generate the histograms. The blue bars indicate the distribution of TM-scores for the dissimilar interfaces ("inactive"), while the orange bars show the distribution of the TM-score for interfaces regarded as similar in the dataset ("active").

**Table 5. Normalized Enrichment Factors (EFs) of the Three Methods iAlign, I2I-SiteEngine, and PiMine with Their Respective Scoring Functions on the PiMineSet**

EF at	iAlign		I2I-SiteEngine		PiMine		
	TM-score	IS-score			SP-score	pharma score	shape score
0.1%	1.0	1.0	1.0		1.0	1.0	1.0
0.5%	1.0	1.0	1.0		1.0	1.0	1.0
1.0%	1.0	1.0	1.0		1.0	1.0	1.0
2.0%	0.98	0.89	0.62		0.96	0.96	0.98
5.0%	0.86	0.73	0.52		0.87	0.88	0.86
10.0%	0.91	0.78	0.58		0.92	0.94	0.88
20.0%	0.94	0.83	0.66		0.96	0.96	0.96

plementarity-based measures, sequence- vs structure-based similarity assessments, or even simple descriptor-based analyses. However, in structure-based modeling, the objective classification of site pairs is rarely undertaken, although it is the only robust way to reliably compare methods with differing underlying similarity measures. An analysis of the currently applied datasets for evaluating interface comparison methods underpins this phenomenon. We summarize the underlying hypotheses for establishing these datasets in the [Supporting Information](#) (Paragraph S2). The *Dimer597* set relies on similar SCOP superfamily assignments of the chains forming the interface. Thus, both chains share a similar fold, and structure comparison methods should detect these similarities (Figure 2). Furthermore, the definition of dissimilar interfaces is exclusively based on the SCOP family. Therefore, interface pairs in the dataset might share a high local similarity in terms of pharmacophore and shape properties. The authors tried to avoid biologically irrelevant interfaces due to crystal packing by scoring the interaction energy. In contrast, potential crystal artifacts were excluded based on distances in the *Keskin* set. This dataset relies on the assumption that similar interfaces should share a similar geometrical arrangement of  $\alpha$ -carbon atoms, identical interface residues, and similar size.

The key research question for tools that detect similar protein interfaces is the identification of potential binding partners. As indicated above, fast global protein comparison tools such as FoldSeek<sup>45</sup> are highly suitable for detecting obvious global similarities to deduce binding partners based on global chain similarity. However, we find cases of interface similarities without global fold similarity, posing a major

challenge for developing interface comparison methods.<sup>51</sup> To overcome the lack of appropriate datasets for such scenarios, we propose a workflow that looks for globally related chains in proteins. However, we consider only protein-protein interface pairs whose second chain pair is globally structurally unrelated and whose interacting chains bind to similar regions. Therefore, we can assume that the globally unrelated chains share common interface properties to enable binding to very similar interfaces. Based on an analysis of interfaces predicted as biologically relevant (see the [Methods](#) section for more details), we could identify 77 pairs of proteins of this type. Hiding them in a set of interface pairs where the partner chain pair binds to globally related chains but in different regions enabled us to establish a dataset of similar interfaces that are not biased by global or fold similarity. Therefore, considering only the globally unrelated chain pairs and assessing whether tools can enrich them based on the score will provide us with the most unbiased set of similar interfaces we can achieve.

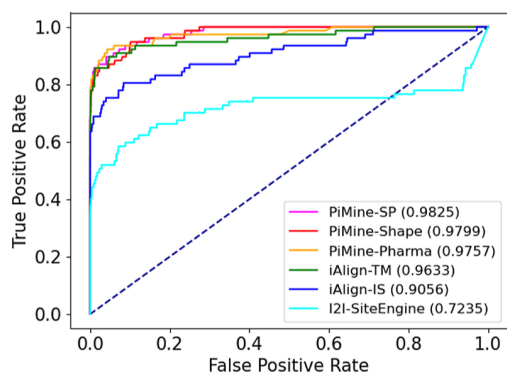
**Evaluating the Performance Using Both Chains.** The performance of PiMine regarding the ability to distinguish similar interfaces from less similar ones is compared with one of two existing methods that are among the most cited in this context: iAlign and I2I-SiteEngine. For evaluating I2I-SiteEngine, we use the total score as formerly reported on the web server. For PiMine, we show the results with the runtime-optimized parameters; results using the accuracy-optimized parameters are shown in the [Supporting Information](#) (Figures S2–S5). Considering both chains of the interface, the methods show promising early enrichment for the *PiMineSet* (Table 5). The EFs at 0.1, 0.5, and 1% are

G

<https://doi.org/10.1021/acs.jcim.3c01462>  
J. Chem. Inf. Model. XXXX, XXX, XXX–XXX



perfect. From 2% onward, the EFs decrease. In particular, iAlign with the TM-score and PiMine, irrespective of the scoring function, manage to correctly predict similar interfaces in the top-ranked pairs, while I2I-SiteEngine is considerably worse at higher percentages. Again, from 2% onward, iAlign's IS-score performs worse than its TM-score, with EFs on average lower by 0.12. The ROC curve reflects this trend (Figure 3). With an AUC of approximately 0.98, all scoring



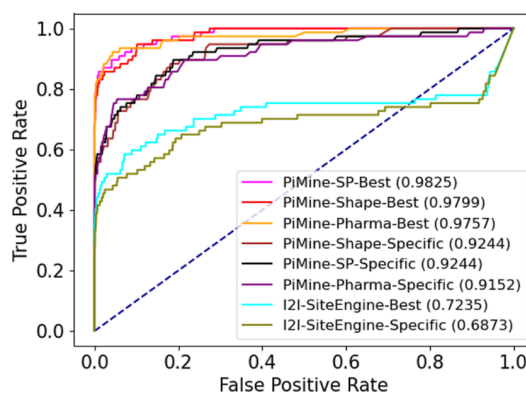
**Figure 3.** ROC curves for predicting related interfaces using iAlign, I2I-SiteEngine, and PiMine when comparing all interface chain pairs of the *PiMineSet*.

functions of PiMine lead to convincing results. iAlign achieves a similar AUC to PiMine with the TM-score. This good performance is likely due to the presence of sequentially related chains that are easy to detect by this sequence-dependent method. Using the IS-score leads to a significantly lower AUC. Thus, the contact overlap factor of iAlign's IS-score reduces the accuracy. Intriguingly, iAlign performs even poorer for sequence-independent calculations (Figure S6), indicating that this setting is not necessarily beneficial for retrieving both-remote interface similarities and sequence-dependent relationships. I2I-SiteEngine has the lowest AUC (0.72) and early enrichment. After an increase up to TPR 0.7, the slope of the ROC curve of I2I-SiteEngine decreases only slightly and the curve drops below the baseline from a TPR of 0.75. I2I-SiteEngine builds triangles for both single-chain protein interfaces of the protein-protein interface and does not focus on only one side. This approach usually considers interactions between chains but, in this case, prevents the enrichment of similar interfaces in the top-ranked interface pairs.

**Evaluating the Performance Using a Single Chain.** To evaluate the performance in binding partner detection, we took all complexes of the *PiMineSet* and removed the sequentially similar protein chains from the complexes for scoring. The second chain of the complex was only used for defining the interface residues of the query chain and was not considered in the comparison steps. Thus, only the sequentially and structurally unrelated chains with a TM-score below 0.5 were compared (Table 6 and Figure 4). iAlign requires two chains for the definition of an interface. A definition of interface residues of only one chain is not feasible with the publicly available tool of iAlign. Therefore, this tool could not be benchmarked accordingly. As expected, the early enrichment of PiMine and I2I-SiteEngine is lower than before. I2I-SiteEngine does not achieve a perfect EF from 1% onward, but overall, the EFs decrease by approximately 6% on average compared with the results for considering the related chains as well. PiMine's

**Table 6.** Normalized enrichment factors (EFs) of I2I-SiteEngine and PiMine on the *PiMineSet* When Evaluating Interface Similarities Only between Dissimilar Chains

EF at	I2I-SiteEngine		PiMine		
	SP-score	pharma score	shape score		
0.1%	1.0	1.0	1.0		1.0
0.5%	1.0	1.0	1.0		1.0
1.0%	0.89	1.0	1.0		1.0
2.0%	0.58	0.78	0.73		0.75
5.0%	0.47	0.68	0.68		0.64
10.0%	0.52	0.75	0.77		0.74
20.0%	0.62	0.87	0.86		0.88

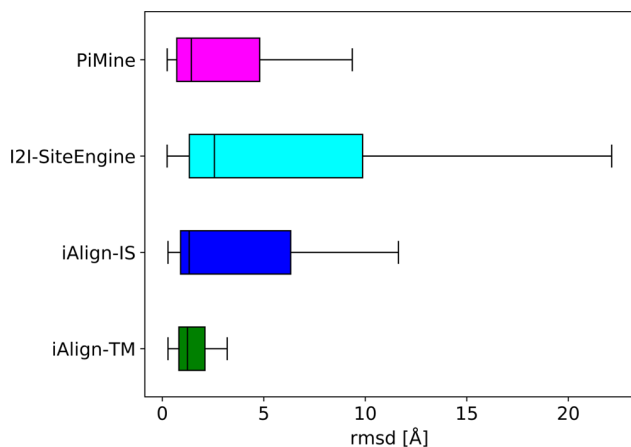


**Figure 4.** ROC curves for predicting related interfaces using PiMine and I2I-SiteEngine on the *PiMineSet*. Both the highest scoring results called “best” and the ones displaying the interface similarity between the dissimilar chains (“specific”) are shown.

EFs decrease by 14.75, 17.5, and 17.75% on average for the SP, pharma, and shape scoring functions, respectively. Except for the EFs at 5% dataset coverage, all EF values are above 0.7, indicating promising early enrichment for binding partner detection. As users will usually investigate only a tiny fraction of the best-scored matches, they can expect a high percentage of meaningful similarities in these results. All scoring functions of PiMine lead to an AUC of approximately 0.92. On average, these AUC values are about 0.06 lower than when also comparing the highly similar chains. I2I-SiteEngine reaches an AUC of 0.69. The difference in the AUC when including similar chains is only 0.03. When excluding similar chains, PiMine performs worse but still gives convincing results. The early enrichment is still higher than I2I-SiteEngine's, with a TPR of 0.55 versus about 0.42. Even above this TPR, the ROC curves rise significantly, showing that PiMine can still distinguish between related and unrelated interface pairs, even without using chains with high sequence and structure similarity.

**Alignment Performance.** Due to the nature of the *PiMineSet*, we can also prepare “correct” alignments based on the sequentially and structurally related chains of both protein-protein complexes. These alignments were obtained using TM-align.<sup>19</sup> Accordingly, we can evaluate the ability of the interface comparison methods to produce the correct alignments. To this end, we calculated the rmsd between the unrelated protein chains of the active interface pairs for alignments of the three methods under investigation. For homodimeric structures, we generated both possible alignments and selected the minimum rmsd for the chains of both alignments. Figure 5 shows the

H

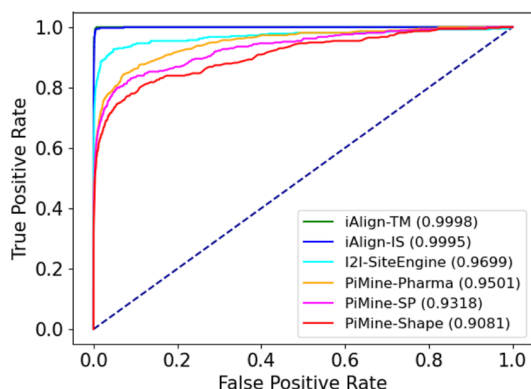


**Figure 5.** Box plots showing the rmsd distributions for the alignments of the active pairs of the *PiMineSet*. The rmsd values of ten pairs not aligned by I2I-SiteEngine are missing in the corresponding plot.

results of this analysis. The overall quality of the I2I-SiteEngine alignments with a median of 2.57 Å, but an upper quartile reaching up to 10 Å is poor. In contrast, iAlign performs well with medians of 1.32 and 1.23 Å for the IS-score and TM-score, respectively. This finding is expectable considering that iAlign compares complete chains, calculates all four possible pairwise chain alignments, and chooses the one with the highest score. Correspondingly, it nearly perfectly reproduces the TM-align-based alignments. As discussed above, we could not test whether aligning the structurally more dissimilar chain pairs might lead to similar results. PiMine shows an equally good performance with a median of 1.43 Å. Considering that the method only compares interface residues, we consider its alignment accuracy as convincing.

**Benchmarking Protein-Protein Interface Comparison Methods on Earlier Datasets.** In the first paragraph, we highlighted the drawbacks of earlier datasets. However, as users might want to focus on a distinct model of interface similarity, we also evaluated PiMine for these datasets in the following.

**The Dimer597 Set.** Figure 6 shows the results for the *Dimer597* set as ROC curves. Here, our structure-driven method performs worse than iAlign with a focus on the protein sequence and slightly worse than I2I-SiteEngine with AUCs of approximately 0.95 for the pharma score (orange), 0.93 for the



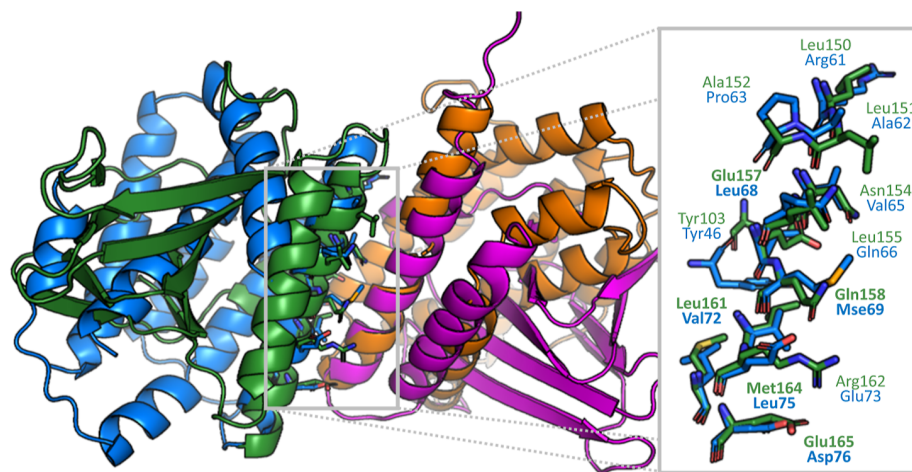
**Figure 6.** ROC curves for predicting related interfaces using three methods, iAlign, I2I-SiteEngine, and PiMine, to compare the interface pairs of the *Dimer597* set.

SP-score (pink), and 0.91 for the shape score (red). Notably, the performance of iAlign with an AUC of nearly 1 is superior for this dataset. The dataset was created to assess the function of iAlign. The authors used TM-align, which strongly resembles iAlign in the algorithmic approach. The latter and the usage of whole protein chains by iAlign explain the nearly perfect AUC and EFs. The PiMine pharma score performs best compared with the worst-performing shape score. The SP-score combines pharma and shape scores, which leads to an AUC of over 0.93, generally indicating the comparatively good performance of a method. In a real-life scenario of screening an interface database, we would expect the methods scoring similar pairs highest and would only consider a tiny fraction of the whole dataset. Therefore, we also analyzed the early enrichment. At a percentage of 0.1% or 168 actives out of 177 pairs of the dataset, iAlign achieves a normalized EF of 0.97 and 0.95 for the TM-score and IS-score, respectively. I2I-SiteEngine's-normalized EF is at 0.86, while the ones for PiMine are 0.72 (SP), 0.72 (pharma), and 0.71 (shape). The main difference between PiMine and the other tools is the scoring method. While I2I-SiteEngine calculates the sum of the similarities of both interface chains and iAlign also considers both chains forming the interface, the PiMine score constitutes the maximum similarity between a pair of chains of both interfaces. When considering both chains of the corresponding interfaces for alignment selection and scoring, PiMine achieves an EF of 0.91 using the pharma score (Table S3). The corresponding ROC curves are depicted in Figures S7 and S8. Nevertheless, we decided against this setting for PiMine, which is specifically designed for single-chain-based applications such as binding partner prediction.

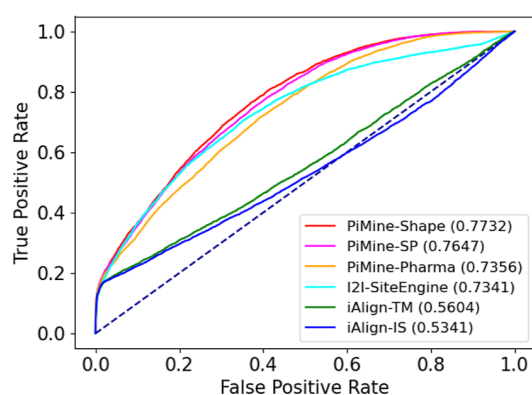
Overall, iAlign and I2I-SiteEngine reliably predict similarities between the interfaces of highly related proteins. However, there might be a significant number of false negatives in the dataset, as it relies on the assumption that chains with low overall structural similarity do not have similar interfaces. However, this does not necessarily hold, as shown in various studies.<sup>51,52</sup> One of these potential false negatives is the detected interface similarity of the structures with PDB code 1xja (interface between chains C and D) and 1wwm (interface between chains A and B) (Figure 7). The score calculated by PiMine for this protein-protein interface is ranked 49th for the 177,248 pairwise comparisons of the *Dimer597* set. The alignment indicates a similarity of the two single-chain interfaces not captured in the dataset.

Thus, even though our method appears inferior when applied to this dataset, it was developed with a focus on similar interfaces without explicit sequence information and finding nonobvious similarities not detectable based on overall structural similarity. This feature might lead to the high false positive rate of our method, as the dataset does not account for such remote relationships.

**The Keskin Set.** For this dataset, PiMine performs comparably to I2I-SiteEngine (Figure 8) and it achieves the best results with the shape and SP-scores. The pharmacophore-based score of PiMine performs worst with an AUC of 0.74, while I2I-SiteEngine achieves an AUC of 0.73. In contrast, iAlign performs significantly worse. If using the sequence-independent setting, the AUC increases slightly, but still, the method performs considerably poorer than PiMine and I2I-SiteEngine (Figure S9). The early enrichment of the three methods (Table 7) shows that all methods perform well at 0.1 and 0.5%. PiMine performs best and achieves EFs of up to 0.93



**Figure 7.** Alignment of the protein-protein interfaces of chains D (green) and C (magenta) of the PDB entry 1xja and chains B (purple) and A (orange) of entry 1wvm. This match represents a *Dimer597* set pair classified as dissimilar. Matching interface residues are shown on the right. Residues in bold represent residues whose  $\alpha$ -carbon atoms were used to generate the best-scoring tetrahedron filter. Molecular graphics generated with the PyMOL(TM) Molecular Graphics System, version 2.3.<sup>53</sup>



**Figure 8.** ROC curves for predicting related interfaces using three methods, iAlign, I2I-SiteEngine, and PiMine, to compare the interface pairs of the *Keskin* set.

or 0.63 in contrast to iAlign with up to 0.86 or 0.56 and I2I-SiteEngine with 0.86 or 0.51 at these percentages.

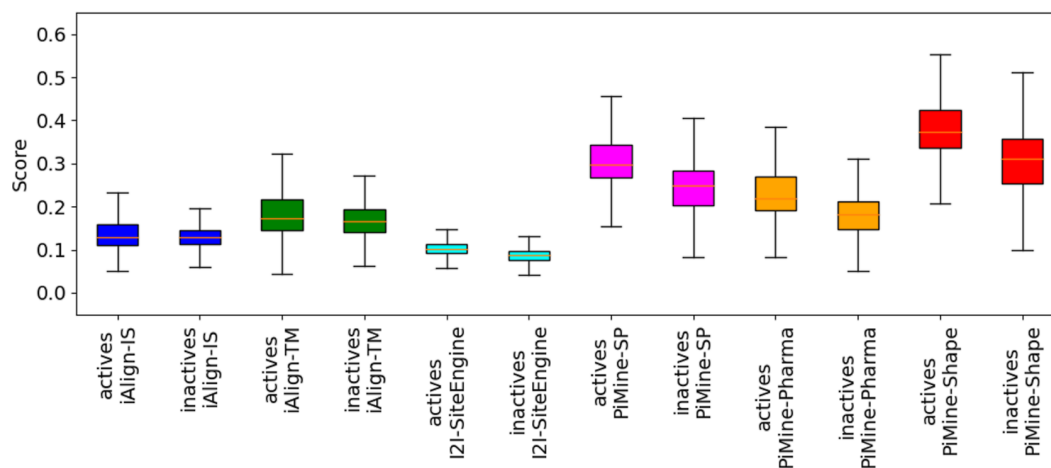
Starting from a true positive rate of about 0.19, iAlign's ROC curve increases slowly. At TPRs between 0.6 and 1.0, the IS-score decreases below the random baseline. This poor performance of iAlign may result from the lack of sequential similarities, as iAlign relies on global structure similarities (see the Introduction). Up to an FPR of 0.58, I2I-SiteEngine performs at a level comparable to the shape and SP-score of PiMine. After that, the I2I-SiteEngine's TPR drops below the one of PiMine. The ROC curves for the shape and SP-scores of PiMine are nearly indistinguishable. While the pharma score achieves a comparable early enrichment, the AUC is considerably lower. To further analyze how well the scores of the methods differentiate between actives and inactives, we

normalized the scores between zero and one and created box plots (Figure 9). A plot containing the outliers considering the complete score range can be found in Supporting Information (Figure S10). According to the box plots, the score distributions of the actives and inactives overlap for each method. For iAlign, however, the interquartile range overlaps nearly completely. The median scores for the similar and dissimilar pairs are nearly identical, leading to the low AUC of iAlign. Compared to the *Dimer597* set, iAlign scores for the actives in the *Keskin* set are much lower (Figure S11) and similar to the scores of the inactive pairs in the *Dimer597* set. In contrast, I2I-SiteEngine and PiMine distinguish well between similar and dissimilar interfaces. Of the PiMine scores, the shape score leads to the highest AUC values. These results suggest a high structural similarity between the interface pairs despite the low sequence similarity and a high degree of similarity between the interface surfaces. In summary, PiMine and I2I-SiteEngine perform well on this set of sequentially unrelated protein chain pairs with, nevertheless, structurally similar interfaces.

**Runtime.** The *RunTimeSet* was used for all subsequent analyses. For the run-time measurements, we randomly generated an index number over all interfaces and picked the interface corresponding to this index to analyze against all others of the dataset. Next, we preprocessed all interfaces with PiMine, iAlign, and I2I-SiteEngine. PiMine created a database comprising 59,803 structures and 169,689 interfaces. Overall, 1300 interfaces of 300 PDB entries were not processed, corresponding to a coverage of 99.8% of the structures and 99.4% of the interfaces. Missing structures can be attributed to very short peptides. PDB entry 1b05, for example, is missing because one chain consists of only three residues. After the

**Table 7.** Normalized EFs at 0.1 and 0.5% of the Three Methods iAlign, I2I-SiteEngine, and PiMine with Their Respective Scoring Functions on the *Keskin* Set

EF at	iAlign-TM	iAlign-IS	I2I-SiteEngine	PiMine		
	TM-score	IS-score		SP-score	pharma score	shape score
0.1%	0.86	0.82	0.86	0.93	0.93	0.91
0.5%	0.56	0.54	0.51	0.63	0.62	0.61



**Figure 9.** Box plots showing the score distributions of actives and inactives for the *Keskin* set using three methods, iAlign, I2I-SiteEngine, and PiMine. Scores are normalized between zero and one. For a better overview, outliers are omitted.

database was created, the runtimes for calculating the similarity scores were measured (Table 8). Generally, PiMine is the

**Table 8. Runtime Analysis of PiMine, iAlign, and I2I-SiteEngine Using the *RunTimeSet* (169,944 Comparisons)<sup>a</sup>**

method	drive	runtime [h]
PiMine (runtime-optimized)	HDD	41.0
PiMine (accuracy-optimized)		87.7
iAlign		3.9
I2I-SiteEngine		371.6
PiMine (runtime-optimized)	SSD	19.1
PiMine (accuracy-optimized)		39.0
iAlign		3.4
I2I-SiteEngine		369.6

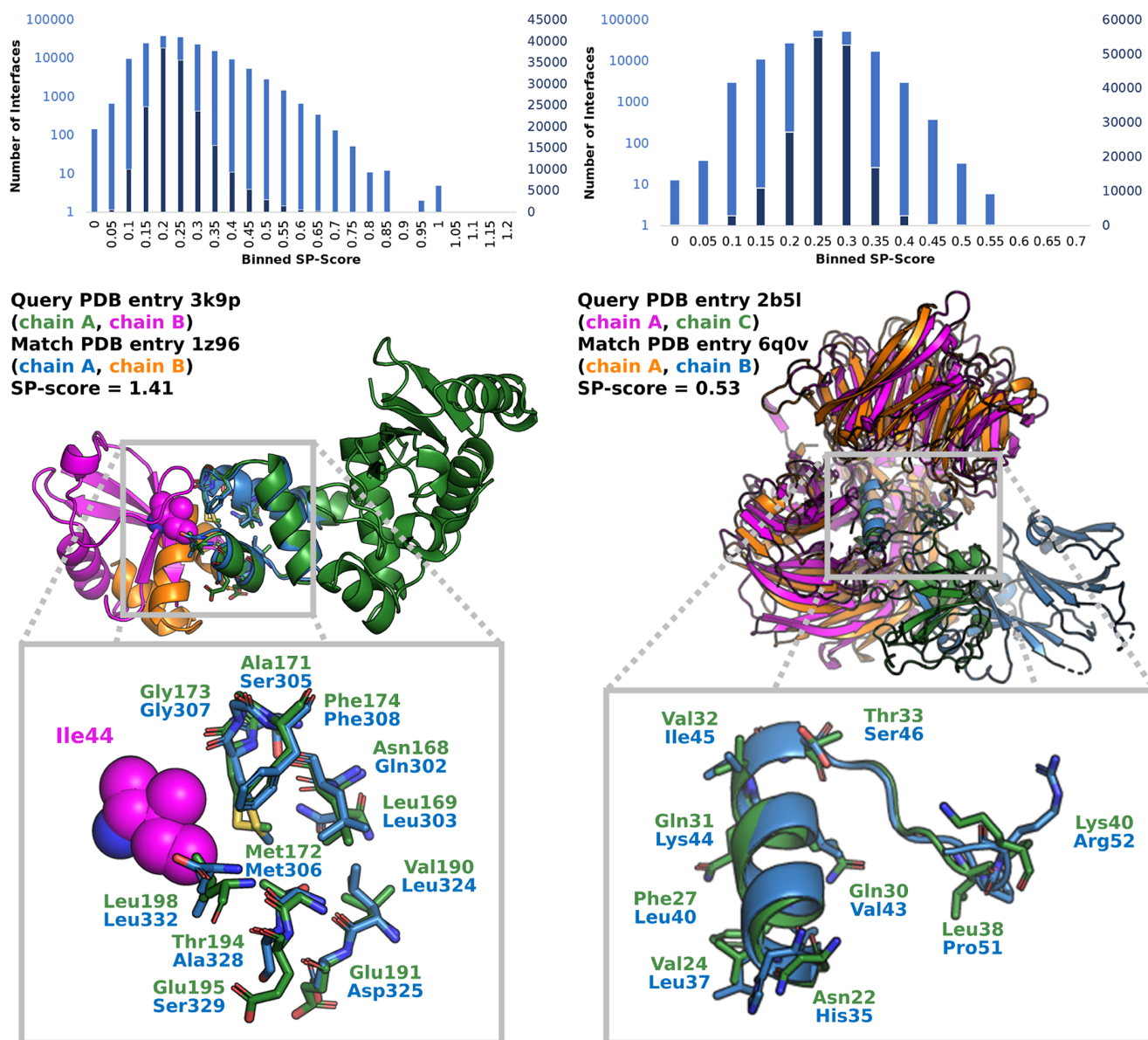
<sup>a</sup>PiMine's runtime is assessed with the runtime-optimized and accuracy-optimized parameters using a PostgreSQL database initialized either on a hard-disk drive (HDD) or a solid-state drive (SSD). We measured the runtimes for screening this set for similarities to the randomly chosen interface between chains A and B of PDB entry 3t4m.

fastest when using runtime-optimized parameters and a PostgreSQL database. First, we only look at the runtimes using the HDD. The accuracy-optimized parameters are approximately 2.1 times slower than the runtime-optimized ones. The runtime of the I2I-SiteEngine is the longest among the three methods (371.6 h). It is 9.1 and 4.2 times slower than PiMine with the runtime-optimized and accuracy-optimized parameters, respectively. iAlign, with a runtime of 3.9 h, is about 95.3 times faster than I2I-SiteEngine and 10.5 or 22.5 times faster than PiMine with the runtime-optimized and accuracy-optimized parameters, respectively. As comparisons with PiMine and I2I-SiteEngine take multiple days and an up-to-date PC is often equipped with an SSD, we reevaluated the runtimes. Because I2I-SiteEngine uses a data structure requiring about 1.8 TB of storage space, we split the dataset into chunks of 10,000 interfaces and screened them consecutively with the tool. The runtime in Table 8 constitutes the sum of the runtimes for the individual chunks. PiMine performs about 54% better on the SSD than on the HDD. It calculates all similarity scores within 1–2 days, depending on the chosen parameters. PiMine's database-driven similarity analysis largely relies on short reading times. In contrast,

iAlign's and I2I-SiteEngine's runtimes on the SSD are similar to the ones on the HDD, with 3.4 h (approximately 13% faster) and 369.6 h (approximately 1% faster), respectively. Therefore, we assume the similarity calculation to be more time-consuming than reading the interfaces. In summary, iAlign is much faster than PiMine and preferable if users intend to screen large databases with evolutionary-related PPIs within minutes. In most other cases, a runtime of 19 h for PiMine with runtime-optimized parameters is acceptable for most use cases.

**Case Studies. Retrospective Application Examples.** We finally investigated the applicability of PiMine in practice. To this end, we looked for application studies using the tools, as listed in Table 1. We screened citing articles for the successful applications of the corresponding tools on PubMed (PubMed, Bethesda (MD): National Library of Medicine (US), <https://www.ncbi.nlm.nih.gov/pubmed/>, last access: November 30th, 2023).

Keren-Kaplan and co-workers use the SiteEngine<sup>16</sup> algorithm to identify ubiquitin-binding domains (UBDs).<sup>54</sup> To assess the performance of PiMine to also detect the same UBDs in a set of interfaces of sequentially diverse chains, we hid the PDB entries with an interface of two chains (PDB entries 3k9o, 2bw b, 1z96, 3b0f, 2o0a, 3ihp, 2qho, 4ae4, and 1wrd) in our *RunTimeSet* and used the interface of PDB entry 3k9p as query (ubiquitin-conjugating enzyme E2 K in a complex with ubiquitin). The three best-scored hits are known examples of UBDs from the corresponding publication (PDB entries 3k9o, 1z96, and 2qho). A fourth PDB entry (2bw b) was on rank 141 with an SP-score of 0.67 (Figure S12). Figure 10 shows an exemplary alignment of the ubiquitin-conjugating enzyme E2K and the ubiquitin-associated (UBA) domain-containing protein Mud1 (chain A of PDB entry 1z96). Other PDB entries reported by Keren-Kaplan et al. were not significantly high-ranked by PiMine. The PDB entry 2o0a is also a homodimer. Comparing this entry to a structure in complex with ubiquitin (PDB entry 2o0b) shows that a reasonable interface was found in the database. Nevertheless, PiMine could not detect any similarity. A visual comparison of the interfaces of PDB entries 3k9p (chain A) and 2o0a (chain B) does not reveal any specific physicochemical similarities besides some similar hydrophobic residues. For chain B of PDB entry 3b0f, a homodimeric structure, we cannot ensure



**Figure 10.** Score distributions and alignments for the retrospective application examples. Published similar protein interfaces were hidden in the *RunTimeSet*. Retrieval of these interfaces with the originally used single-chain query interfaces was analyzed. Score distributions are represented in a logarithmic (blue) and linear (orange) scale. Left: according to a published example by Keren-Kaplan and co-workers,<sup>54</sup> chain A of PDB entry 3k9p (ubiquitin-conjugating enzyme E2 K) was used to query the dataset. Alignment for one of the three highest-scored true positive matches (chain A of PDB entry 1z96, UBA-domain protein Mud1) is depicted below the corresponding score distribution. Similar residues are labeled in the enlarged depiction at the bottom. Right: according to a published example by Cheng and colleagues,<sup>24</sup> chain C of PDB entry 2b5l (Simian virus 5 V protein) was used to query the dataset. The alignment with the second highest-scored match (chain B of PDB entry 6q0v, DDB1- and CUL4-associated factor 15) is depicted below the corresponding score distribution. Residues 398 to 713 of chain B of PDB entry 2b5l were omitted for visualization purposes. Similar residues are labeled in the enlarged depiction at the bottom. Molecular graphics generated with the PyMOL(TM) Molecular Graphics System, version 2.3.<sup>53</sup>

that the interface relevant to the interaction with ubiquitin was stored in the database, as there are no known complexes with ubiquitin. An alignment to the homodimer of PDB entry 2oaa shows that the screened interface does not correspond to the ubiquitin-binding interface. The interface region between ubiquitin and chain A of PDB entry 3ihp is much larger than the one in the query structure, explaining the low rank of this interface (SP-score = 0.19). For PDB entry 4ae4, we cannot ensure having stored the correct interface region interacting with ubiquitin, as the structure is a homodimer and there is no known structure in a complex with ubiquitin. The SP-score for

the interface of PDB entry 1wrđ (SP-score = 0.38) indicates no significant similarities detected by PiMine. However, both interfaces share some common residues. Searching with the interfaces of both chains of the query complex, we find PDB entry 1wrđ at rank 11 (interface between chains A and B). Besides the four detected hits, we find several high-scoring matches with uncharacterized proteins, KDPG aldolases from different organisms (e.g., PDB entry 1vhc, *Haemophilus influenzae*), and the Holliday junction ATP-dependent DNA helicase RuvA (e.g., PDB entry 3ik5, *Salmonella enterica*), which is already known for its similarity to other proteins of

L

<https://doi.org/10.1021/acs.jcim.3c01462>  
J. Chem. Inf. Model. XXXX, XXX, XXX–XXX

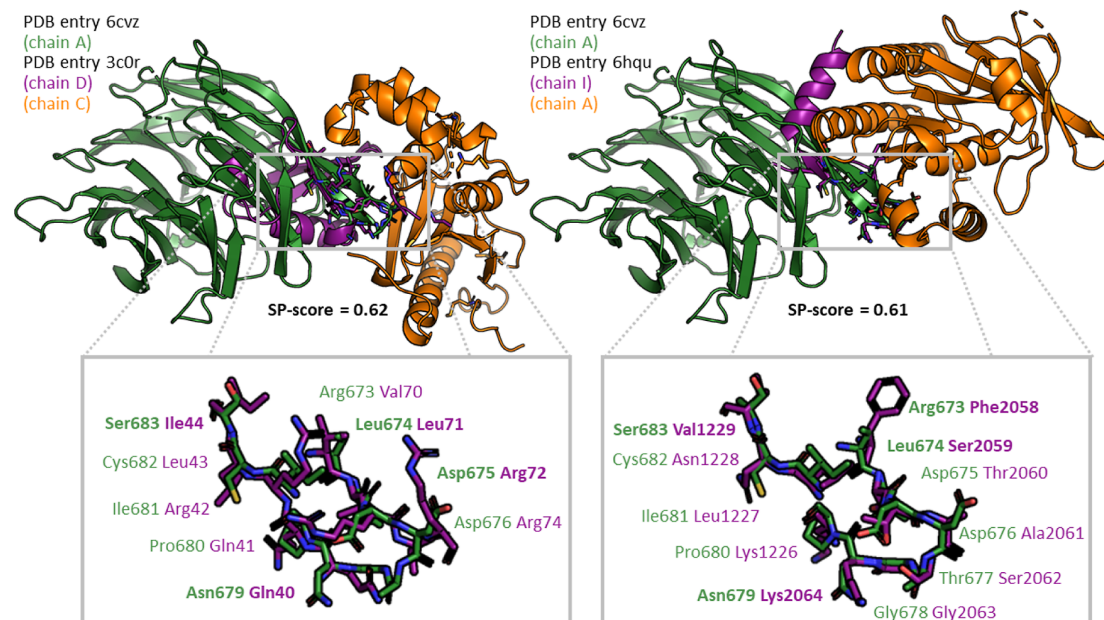
the CATH superfamily of UBA domains. Keren-Kaplan and co-workers might not have found these similar interfaces, as they restricted their search to PDB chains from eukaryotic organisms. Exemplary alignments are depicted in the [Supporting Information](#) (Figure S12) and show convincing similarities in a helix interacting with the recognition patch of ubiquitin with a characteristic Ile44. To the best of our knowledge, these proteins are not annotated as ubiquitin-binding partners. However, the interactions of proteins with ubiquitin are manifold<sup>55</sup> and highly variable. The high SP-score indicates potentially uncharacterized ubiquitin-binding patches on these proteins. However, we could not verify this finding due to a lack of structures of these proteins in complex with ubiquitin. In summary, we can show that PiMine enriches similar interfaces, although we are missing some of the previously detected interfaces hidden in the dataset.

Cheng and co-workers used their method PCAlign to explore known cases of viral mimicry.<sup>24</sup> Altogether, they presented three case studies. We explored the ability of PiMine to retrieve the described interface similarities in our *RunTimeSet*. To this end, we hid the corresponding similar interfaces to the viral interfaces in this set. The first example is the known interaction of the murine herpesvirus 4 M3 protein, a known chemokine binding protein, with CC chemokine ligand 2 (PDB entry 2nz1), modulating the human immune response.<sup>56</sup> The aim was to show a similarity toward a homodimer of the C–C motif chemokine 2 (PDB entry 1dok). Screening our dataset with the chain representing the M3 protein, we found this target interface only at rank 8482 with a very low SP-score of 0.44. The first human chemokine dimer pair that matches our single-chain interface is on rank 1013 (PDB entry 5cmd). The best-scored match is a complex between the M3 protein and the C–C motif chemokine 2 (PDB entry 1ml0). Besides, the most significant similarities were found to transcription factors RelA, RelB, nuclear factor NF- $\kappa$ -B p105 subunit (NFKB1), and nuclear factor NF- $\kappa$ -B p100 subunit (NFKB2), with SP-scores above 0.7. The score distribution indicates the high relevance of these matches ([Supporting Information](#), Figure S13). However, upon visualizing the corresponding alignments, we only find similarities regarding a single  $\beta$ -strand. The match cannot be used to relate to novel interaction partners as they severely clash in the alignment with the query chain. When using both chains of the interface for screening the dataset, we only find chemokines in the best-scored 50 results, indicating the general applicability of PiMine. However, these hits would probably not have been found by using only the structure of the viral protein. In a second study, Cheng and co-workers analyzed the structure of the Simian virus 5 V protein. Its interface is said to be similar to that of DNA damage-binding protein 2 (DDB2). The latter binds to DDB1, thus participating in UV-induced nucleotide excision repair, and stimulates E2F1-activated transcription. A blockage of this interaction by the V protein is known to support viral pathogenesis. Using chain C of PDB entry 2b5l, representing the interface of the V protein, we screened for the documented similar site of PDB entry 3ei4 (chains A and B) hidden in our *RunTimeSet*. Although this protein was only on rank 803, we found a complex of DDB1 and DDB1- and CUL4-associated factor 15 (DCAF15; PDB entry 6q0v, [Figure 10](#)) on the second-highest rank. The best-scored match corresponds to the interface of a dimer of 6-deoxyerythronolide-B synthase EryA2, modules 3 and 4 (PDB entry 1pzq), and seems to be a false positive match. This hypothesis could be verified by

visually inspecting the detected similarity. Only a single query helix overlaps with the found interface. More importantly, there are several clashes among the corresponding interface chains. Also, the ratio of the pharma and shape scores is low, indicating insufficient validity of the match. Intriguingly, other complexes of DDB1 and DDB2 were also found on very low ranks with PiMine. Using both chains, we find a complex of DDB1 and DDB2 at rank 6 (PDB entry 4a0l). The example of this complex found by Cheng and co-workers (PDB entry 3ei4) is on rank 28. Analyzing the alignments using only the viral chain and both interface chains as queries, we only find minor differences and see that an N-terminal helix as a crucial part of the interface is the main similarity. The best-scored hit, however, is a complex of DDB1 and the DNA excision repair protein ERCC-8 (PDB entry 4a11). Again, we find the N-terminal helix as the most striking similarity between the viral protein and the DDB1 binding partner. This helix was reported earlier as the crucial viral motif for mimicking DDB1 binding partners.<sup>57</sup> Depictions of all discussed alignments and score distributions can be found in the [Supporting Information](#) (Figure S14). Another example of viral mimicry is the interaction of Hendra virus glycoprotein G (PDB entry 2vsk, chain A) with ephrin-B2. Similarities of the viral glycoprotein and the cognate human cell–surface receptor (Eph) are known.<sup>58</sup> Using the viral glycoprotein interface (only chain A) to search for a known complex of EphB4 and ephrin-B2 (PDB entry 2hle) hidden in our *RunTimeSet*, we found various matches with glycoprotein complexes of other viruses (Nipah virus, e.g., PDB entry 2vsm, Ghanaian henipavirus, e.g., PDB entry 4uf7, Cedar henipavirus, e.g., PDB entry 6p7y, Cedar virus, e.g., PDB entry 6thg) in the best-scored 17 hits. However, the hidden interface is only on rank 19,382. A complex of an ephrin and its receptor is on rank 157 (PDB entry 1kgy): ephrin-B2 in a complex with ephrin type-B receptor 2. Using both interface chains of the query, this match is on rank 20, while PDB entry 4bkf, representing a complex of ephrin-B3 and ephrin type-A receptor 4, is on rank 9, with scores similar to the ones obtained for viral glycoprotein-ephrin complexes. We can explore the structural similarities between the human receptor proteins and the viral glycoproteins by visualizing the corresponding alignments ([Supporting Information](#), Figure S15). The similar interface detected by Cheng and co-workers is still on the lower ranks of the hit list with a low SP-score. A flexible loop region of ephrin proteins is mainly responsible for interactions with other proteins (GH loop). This loop interacts with the DE loop of the receptor proteins and a highly flexible JK loop.<sup>59</sup> Interactions between these loops are conformationally variable<sup>60</sup> and might explain why the enrichment and alignment of the viral protein to the similar host interface is challenging for PPI search tools like PiMine.

Altogether, these examples highlight that PiMine retrieves host partners of viral proteins. A visual inspection considering score distributions, the ratio of pharma to shape score, and an analysis of the alignments generated by PiMine enables the user to find truly valid hits. The corresponding alignments show that PiMine not only enriches true positive hits but also provides reliable alignments, often even in the absence of the corresponding binding partner interface. Most intriguingly, our method finds reliable hits even after omitting the globally similar chains of the interfaces.

**Showcase Study.** To demonstrate the applicability of our novel tool to reveal previously unknown similarities, we



**Figure 11.** Predicting interaction partners and the structure of the corresponding protein-protein complexes with PiMine. PiMine-based alignment of the protein-protein interfaces of the SPPIDER-predicted interface of chain A (green) of the E3 ubiquitin-protein ligase RFW3 (PDB entry 6cvz) and chains D (purple, human ubiquitin) and C (orange) of entry 3c0r (left), and chains I (purple) and A (orange, DNA repair and recombination protein RadA from *Pyrococcus furiosus* DSM 3638) of PDB entry 6hqu (right). Matching interface residues are shown below the alignments of the complete proteins. Residues in bold represent residues whose  $\alpha$ -carbon atoms were used to generate the best-scoring tetrahedron filter. Molecular graphics generated with the PyMOL(TM) Molecular Graphics System, version 2.3.<sup>53</sup>

investigated whether we could identify potential protein-protein complexes for structurally known proteins with low sequence similarity to other proteins in the PDB. We looked for proteins with known interaction partners but no known structure of the complex in the PDB. One example is the structure of human E3 ubiquitin-protein ligase RFW3 (PDB entry 6cvz). To date, sequentially related homologues with a sequence similarity of at least 30% cannot be found in the PDB. However, the protein interacts with Rad51, thereby mediating its ubiquitinylation and removal from DNA damage sites,<sup>61</sup> which, in consequence, enables homologous replication. RFW3 variants cause Fanconi anemia, complementation group W (FANCW) causing anemia, leukopenia, and thrombopenia,<sup>62</sup> indicating the impact of the knowledge of the structure of its interactions with other proteins. We predicted the interfaces of RFW3 using the web server SPPIDER<sup>63</sup> for protein chain A of PDB entry 6cvz. Upon looking for residues that are in close contact, creating a continuous interface, we chose residues Val570, Glu578, Val580, Gln582, Met622, Asp623, Trp627, Val630, Arg673, Leu674, Asp675, Asp676, Thr677, Gly678, Asn679, Ile681, Ser683, and Gln685 for a PiMine search in protein complexes predicted as biologically relevant (*RunTimeSet*). Unfortunately, comparing PiMine to iAlign and I2I-SiteEngine was infeasible, as both methods rely on a known complex structure between two chains. The alignments of the two highest-scored matches are presented in Figure 11. The score table for this PiMine search is provided in the repository.<sup>48</sup> We emphasize that PiMine did not find any false positive hits in the top-scored 30 hits of the three benchmark sets used for evaluating the method. Therefore, a user might not expect too many false positives in the highest-scored matches. However, a visualization of the alignments and the score distribution, both parts of the output of PiMine, will help to learn more about the detected similarities. Also, the

user should take a look at the highest scores. As seen and discussed earlier, looking at the score distribution might help to learn whether there are any significant similarities to consider.

The best-scored hit of this search was a match to the interface of chains C and D of PDB entry 3c0r (SP-score = 0.62). The interface of our query chain was aligned with the interface of ubiquitin (chain D) in this complex of human ubiquitin and the ubiquitin thioesterase OTU1. This enzyme, to the best of our knowledge, was never predicted as an interaction partner of our query protein. Therefore, we cannot evaluate the validity of this hit. However, a ubiquitin-like interaction with a target protein does not seem to correspond to the annotated function of the protein. Nevertheless, a striking similarity is evident. The significant difference between this score and the average score is depicted in Figure S16 in the [Supporting Information](#) and shows a considerable similarity compared with most other interfaces in the dataset. One also finds striking similarities to the so-called four-strand barrelizing versions of the  $\beta$ -grasp fold.<sup>64</sup> Although the helical structure in this type of protein is missing in the structure of RFW3, the structure is characterized by two strands forming a conserved insert in this type of protein fold, which is also the region aligned to ubiquitin by PiMine.

The second-best hit (SP-score = 0.61), however, is the humanized RadA mutant HumRadA22 from *Pyrococcus furiosus* DSM 3638 in complex with breast cancer type 2 susceptibility protein (BRCA2), which potentiates recombinational DNA repair (PDB entry 6hqu, interface between chains A and I). In the PiMine alignment, the two  $\beta$ -strands of the latter interaction partner are nicely superposed to the two  $\beta$ -strands of RFW3 (Figure 11). Based on the alignment, we find that a complex between RFW3 and HumRadA22 might well form in this way without steric clashes. One can hypothesize that

this predicted interaction might be relevant to the activity of RFW3 as a ubiquitin-protein ligase, enabling further structure-based research. In summary, this application example shows the benefit of PiMine enabling comparisons of predicted interfaces for comparisons to protein-protein interfaces and the suitability of PiMine to provide hints to the potential structure of an interface of two predicted interacting proteins.

## CONCLUSIONS

We presented PiMine, a sequence-independent structural similarity calculation and alignment method for protein-protein interfaces. PiMine aims to detect similarities between the interfaces of evolutionary unrelated protein chain pairs. We have shown that it finds similar protein-protein interfaces in the complete PDB within a single day and is considerably faster than I2I-SiteEngine. iAlign, which is much faster than both methods, reliably detects similarities between evolutionary-related complexes, while it performs weaker than PiMine and I2I-SiteEngine for sequentially unrelated but similar interface pairs. PiMine's ability to assess the individual scores for single interface pairs, avoiding the necessity of two chains defining an interface for comparison, renders it a valuable tool for predicting the structure of protein-protein complexes, identifying unknown partners of protein chains, or finding potential small molecule-binders of interfaces. Furthermore, PiMine is the method of choice if only one partner of a protein-protein interface is available to search for potentially interacting proteins. We could validate the usability of PiMine based on retrospective application analyses and a predictive case study. For screening settings, where only a low percentage of highest-scoring hits is analyzed, we recommend using the runtime-optimized PiMine parameters. However, if users are interested in specific similarities and a reliable classification of all interface pairs under investigation, e.g., for interface clustering, they can use PiMine's accuracy mode. The possibility of using single-chain interfaces predicted with external programs to screen a database of structurally characterized protein-protein interfaces is unique for the tool. Based on the score distributions, users get a good estimate of outstanding similarities. A future improvement might involve a statistical measure of the significance of a match. As it is currently possible to search only in known protein-protein interfaces, we intend to extend the search space to global protein surfaces in the future. Following this, an automated clash detection procedure between the individual chains might help to quickly eliminate false positive hits without the need for user intervention. In summary, we presented PiMine as a novel and reliable tool to compare and align protein-protein interfaces. We hope its capabilities and features will assist in broadening our structural and functional understanding of PPIs.

## ASSOCIATED CONTENT

### Data Availability Statement

PiMine is available online as part of the NAOMI ChemBio Suite (<https://uhh.de/naomi>), which is free for academic use and licensed for commercial use. All datasets and the similarity scores calculated by PiMine, iAlign, and I2I-SiteEngine are available at [10.25592/uhhfdm.13227](https://doi.org/10.25592/uhhfdm.13227).

### Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jcim.3c01462>.

examples for similar and dissimilar interfaces in the *PiMineSet*; ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (accuracy-optimized parameters) on the *DimerS97* set; ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (accuracy-optimized parameters) on the *Keskin* set; ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (accuracy-optimized parameters) on the *PiMineSet*; ROC curves for predicting similar interfaces of sequentially and structurally similar chains using I2I-SiteEngine and PiMine (accuracy-optimized parameters) on the *PiMineSet* excluding the interfaces of related single chains; ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters) on the *PiMineSet*; ROC curves for predicting related interfaces using the methods, iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters and scoring using both interfaces of the PPIs) on the *DimerS97* set; ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (accuracy-optimized parameters and scoring using both interfaces of the PPIs) on the *DimerS97* set; ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters) on the *Keskin* set; box plots showing the score distributions of actives (similar interface pairs) and inactives (dissimilar interface pairs) including the outliers for the *Keskin* set using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters); box plots showing the score distributions of actives (similar interface pairs) and inactives (dissimilar interface pairs) for the *DimerS97* set using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters); matches found with a PiMine search for chain A of PDB entry 3k9p (ubiquitin-conjugating enzyme E2-K in complex with ubiquitin); matches found with a PiMine search for chain A of PDB entry 2nz1 (M3 protein of murid herpesvirus 4 in a complex with C-C motif chemokine 2) and interface chains A (green) and D (magenta); matches found with a PiMine search for chain C of PDB entry 2b5l (Simian virus 5 nonstructural protein V in a complex with damage-specific DNA-binding protein (1) and interface chains C (green) and A (magenta)); matches found with a PiMine search for chain A of PDB entry 2vsk (Henda virus attachment protein glycoprotein G in a complex with ephrin-B2) and interface chains A (green) and B (magenta); score distribution for the PiMine search for similar interfaces to a predicted one of PDB entry 6cvz (human E3 ubiquitin-protein ligase RFW3) in the *RunTimeSet*; PostgreSQL 14.6 custom parameters for PiMine runs on the SSD (on the *RunTimeSet*); PostgreSQL 14.6 custom parameters for PiMine runs on the HDD (on the *RunTimeSet*); normalized enrichment factors of the methods iAlign, I2I-SiteEngine, and PiMine (scoring using both interfaces of the PPIs) on the *DimerS97* set; PiMine interface input; and external benchmark datasets for protein-protein interface comparisons (PDF)



## AUTHOR INFORMATION

### Corresponding Authors

Christiane Ehrh – Universität Hamburg, ZBH—Center for Bioinformatics, 22761 Hamburg, Germany; [orcid.org/0000-0003-1428-0042](https://orcid.org/0000-0003-1428-0042); Email: [christiane.ehrh@uni-hamburg.de](mailto:christiane.ehrh@uni-hamburg.de)

Matthias Rarey – Universität Hamburg, ZBH—Center for Bioinformatics, 22761 Hamburg, Germany; [orcid.org/0000-0002-9553-6531](https://orcid.org/0000-0002-9553-6531); Email: [matthias.rarey@uni-hamburg.de](mailto:matthias.rarey@uni-hamburg.de)

### Authors

Joel Graef – Universität Hamburg, ZBH—Center for Bioinformatics, 22761 Hamburg, Germany; [orcid.org/0000-0001-8327-4936](https://orcid.org/0000-0001-8327-4936)

Thorben Reim – Universität Hamburg, ZBH—Center for Bioinformatics, 22761 Hamburg, Germany; [orcid.org/0009-0002-7712-8515](https://orcid.org/0009-0002-7712-8515)

Complete contact information is available at: <https://pubs.acs.org/10.1021/acs.jcim.3c01462>

### Author Contributions

J.G. extended the TetraScan algorithm to protein-protein interface similarity searching and developed and tested PiMine. C.E. created the new datasets, performed the analyses of iAlign and I2I-SiteEngine, and carried out the case studies. J.G. and T.R. jointly developed the TetraScan algorithm for similarity searches. C.E. and M.R. participated in the method development process. M.R. supervised the project. J.G., C.E., and M.R. wrote the manuscript.

### Notes

The authors declare the following competing financial interest(s): ProteinsPlus and the NAOMI ChemBioSuite use some methods jointly owned by and/or licensed to BioSolveIT GmbH, Germany. M.R. is a shareholder of BioSolveIT GmbH.

## ACKNOWLEDGMENTS

The authors thank the whole development team of the NAOMI library, which formed the basis of this work. This work was supported by the German Federal Ministry of Education and Research as part of CompLS and de.NBI [031L0172, 031L0105]. C.E. and T.R. are funded by Data Science in the Hamburg—Helmholtz Graduate School for the Structure of Matter (Grant-ID: HIDSS-0002).

## ABBREVIATIONS

AUC area under the ROC curve  
 EF enrichment factor  
 HDD hard-disk drive  
 PPI protein-protein interaction  
 rmsd root-mean-square deviation  
 ROC receiver operating characteristics  
 SSD solid-state drive

## REFERENCES

- (1) Lu, H.; Zhou, Q.; He, J.; Jiang, Z.; Peng, C.; Tong, R.; Shi, J. Recent advances in the development of protein-protein interactions modulators: mechanisms and clinical trials. *Signal Transduct. Targeted Ther.* **2020**, *5*, 213.
- (2) Fischer, P. Protein-Protein Interactions in Drug Discovery. *Drug Des. Rev.* **2005**, *2*, 179–207.
- (3) Deeks, E. D. Venetoclax: First Global Approval. *Drugs* **2016**, *76*, 979–987.

(4) Oltersdorf, T.; Elmore, S. W.; Shoemaker, A. R.; Armstrong, R. C.; Augeri, D. J.; Belli, B. A.; Bruncko, M.; Deckwerth, T. L.; Dinges, J.; Hajduk, P. J.; Joseph, M. K.; Kitada, S.; Korsmeyer, S. J.; Kunzer, A. R.; Letai, A.; Li, C.; Mitten, M. J.; Nettesheim, D. G.; Ng, S.; Nimmer, P. M.; O'Connor, J. M.; Oleksijew, A.; Petros, A. M.; Reed, J. C.; Shen, W.; Tahir, S. K.; Thompson, C. B.; Tomaselli, K. J.; Wang, B.; Wendt, M. D.; Zhang, H.; Fesik, S. W.; Rosenberg, S. H. An inhibitor of Bcl-2 family proteins induces regression of solid tumours. *Nature* **2005**, *435*, 677–681.

(5) Tse, C.; Shoemaker, A. R.; Adickes, J.; Anderson, M. G.; Chen, J.; Jin, S.; Johnson, E. F.; Marsh, K. C.; Mitten, M. J.; Nimmer, P.; Roberts, L.; Tahir, S. K.; Xiao, Y.; Yang, X.; Zhang, H.; Fesik, S.; Rosenberg, S. H.; Elmore, S. W. ABT-263: A Potent and Orally Bioavailable Bcl-2 Family Inhibitor. *Cancer Res.* **2008**, *68*, 3421–3428.

(6) Souers, A. J.; Levenson, J. D.; Boghaert, E. R.; Ackler, S. L.; Catron, N. D.; Chen, J.; Dayton, B. D.; Ding, H.; Enschede, S. H.; Fairbrother, W. J.; Huang, D. C. S.; Hymowitz, S. G.; Jin, S.; Khaw, S. L.; Kovar, P. J.; Lam, L. T.; Lee, J.; Maecker, H. L.; Marsh, K. C.; Mason, K. D.; Mitten, M. J.; Nimmer, P. M.; Oleksijew, A.; Park, C. H.; Park, C.-M.; Phillips, D. C.; Roberts, A. W.; Sampath, D.; Seymour, J. F.; Smith, M. L.; Sullivan, G. M.; Tahir, S. K.; Tse, C.; Wendt, M. D.; Xiao, Y.; Xue, J. C.; Zhang, H.; Humerickhouse, R. A.; Rosenberg, S. H.; Elmore, S. W. ABT-199, a potent and selective BCL-2 inhibitor, achieves antitumor activity while sparing platelets. *Nat. Med. (N.Y., NY, U.S.)* **2013**, *19*, 202–208.

(7) Shin, W.-H.; Kumazawa, K.; Imai, K.; Hirokawa, T.; Kihara, D. Current Challenges and Opportunities in Designing Protein-Protein Interaction Targeted Drugs. *Adv. Appl. Bioinf. Chem.* **2020**, *13*, 11–25.

(8) Perkins, J. R.; Diboun, I.; Dessailly, B. H.; Lees, J. G.; Orenco, C. Transient Protein-Protein Interactions: Structural, Functional, and Network Properties. *Structure* **2010**, *18*, 1233–1243.

(9) Smith, M. C.; Gestwicki, J. E. Features of protein-protein interactions that translate into potent inhibitors: topology, surface area and affinity. *Expert Rev. Mol. Med.* **2012**, *14*, No. e16.

(10) Szklarczyk, D.; Kirsch, R.; Koutrouli, M.; Nastou, K.; Mehryary, F.; Hachilif, R.; Gable, A. L.; Fang, T.; Doncheva, N. T.; Pyysalo, S.; Bork, P.; Jensen, L. J.; von Mering, C. The STRING database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Res.* **2023**, *51*, D638–D646.

(11) Mugur, R.; Smitha, P. S.; Pallavi, M. S. Predicting the Functions of Unknown Protein by Analyzing Known Protein Interaction: A survey. *Biomed. Pharmacol. J.* **2018**, *11*, 1707–1715.

(12) Labbé, C. M.; Kuenemann, M. A.; Zarzycka, B.; Vriend, G.; Nicolaes, G. A.; Lagorce, D.; Miteva, M. A.; Villoutreix, B. O.; Sperandio, O. iPPI-DB: an online database of modulators of protein-protein interactions. *Nucleic Acids Res.* **2016**, *44*, D542–D547.

(13) Wallace, A. C.; Laskowski, R. A.; Thornton, J. M. Derivation of 3D coordinate templates for searching structural databases: Application to ser-His-Asp catalytic triads in the serine proteinases and lipases. *Protein Sci.* **1996**, *5*, 1001–1013.

(14) Xue, L. C.; Dobbs, D.; Bonvin, A. M.; Honavar, V. Computational prediction of protein interfaces: A review of data driven methods. *FEBS Lett.* **2015**, *589*, 3516–3526.

(15) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.

(16) Shulman-Peleg, A.; Mintz, S.; Nussinov, R.; Wolfson, H. J. *Protein-Protein Interfaces: Recognition of Similar Spatial and Chemical Organizations. Algorithms in Bioinformatics*; Springer: Berlin, Heidelberg, 2004; pp 194–205.

(17) Shulman-Peleg, A.; Nussinov, R.; Wolfson, H. J. SiteEngines: recognition and comparison of binding sites and protein-protein interfaces. *Nucleic Acids Res.* **2005**, *33*, W337–W341.

(18) Gao, M.; Skolnick, J. iAlign: a method for the structural comparison of protein-protein interfaces. *Bioinformatics* **2010**, *26*, 2259–2265.

- (19) Zhang, Y.; Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* **2005**, *33*, 2302–2309.
- (20) Connolly, M. L. Measurement of Protein Surface Shape by Solid Angles. *J. Mol. Graph.* **1986**, *4*, 3–6.
- (21) Duhovny, D.; Nussinov, R.; Wolfson, H. J. *Efficient Unbound Docking of Rigid Molecules. Algorithms in Bioinformatics*; Springer: Berlin, Heidelberg, 2002; pp 185–200.
- (22) Pulim, V.; Berger, B.; Bienkowska, J. Optimal contact map alignment of protein–protein interfaces. *Bioinformatics* **2008**, *24*, 2324–2328.
- (23) Smith, T.; Waterman, M. Identification of common molecular subsequences. *J. Mol. Biol.* **1981**, *147*, 195–197.
- (24) Cheng, S.; Zhang, Y.; Brooks, C. L. PCalign: a method to quantify physicochemical similarity of protein–protein interfaces. *BMC Bioinf.* **2015**, *16*, 33.
- (25) Cui, X.; Naveed, H.; Gao, X. Finding optimal interaction interface alignments between biological complexes. *Bioinformatics* **2015**, *31*, i133–i141.
- (26) Mirabello, C.; Wallner, B. Topology independent structural matching discovers novel templates for protein interfaces. *Bioinformatics* **2018**, *34*, 1787–1794.
- (27) Budowski-Tal, I.; Kolodny, R.; Mandel-Gutfreund, Y. A Novel Geometry-Based Approach to Infer Protein Interface Similarity. *Sci. Rep.* **2018**, *8*, 8192.
- (28) Reim, T.; Ehrh, C.; Graef, J.; Günther, S.; Meents, A.; Rarey, M. SiteMine: Large-scale binding site similarity searching in protein structure databases. *Arch. Pharm.* **2024**, No. e2300661.
- (29) Graef, J.; Ehrh, C.; Diedrich, K.; Poppinga, M.; Ritter, N.; Rarey, M. Searching Geometric Patterns in Protein Binding Sites and Their Application to Data Mining in Protein Kinase Structures. *J. Med. Chem.* **2022**, *65*, 1384–1395.
- (30) Bietz, S.; Urbaczek, S.; Schulz, B.; Rarey, M. Protoss: A Holistic Approach to Predict Tautomers and Protonation States in Protein–Ligand Complexes. *J. Cheminf.* **2014**, *6*, 12.
- (31) Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr., Sect. A: Found. Adv.* **1976**, *32*, 922–923.
- (32) Kabsch, W. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr., Sect. A: Found. Adv.* **1978**, *34*, 827–828.
- (33) Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380.
- (34) Calenoff, E. Interplaying Factors That Effect Multiple Sclerosis Causation and Sustainance. *ISRN Neurol.* **2012**, *2012*, 851541.
- (35) Blanco, J. L.; Rai, P. K. *Nanoflann: A C++ Header-Only Fork of FLANN, a Library for Nearest Neighbor (NN) with KD-Trees*, 2014. <https://github.com/jlblancoc/nanoflanwebn>.
- (36) Barradas-Bautista, D.; Almajed, A.; Oliva, R.; Kalnis, P.; Cavallo, L. Improving classification of correct and incorrect protein–protein docking models by augmenting the training set. *Bioinform. Adv.* **2023**, *3*, vbad012.
- (37) Barradas-Bautista, D.; Oliva, R.; Cavallo, L. *A Protein–Protein Docking Decoys Set from Three Different Rigid Body Methods*; Zendo, 2020.
- (38) Méndez, R.; Leplae, R.; De Maria, L.; Wodak, S. J. Assessment of blind predictions of protein–protein interactions: Current status of docking methods. *Proteins: Struct., Funct., Bioinf.* **2003**, *52*, 51–67.
- (39) Keskin, O.; Tsai, C.-J.; Wolfson, H.; Nussinov, R. A new, structurally nonredundant, diverse data set of protein–protein interfaces and its implications. *Protein Sci.* **2004**, *13*, 1043–1055.
- (40) Vreven, T.; Moal, I. H.; Vangone, A.; Pierce, B. G.; Kastriitis, P. L.; Torchala, M.; Chaleil, R.; Jiménez-García, B.; Bates, P. A.; Fernandez-Recio, J.; Bonvin, A. M.; Weng, Z. Updates to the Integrated Protein–Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2. *J. Mol. Biol.* **2015**, *427*, 3031–3041.
- (41) Lu, H.; Lu, L.; Skolnick, J. Development of Unified Statistical Potentials Describing Protein–Protein Interactions. *Biophys. J.* **2003**, *84*, 1895–1901.
- (42) Chen, H.; Skolnick, J. M-TASSER: An Algorithm for Protein Quaternary Structure Prediction. *Biophys. J.* **2008**, *94*, 918–928.
- (43) Duarte, J. M.; Srebnik, A.; Schärer, M. A.; Capitani, G. Protein interface classification by evolutionary analysis. *BMC Bioinf.* **2012**, *13*, 334.
- (44) Steinegger, M.; Söding, J. Clustering huge protein sequence sets in linear time. *Nat. Commun.* **2018**, *9*, 2542.
- (45) van Kempen, M.; Kim, S. S.; Tumescheit, C.; Mirdita, M.; Lee, J.; Gilchrist, C. L. M.; Söding, J.; Steinegger, M. Fast and accurate protein structure search with Foldseek. *Nat. Biotechnol.* **2023**, *42*, 243–246.
- (46) Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. UCSF Chimera—A visualization system for exploratory research and analysis. *J. Comput. Chem.* **2004**, *25*, 1605–1612.
- (47) Meng, E. C.; Pettersen, E. F.; Couch, G. S.; Huang, C. C.; Ferrin, T. E. Tools for integrated sequence–structure analysis with UCSF Chimera. *BMC Bioinf.* **2006**, *7*, 339.
- (48) Graef, J.; Ehrh, C.; Reim, T.; Rarey, M. *Optimization and Evaluation Datasets for PiMine*, 2024.
- (49) Agirre, J.; Atanasova, M.; Bagdonas, H.; Ballard, C. B.; Baslé, A.; Beilstein-Edmands, J.; Borges, R. J.; Brown, D. G.; Burgos-Mármol, J. J.; Berrisford, J. M.; Bond, P. S.; Caballero, I.; Catapano, L.; Chojnowski, G.; Cook, A. G.; Cowtan, K. D.; Croll, T. I.; Debreczeni, J. E.; Devenish, N. E.; Dodson, E. J.; Drevon, T. R.; Emsley, P.; Evans, G.; Evans, P. R.; Fando, M.; Foadi, J.; Fuentes-Montero, L.; Garman, E. F.; Gerstel, M.; Gildea, R. J.; Hatti, K.; Hekkelman, M. L.; Heuser, P.; Hoh, S. W.; Hough, M. A.; Jenkins, H. T.; Jiménez, E.; Joosten, R. P.; Keegan, R. M.; Keep, N.; Krissinel, E. B.; Kolenko, P.; Kovalevskiy, O.; Lamzin, V. S.; Lawson, D. M.; Lebedev, A. A.; Leslie, A. G. W.; Lohkamp, B.; Long, F.; Malý, M.; McCoy, A. J.; McNicholas, S. J.; Medina, A.; Millán, C.; Murray, J. W.; Murshudov, G. N.; Nicholls, R. A.; Noble, M. E. M.; Oeffner, R.; Pannu, N. S.; Parkhurst, J. M.; Pearce, N.; Pereira, J.; Perrakis, A.; Powell, H. R.; Read, R. J.; Rigden, D. J.; Rochira, W.; Sammito, M.; Sánchez Rodríguez, F.; Sheldrick, G. M.; Shelley, K. L.; Simkovic, F.; Simpkin, A. J.; Skubak, P.; Sobolev, E.; Steiner, R. A.; Stevenson, K.; Tews, I.; Thomas, J. M. H.; Thorn, A.; Valls, J. T.; Uski, V.; Usón, I.; Vagin, A.; Velankar, S.; Vollmar, M.; Walden, H.; Waterman, D.; Wilson, K. S.; Winn, M. D.; Winter, G.; Wojdyr, M.; Yamashita, K. The CCP4 suite: integrative software for macromolecular crystallography. *Acta Crystallogr., Sect. D: Struct. Biol.* **2023**, *79*, 449–461.
- (50) Kellenberger, E.; Schalon, C.; Rognan, D. How to Measure the Similarity Between Protein Ligand-Binding Sites? *Curr. Comput.-Aided Drug Des.* **2008**, *4*, 209–220.
- (51) Keskin, O.; Nussinov, R. Similar Binding Sites and Different Partners: Implications to Shared Proteins in Cellular Pathways. *Structure* **2007**, *15*, 341–354.
- (52) Kim, W. K.; Henschel, A.; Winter, C.; Schroeder, M. The Many Faces of Protein–Protein Interactions: A Compendium of Interface Geometry. *PLoS Comput. Biol.* **2006**, *2*, No. e124.
- (53) Schrödinger, L.; DeLano, W. PyMOL. <http://www.pymol.org/pymol,version2.4.0,2020-05-2web0>.
- (54) Keren-Kaplan, T.; Attali, I.; Estrin, M.; Kuo, L. S.; Farkash, E.; Jerabek-Willemsen, M.; Blutraich, N.; Artzi, S.; Peri, A.; Freed, E. O.; Wolfson, H. J.; Prag, G. Structure-based in silico identification of ubiquitin-binding domains provides insights into the ALIX-V:ubiquitin complex and retrovirus budding. *EMBO J.* **2013**, *32*, 538–551.
- (55) Hurley, J. H.; Lee, S.; Prag, G. Ubiquitin-binding domains. *Biochem. J.* **2006**, *399*, 361–372.
- (56) González-Motos, V.; Kropp, K. A.; Viejo-Borbolla, A. Chemokine binding proteins: An immunomodulatory strategy going viral. *Cytokine Growth Factor Rev.* **2016**, *30*, 71–80.
- (57) Li, T.; Robert, E. I.; van Breugel, P. C.; Strubin, M.; Zheng, N. A promiscuous  $\alpha$ -helical motif anchors viral hijackers and substrate

receptors to the CUL4–DDB1 ubiquitin ligase machinery. *Nat. Struct. Mol. Biol.* **2010**, *17*, 105–111.

(58) Bowden, T. A.; Aricescu, A. R.; Gilbert, R. J. C.; Grimes, J. M.; Jones, E. Y.; Stuart, D. I. Structural basis of Nipah and Hendra virus attachment to their cell-surface receptor ephrin-B2. *Nat. Struct. Mol. Biol.* **2008**, *15*, 567–572.

(59) Singla, N.; Goldgur, Y.; Xu, K.; Paavilainen, S.; Nikolov, D. B.; Himanen, J. P. Crystal structure of the ligand-binding domain of the promiscuous EphA4 receptor reveals two distinct conformations. *Biochem. Biophys. Res. Commun.* **2010**, *399*, 555–559.

(60) Dai, D.; Huang, Q.; Nussinov, R.; Ma, B. Promiscuous and specific recognition among ephrins and Eph receptors. *Biochim. Biophys. Acta, Proteins Proteomics* **2014**, *1844*, 1729–1740.

(61) Inano, S.; Sato, K.; Katsuki, Y.; Kobayashi, W.; Tanaka, H.; Nakajima, K.; Nakada, S.; Miyoshi, H.; Knies, K.; Takaori-Kondo, A.; Schindler, D.; Ishiai, M.; Kurumizaka, H.; Takata, M. RFWD3-Mediated Ubiquitination Promotes Timely Removal of Both RPA and RAD51 from DNA Damage Sites to Facilitate Homologous Recombination. *Mol. Cell* **2017**, *66*, 622–634.e8.

(62) Knies, K.; Inano, S.; Ramírez, M. J.; Ishiai, M.; Surrallés, J.; Takata, M.; Schindler, D. Biallelic mutations in the ubiquitin ligase RFWD3 cause Fanconi anemia. *J. Clin. Invest.* **2017**, *127*, 3013–3027.

(63) Porollo, A.; Meller, J. Prediction-based fingerprints of protein–protein interactions. *Proteins: Struct., Funct., Bioinf.* **2007**, *66*, 630–645.

(64) Burroughs, A. M. The natural history of ubiquitin and ubiquitin-related domains. *Front. Biosci.* **2012**, *17*, 1433.

# Supporting Information

## Database-Driven Identification of Structurally Similar Protein-Protein Interfaces

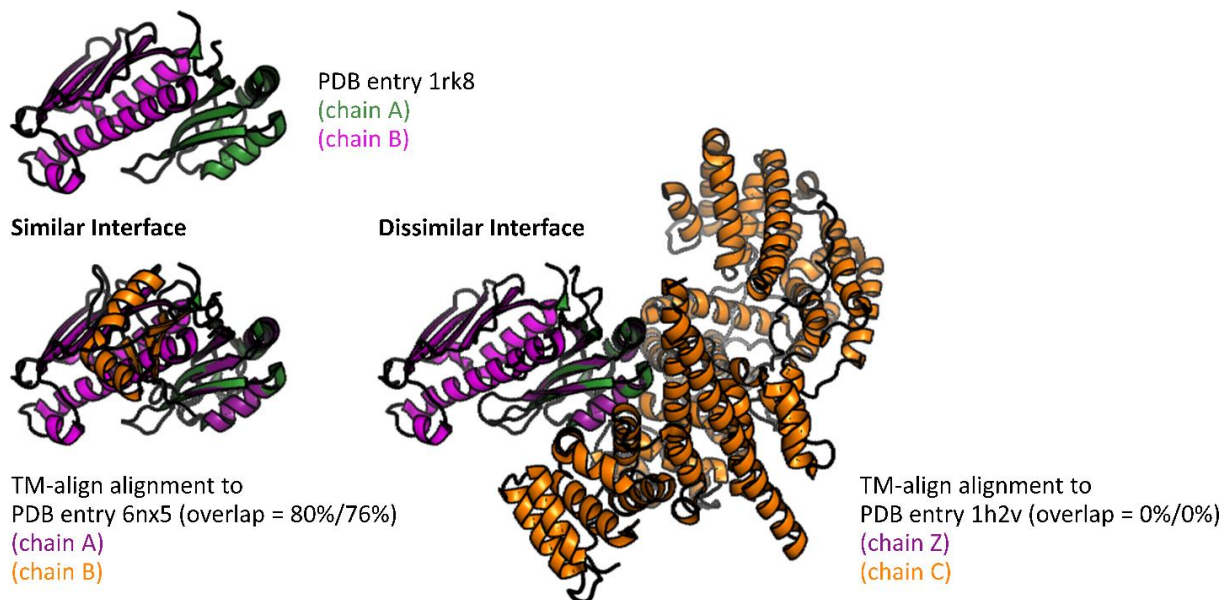
Joel Graef, Christiane Ehrt\*, Thorben Reim, Matthias Rarey\*

Universität Hamburg, ZBH - Center for Bioinformatics, Albert-Einstein-Ring 8-10, 22761 Hamburg, Germany

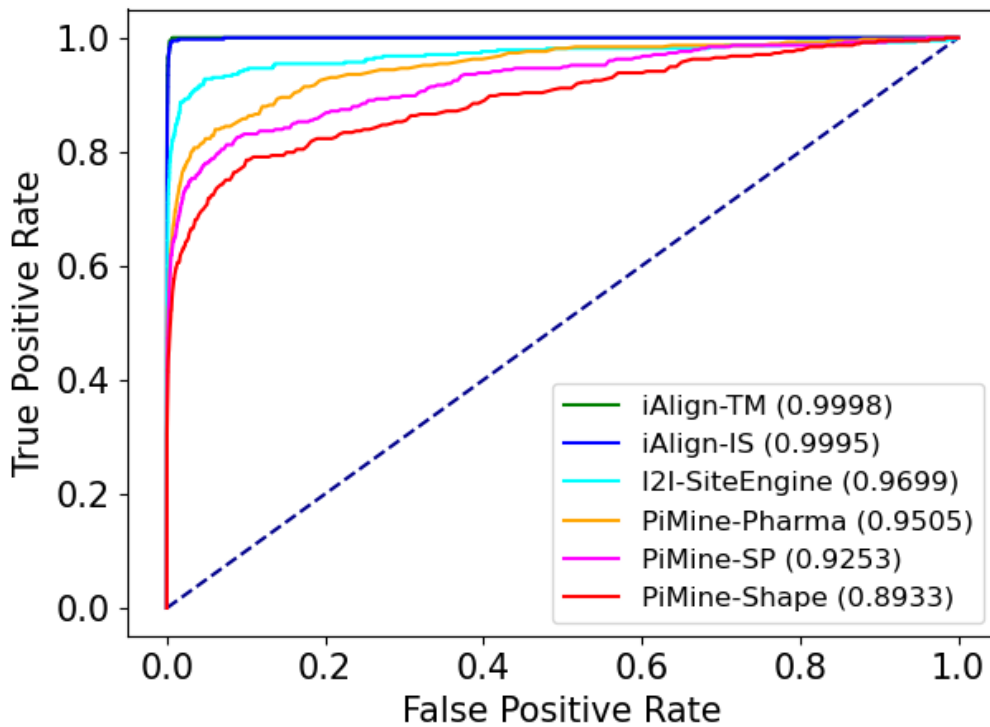
E-mail: christiane.ehrt@uni-hamburg.de, matthias.rarey@uni-hamburg.de

### Table of Contents

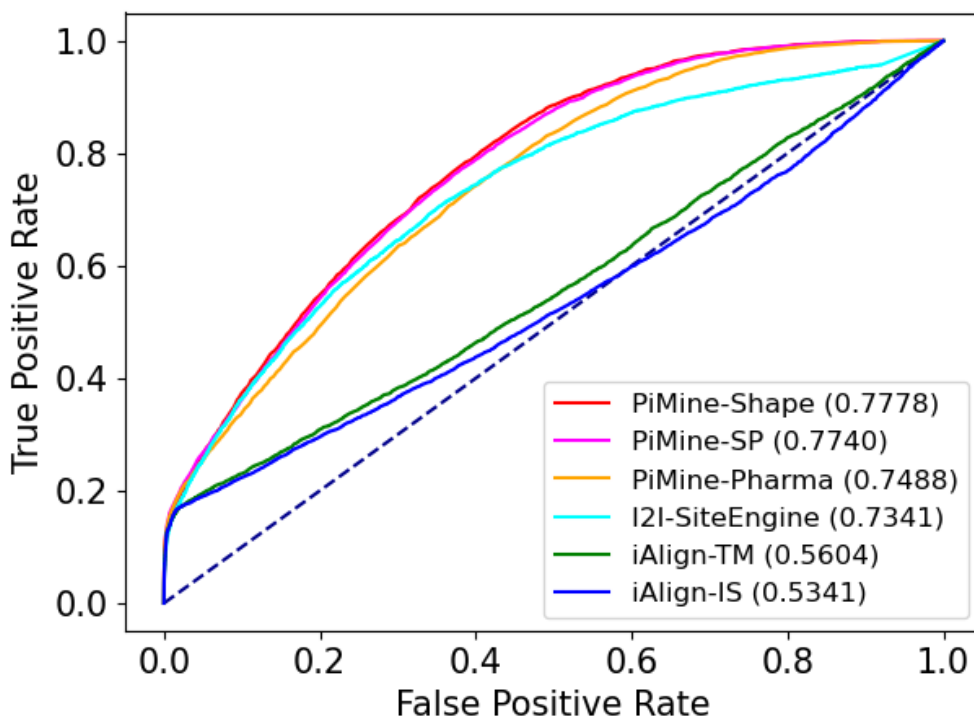
Figure S1.....	S2
Figure S2.....	S3
Figure S3.....	S3
Figure S4.....	S4
Figure S5.....	S4
Figure S6.....	S5
Figure S7.....	S5
Figure S8.....	S6
Figure S9.....	S6
Figure S10.....	S7
Figure S11.....	S7
Figure S12.....	S8
Figure S13.....	S9
Figure S14.....	S10
Figure S15.....	S11
Figure S16.....	S12
Table S1.....	S12
Table S2.....	S12
Table S3.....	S13
Paragraph S1.....	S14
Paragraph S2.....	S14
References.....	S15



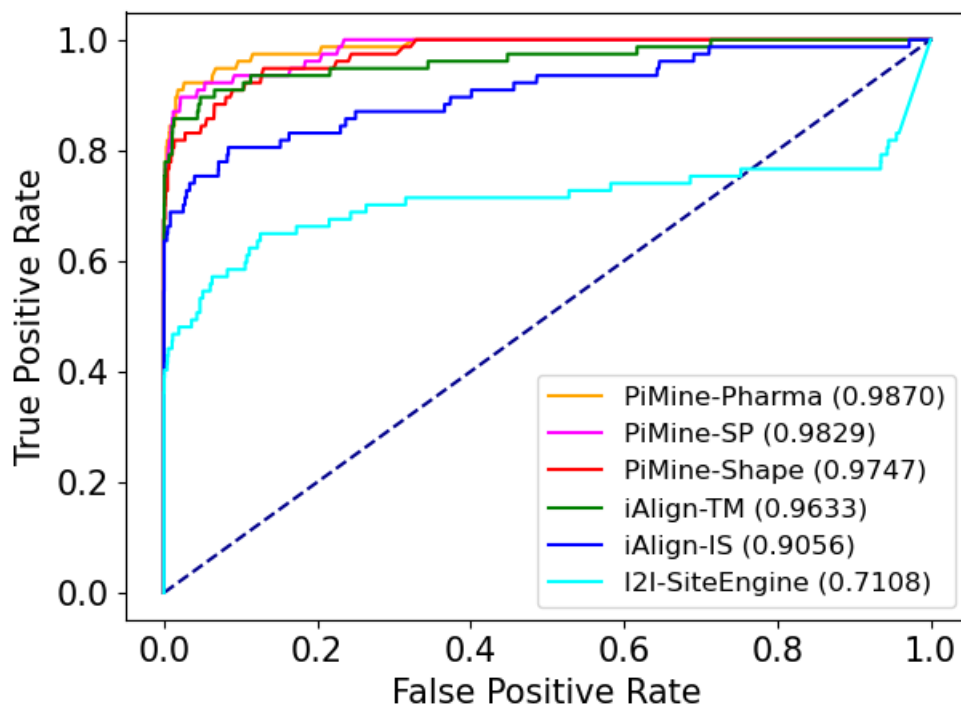
**Figure S1.** Examples for similar and dissimilar interfaces in the *PiMineSet*. The interface of chains A and B of the query PDB entry 1rk8 is similar to the interface between chains A and B of PDB entry 6nx5. The overlap between the interface residues of the interfaces based on the TM-align alignment is 80% and 76% (left). In contrast, there is no overlap with chains Z and C of PDB entry 1h2v. So, the latter interfaces do not have any similarity based on the TM-align alignment (right).



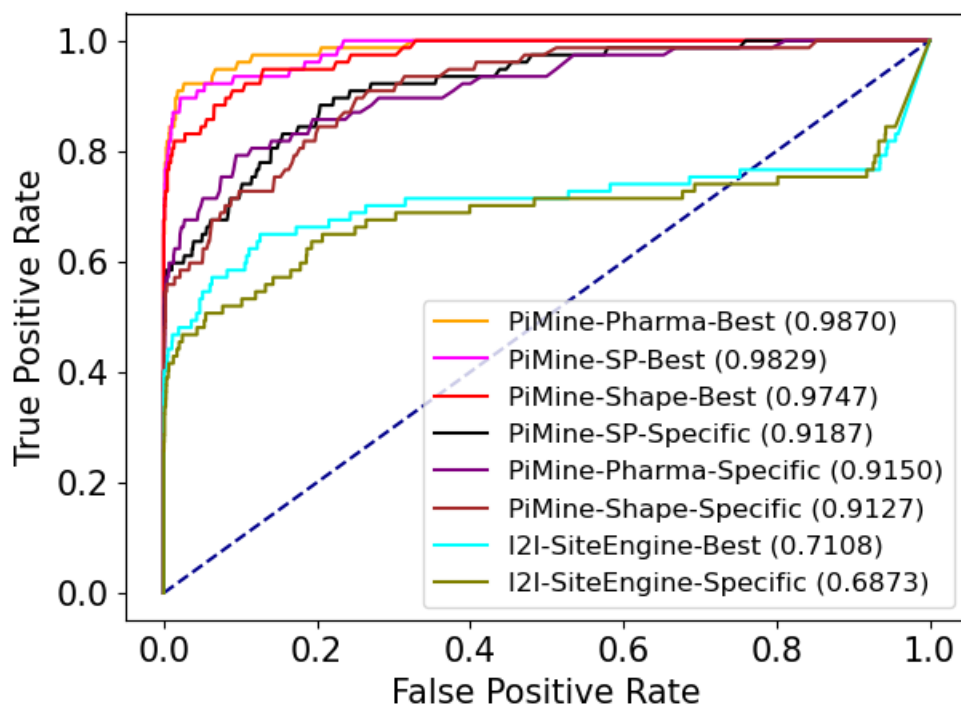
**Figure S2.** The ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (accuracy-optimized parameters) on the *Dimer597* set.



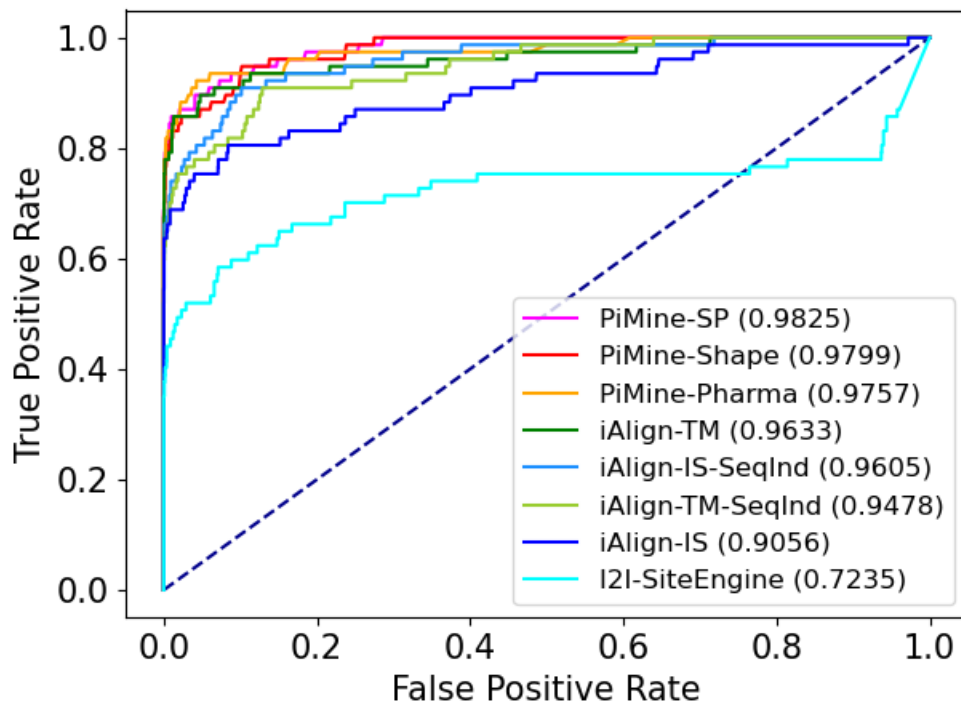
**Figure S3.** The ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (accuracy-optimized parameters) on the *Keskin* set.



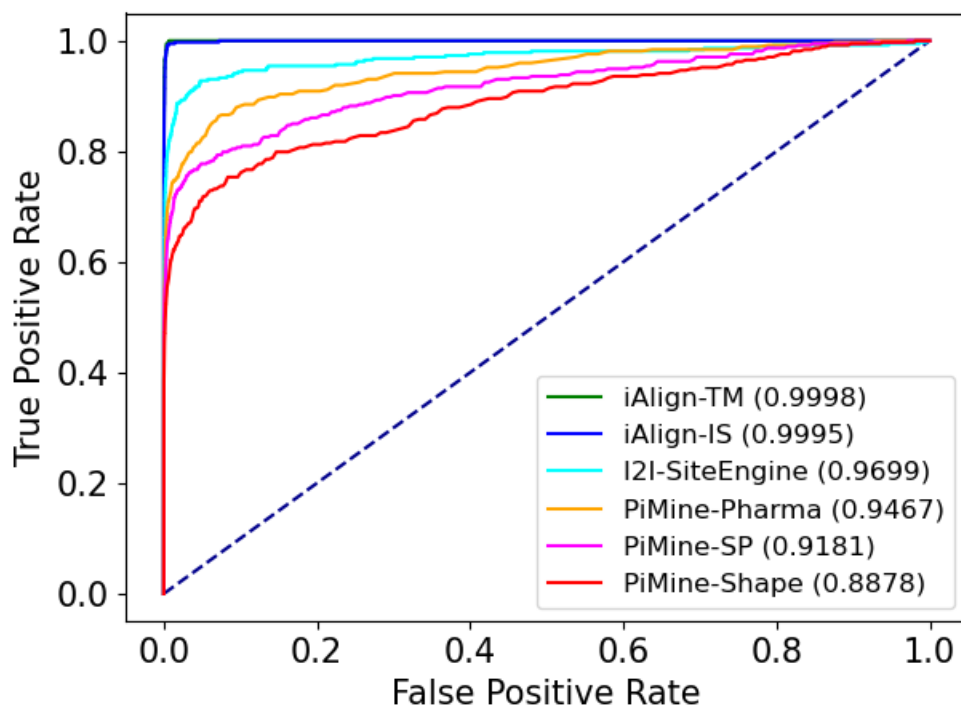
**Figure S4.** The ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (accuracy-optimized parameters) on the *PiMineSet*.



**Figure S5.** The ROC curves for predicting similar interfaces of sequentially and structurally similar chains using I2I-SiteEngine and PiMine (accuracy-optimized parameters) on the *PiMineSet* excluding the interfaces of related single chains. The sequentially and structurally similar chains were excluded from scoring. This scenario is not realizable with iAlign; therefore, the ROC curves for iAlign are missing.

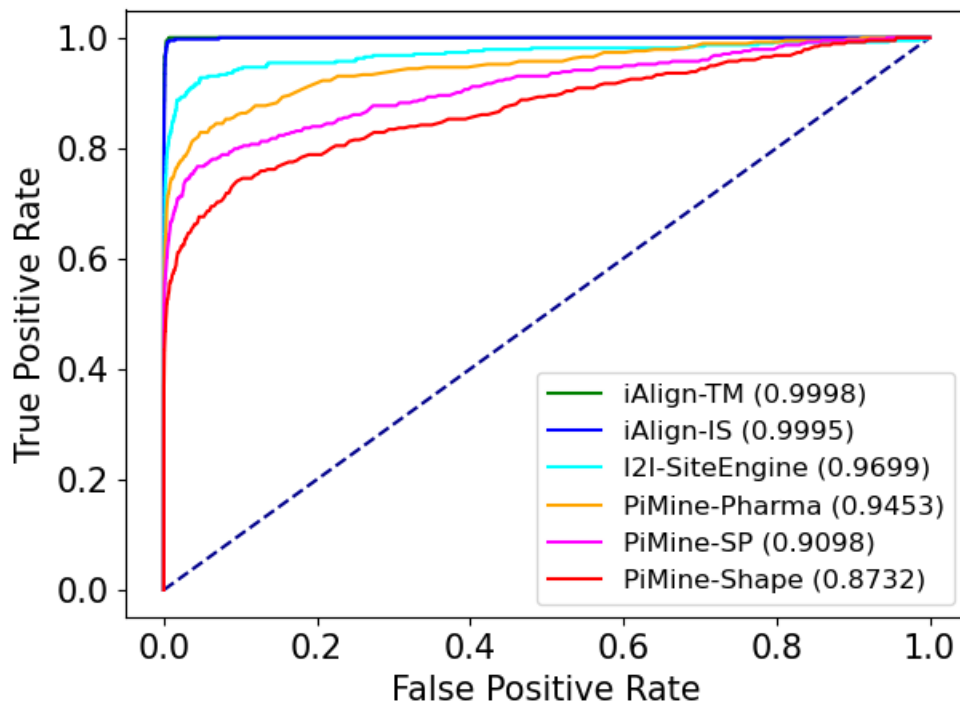


**Figure S6.** The ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters) on the *PiMineSet*. The performance of iAlign in sequence-independent (SeqInd) mode is given in light green (TM-score) and light blue (IS-score).

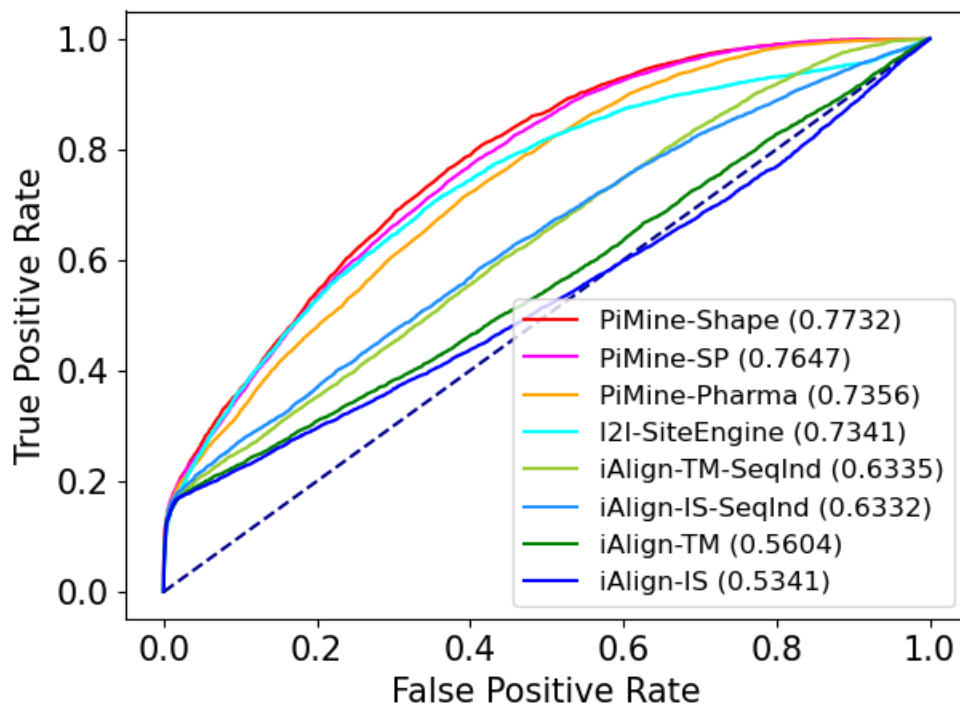


**Figure S7.** The ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters and scoring using both interfaces of the PPIs) on the *Dimer597* set.

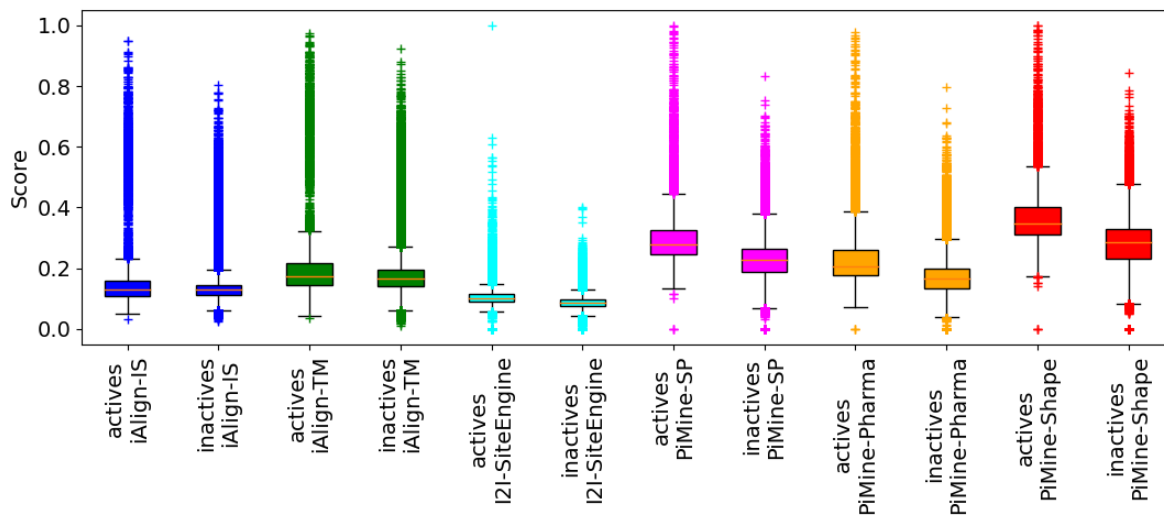




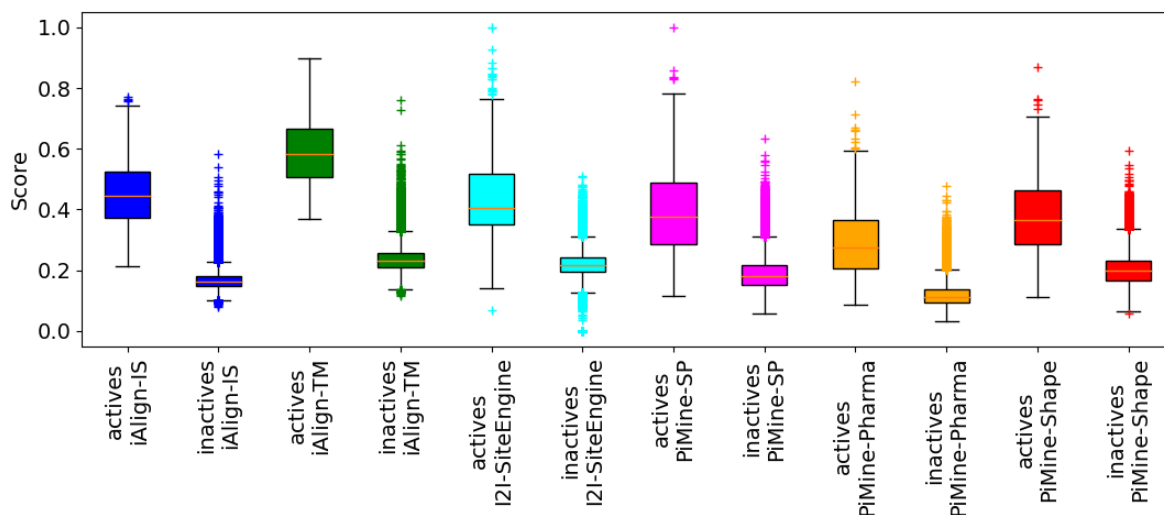
**Figure S8.** The ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (accuracy-optimized parameters and scoring using both interfaces of the PPIs) on the *Dimer597* set.



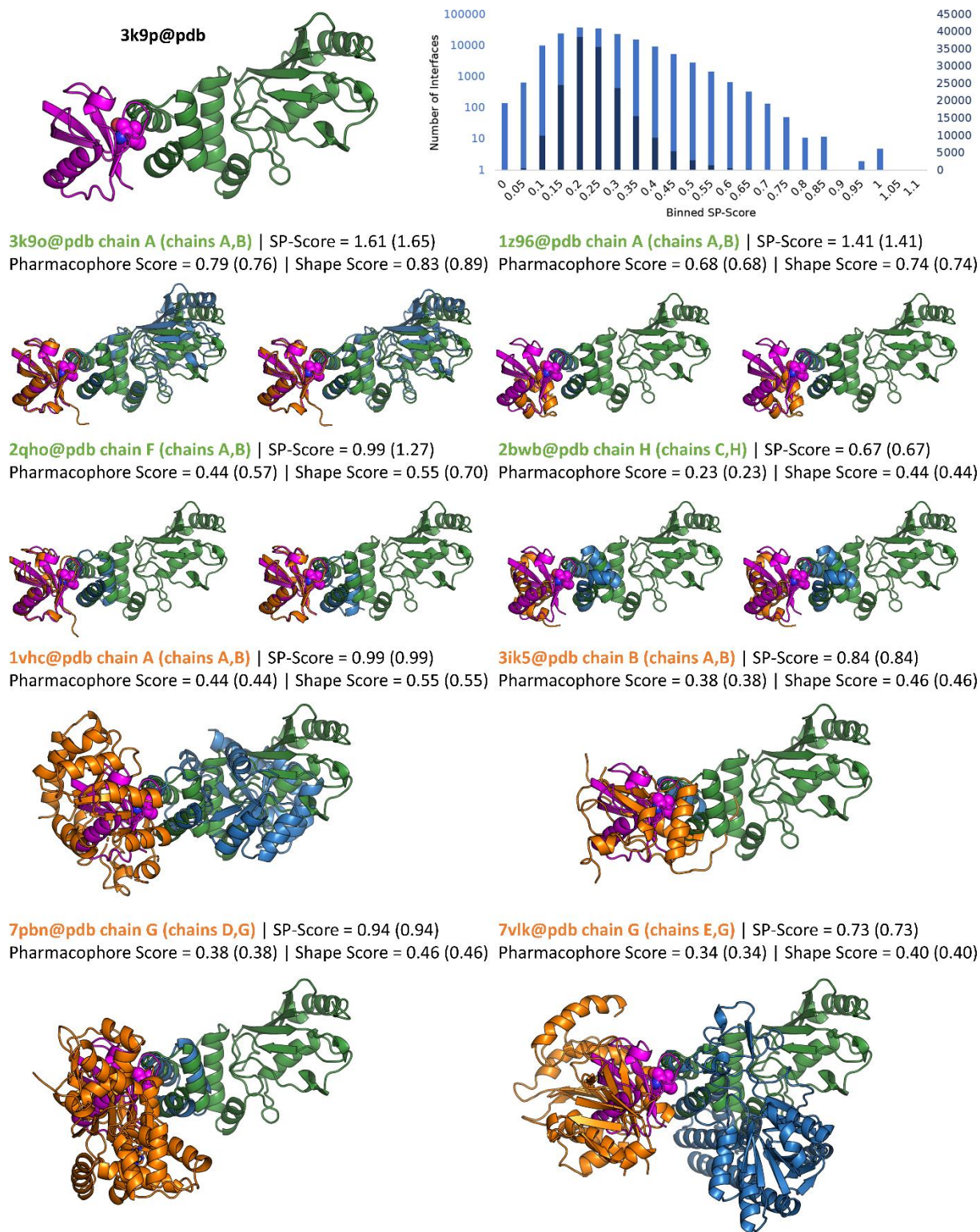
**Figure S9.** The ROC curves for predicting related interfaces using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters) on the *Keskin* set. The performance of iAlign in sequence-independent (SeqInd) mode is given in light green (TM-score) and light blue (IS-score).



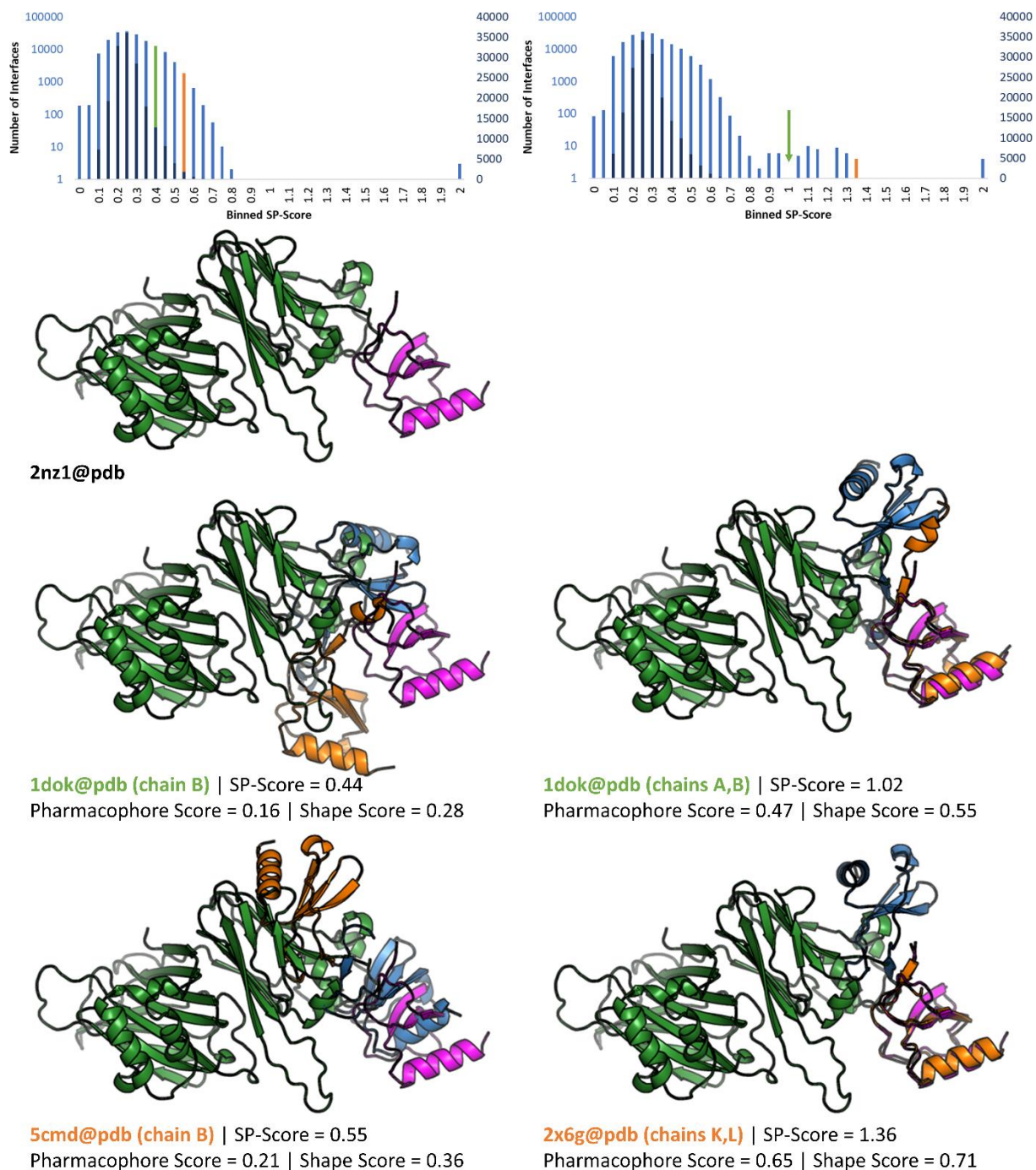
**Figure S10.** Box plots showing the score distributions of actives (similar interface pairs) and inactives (dissimilar interface pairs) including the outliers for the *Keskin* set using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters).



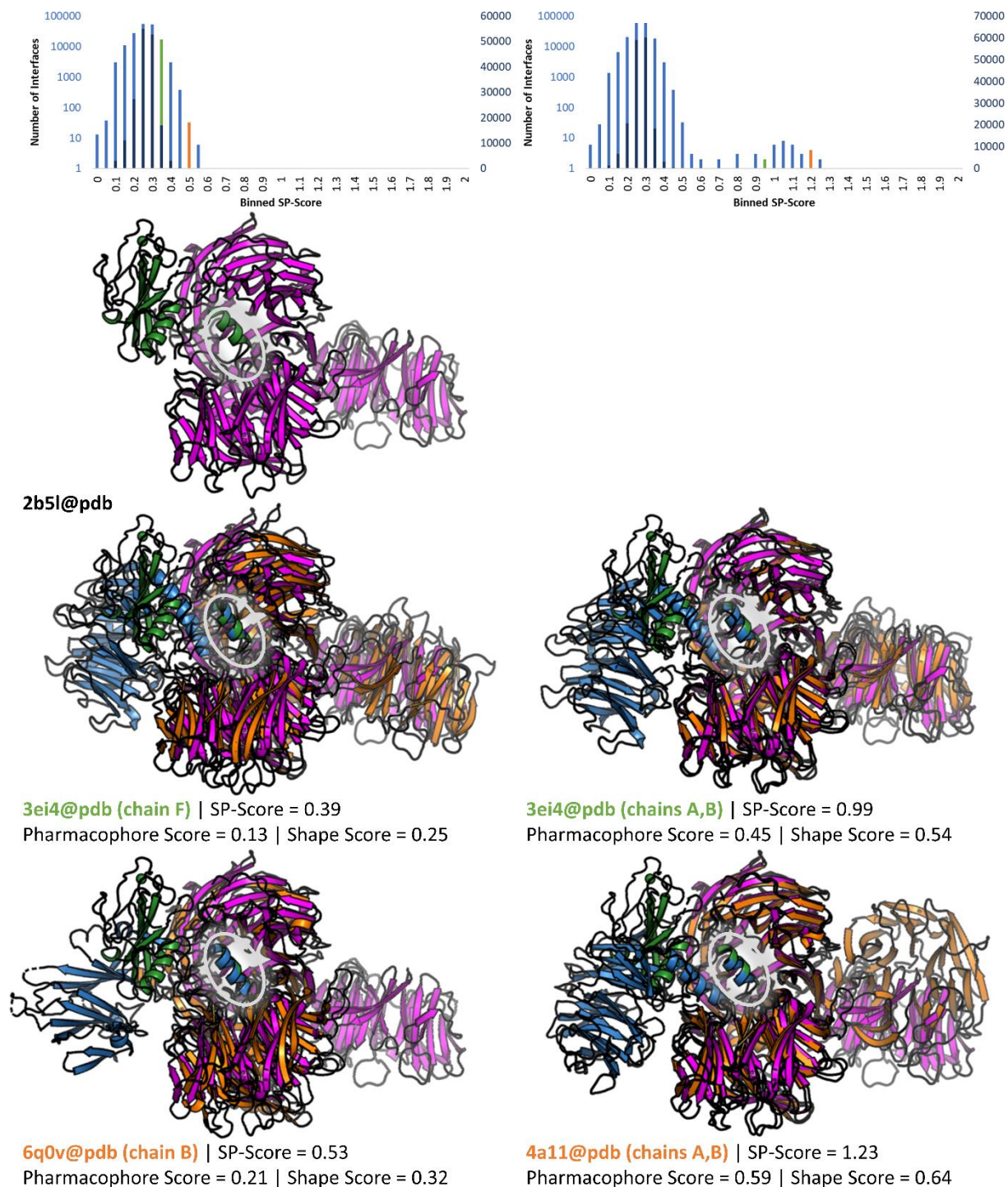
**Figure S11.** Box plots showing the score distributions of actives (similar interface pairs) and inactives (dissimilar interface pairs) for the *Dimer597* set using the methods iAlign, I2I-SiteEngine, and PiMine (runtime-optimized parameters).



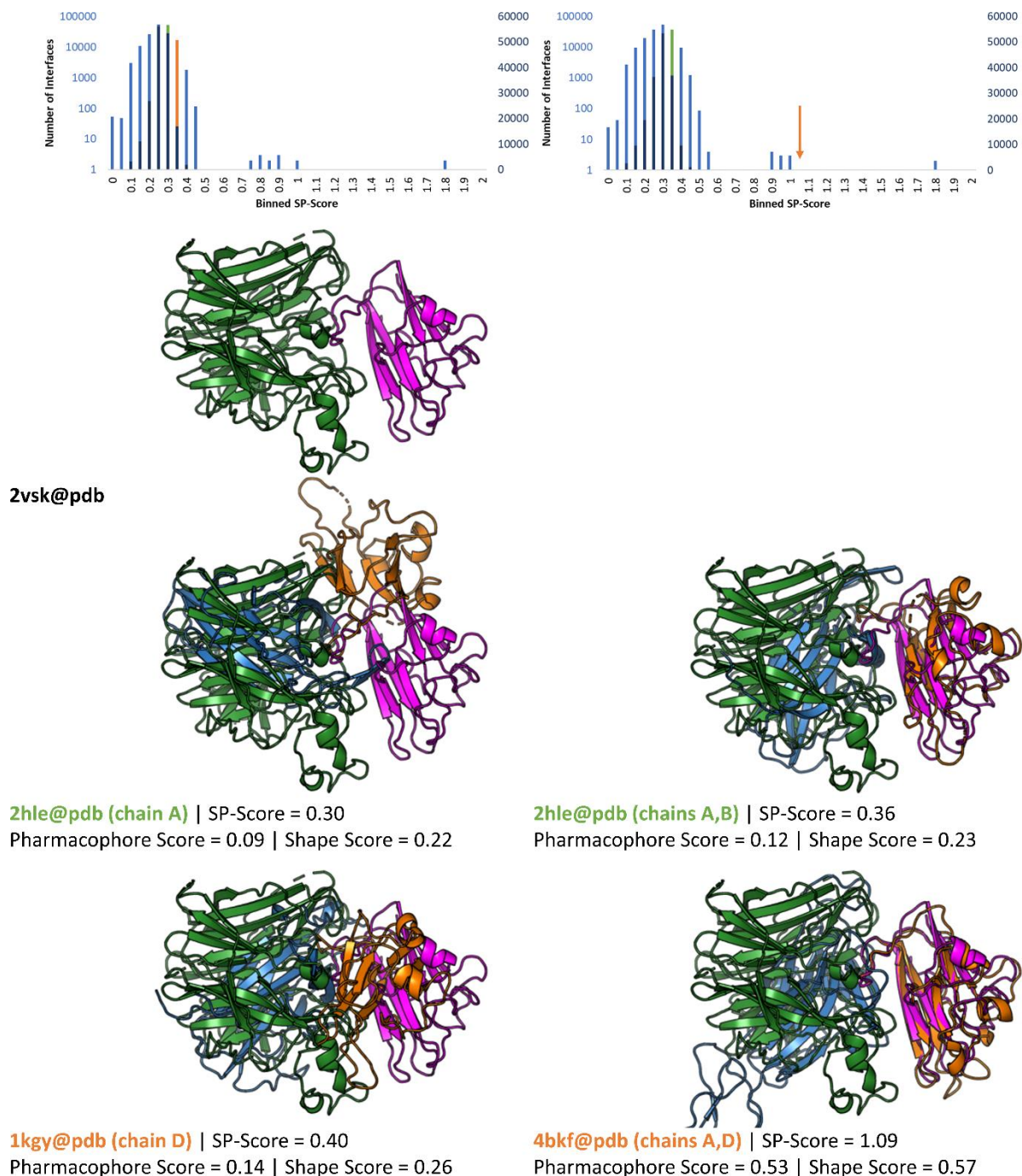
**Figure S12.** Matches found with a PiMine search for chain A of PDB entry 3k9p (ubiquitin-conjugating enzyme E2-K in complex with ubiquitin). Above, we see the query interface with chain A in green and its binding partner in magenta. On the right, the score distribution for the *RunTimeSet* with the added similar interfaces from the work of Keren-Kaplan and colleagues<sup>1</sup> is shown in the top right corner. Below, alignments of the similar interfaces reported by Keren-Kaplan et al. are shown using only chain A of the query interface (matching chains are depicted in blue) and using both interface chains (green font). The other matches show alignments for high-scoring hits additionally found with PiMine using chain A of the query interface (orange font).



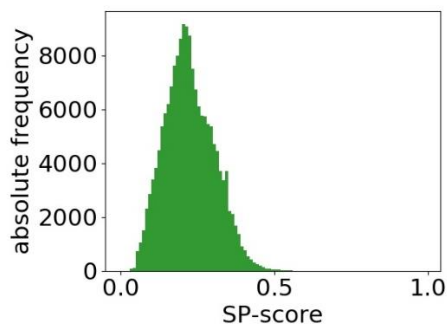
**Figure S13.** Matches found with a PiMine search for chain A of PDB entry 2nz1 (M3 protein of murid herpesvirus 4 in a complex with C-C motif chemokine 2) and interface chains A (green) and D (magenta). On the top, the score distributions for the *RunTimeSet* with the added similar interfaces from the work of Cheng and colleagues<sup>2</sup> is shown for a single-chain interface (left) and the interfaces of both chains (right). Below, alignments of the similar interfaces reported by Cheng et al. are shown using only chain A of the query interface (left) and using both interface chains (right). The matches at the bottom show alignments for high-scoring hits additionally found with PiMine using chain A of the query interface (left) and using both chain interfaces (right). The scores for these alignments are highlighted in green (reported example) and orange (high-scoring hit) in the distribution plots above.



**Figure S14.** Matches found with a PiMine search for chain C of PDB entry 2b5l (Simian virus 5 non-structural protein V in a complex with damage-specific DNA-binding protein 1) and interface chains C (green) and A (magenta). On the top, the score distributions for the *RunTimeSet* with the added similar interfaces from the work of Cheng and colleagues<sup>2</sup> is shown for a single-chain interface (left) and the interfaces of both chains (right). Below, alignments of the similar interfaces reported by Cheng et al. are shown using only chain C of the query interface (left) and using both interface chains (right). The matches at the bottom show alignments for high-scoring hits additionally found with PiMine using chain C of the query interface (left) and using both chain interfaces (right). The common helical motif discussed in the main paper is highlighted by a grey circle. The scores for these alignments are highlighted in green (reported example) and orange (high-scoring hit) in the distribution plots above.



**Figure S15.** Matches found with a PiMine search for chain A of PDB entry 2vsk (Henda virus attachment protein glycoprotein G in a complex with ephrin-B2) and interface chains A (green) and B (magenta). On the top, the score distributions for the *RunTimeSet* with the added similar interfaces from the work of Cheng and colleagues<sup>2</sup> is shown for a single-chain interface (left) and the interfaces of both chains (right). Below, alignments of the similar interfaces reported by Cheng et al. are shown using only chain A of the query interface (left) and using both interface chains (right). The matches at the bottom show alignments for high-scoring hits additionally found with PiMine using chain A of the query interface (left) and using both chain interfaces (right). The scores for these alignments are highlighted in green (reported example) and orange (high-scoring hit) in the distribution plots above.



**Figure S16.** Score distribution for the PiMine search for similar interfaces to a predicted one of PDB entry 6cvz (human E3 ubiquitin-protein ligase RFWD3) in the *RunTimeSet*. The SP-scores were binned in 0.01 bins.

**Table S1.** PostgreSQL 14.6 custom parameters for PiMine runs on the SSD (on the *RunTimeSet*).

max_connections	20
shared_buffers	2 GB
effective_cache_size	8 GB
maintenance_work_mem	2 GB
checkpoint_completion_target	0.9
wal_buffers	16 MB
default_statistics_target	100
random_page_cost	1.1
effective_io_concurrency	200
work_mem	29127 kB
min_wal_size	100 MB
max_wal_size	2 GB
max_worker_processes	6
max_parallel_workers_per_gather	3
max_parallel_workers	6
max_parallel_maintenance_workers	3

**Table S2.** PostgreSQL 14.6 custom parameters for PiMine runs on the HDD (on the *RunTimeSet*).

max_connections	20
shared_buffers	2 GB
effective_cache_size	8 GB
maintenance_work_mem	2 GB
checkpoint_completion_target	0.9
wal_buffers	16 MB
default_statistics_target	100
random_page_cost	4
effective_io_concurrency	2
work_mem	29127 kB
min_wal_size	100 MB
max_wal_size	2 GB
max_worker_processes	6
max_parallel_workers_per_gather	3
max_parallel_workers	6
max_parallel_maintenance_workers	3

**Table S3.** Normalized enrichment factors of the methods iAlign, I2I-SiteEngine, and PiMine (scoring using both interfaces of the PPIs) on the *Dimer597* set.

Name	0.1%	0.5%	1%	2%	5%	10%	20%
iAlign-TM	0.97	0.98	1.0	1.0	1.0	1.0	1.0
iAlign-IS	0.95	0.98	0.99	1.0	1.0	1.0	1.0
I2I-SiteEngine	0.86	0.77	0.83	0.89	0.93	0.94	0.95
PiMine-SP (runtime-optimized)	0.86	0.60	0.66	0.71	0.77	0.80	0.84
PiMine-pharmacophore (runtime-optimized)	0.91	0.68	0.74	0.77	0.83	0.86	0.92
PiMine-shape (runtime-optimized)	0.78	0.52	0.56	0.61	0.68	0.74	0.79
PiMine-SP (accuracy-optimized)	0.88	0.62	0.68	0.73	0.78	0.81	0.86
PiMine-pharmacophore (accuracy-optimized)	0.88	0.68	0.72	0.75	0.82	0.88	0.91
PiMine-shape (accuracy-optimized)	0.80	0.55	0.60	0.64	0.71	0.76	0.81



## Paragraph S1. PiMine Interface Input

Interfaces can be either loaded from the database by giving the PDB code and the identifiers of two interacting chains, or by giving a complex structure in the PDB or mmCIF file formats. In the case of the latter, either the PPI chain identifiers or a file formatted as PDB or mmCIF containing only the interface atoms must be provided. Complex structures given by the user are automatically pre-processed in the same way as in the database creation step.

## Paragraph S2. External Benchmark Data Sets for Protein-Protein Interface Comparisons.

As an established protein-protein interface similarity data set, we used the so-called *Dimer597* set (<https://sites.gatech.edu/cssb/ialign/>) comprising 597 interfaces with 373 related pairs and 176,875 unrelated pairs and generated for the benchmarking of iAlign.<sup>3</sup>

The data set was created by first selecting dimers with chains of at least 200 residues and SCOP assignments, as the latter enable selecting biologically related protein domains.<sup>4</sup> Two protein-protein interfaces are considered as related if they share the same SCOP superfamily assignment and their overlap ratio is at least 30% based on the best-scored TM-align chain superposition, i.e., at least 30% of the total interface contacts have to overlap when comparing both interfaces. An overlapping contact is found if the distances between both contacts are below  $d = 1.5[\min(|\text{residues}_{\text{target}}|, |\text{residues}_{\text{query}}|)]^{0.3} + 3.5$  and  $< 8 \text{ \AA}$ .<sup>1</sup> Interfaces are regarded as unrelated if the chain pairs of the aligned protein-protein complexes have differing SCOP superfamily assignments, a contact overlap ration of zero, and if less than 15% of the interface residues of the smaller interface in terms of number of residues are aligned by TM-align.

To verify whether PiMine recognizes interface similarities across unrelated protein dimer structures, we used a data set developed by Keskin et al.<sup>5</sup> All available PDB structures (July 18th, 2002) were filtered for multimers. To this end, two residues are regarded as interacting if the distance between any two atoms between residues from two chains is less than the sum of their van der Waals radii plus  $0.5 \text{ \AA}$ . The authors removed all interfaces with less than ten interacting residues to exclude artificial crystal interfaces. Next, interfaces were extracted for the identified chain pairs by checking the distance between each  $C\alpha$  atom of the interacting residues and the surrounding  $C\alpha$  atoms. If this distance was less than  $6 \text{ \AA}$ , the corresponding residue was assigned to the interface. Next, these approx. 20,000 interfaces are compared with each other using geometric hashing. The matching pairs are clustered by a heuristic iterative procedure. In each iteration cycle, the similarity definition is gradually relaxed. Of all resulting clusters, the sequences were compared using CLUSTAL W<sup>6</sup> and the BLOSSUM90 substitution matrix<sup>7</sup> within their cluster. To eliminate redundancy, interfaces whose sequence similarity is higher as 50% to at least one other interface of the current cluster are removed. Clusters with less than five members were also removed (<https://web.archive.org/web/20200220060703/http://home.ku.edu.tr/~okeskin/INTERFACE/nonred-interface.list2>). We generated all pairs of interfaces from the data set interfaces. Pairs annotated with the same cluster number were used as similar pairs. Otherwise, the pairs were classified as dissimilar. The resulting data set, which we call *Keskin* set, contains 103 clusters with 4,876 related interface pairs and 176,627 unrelated interface pairs.

## References

- (1) Keren-Kaplan, T.; Attali, I.; Estrin, M.; Kuo, L. S.; Farkash, E.; Jerabek-Willemsen, M.; Blutraich, N.; Artzi, S.; Peri, A.; Freed, E. O.; Wolfson, H. J.; Prag, G. Structure-based in silico identification of ubiquitin-binding domains provides insights into the ALIX-V:ubiquitin complex and retrovirus budding. *EMBO J.* **2013**, *32*, 538–551.
- (2) Cheng, S.; Zhang, Y.; Brooks, C. L. PCalign: a method to quantify physicochemical similarity of protein-protein interfaces. *BMC Bioinf.* **2015**, *16*, 33
- (3) Gao, M.; Skolnick, J. iAlign: a method for the structural comparison of protein–protein interfaces. *Bioinformatics* **2010**, *26*, 2259-2265.
- (4) Murzin, A.; Brenner, S.; Hubbard, T.; Chothia, C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **1995**, *247*, 536-540.
- (5) Keskin, O.; Tsai, C.-J.; Wolfson, H.; Nussinov, R. A new, structurally nonredundant, diverse data set of protein–protein interfaces and its implications. *Protein Sci.* **2004**, *13*, 1043-1055.
- (6) Thompson, J. D.; Higgins, D. G.; Gibson, T. J. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **1994**, *22*, 4673-4680.
- (7) Henikoff, S.; Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U. S. A.* **1992**, *89*, 10915-10919.







## F.7 SiteMine: Large Scale Binding Site Similarity Searching in Protein Structure Databases

- [D7] Reim, T., Ehrt, C., **Graef, J.**, Günther, S., Meents, A., Rarey, M., SiteMine: Large Scale Binding Site Similarity Searching in Protein Structure Databases. *Arch. Pharm.* (2024). DOI: 10.1002/ardp.202300661.

Nachdruck mit Erlaubnis von [D7]. Dieser Artikel darf für nicht-kommerzielle Zwecke gemäß den Wiley Terms and Conditions für die Selbstarchivierung verwendet werden.

Die angehängte *Supporting Information* wurde auf den für diese Arbeit relevanten Teil gekürzt. Eine ungekürzte Fassung ist unter [D7] verfügbar.

# SiteMine: Large-scale binding site similarity searching in protein structure databases

Thorben Reim<sup>1</sup>  | Christiane Ehart<sup>1</sup>  | Joel Graef<sup>1</sup>  | Sebastian Günther<sup>2</sup>  | Alke Meents<sup>2</sup>  | Matthias Rarey<sup>1</sup> 

<sup>1</sup>ZBH - Center for Bioinformatics, Universität Hamburg, Hamburg, Germany

<sup>2</sup>Center for Free-Electron Laser Science CFEL, Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany

## Correspondence

Matthias Rarey, Universität Hamburg, ZBH - Center for Bioinformatics, Albert-Einstein-Ring 8-10, Hamburg 22761, Germany.  
Email: [matthias.rarey@uni-hamburg.de](mailto:matthias.rarey@uni-hamburg.de)

## Funding information

Data Science in Hamburg - Helmholtz Graduate School for the Structure of Matter, Grant/Award Number: HIDSS-0002; Helmholtz Association, Grant/Award Numbers: FISCOV, SFRagX, HIR3X(InternLabs-0011)

## Abstract

Drug discovery and design challenges, such as drug repurposing, analyzing protein–ligand and protein–protein complexes, ligand promiscuity studies, or function prediction, can be addressed by protein binding site similarity analysis. Although numerous tools exist, they all have individual strengths and drawbacks with regard to run time, provision of structure superpositions, and applicability to diverse application domains. Here, we introduce SiteMine, an all-in-one database-driven, alignment-providing binding site similarity search tool to tackle the most pressing challenges of binding site comparison. The performance of SiteMine is evaluated on the ProSPECCTs benchmark, showing a promising performance on most of the data sets. The method performs convincingly regarding all quality criteria for reliable binding site comparison, offering a novel state-of-the-art approach for structure-based molecular design based on binding site comparisons. In a SiteMine showcase, we discuss the high structural similarity between cathepsin L and calpain 1 binding sites and give an outlook on the impact of this finding on structure-based drug design. SiteMine is available at <https://uhh.de/naomi>.

## KEYWORDS

binding site comparison, cathepsin L, drug repurposing, off-target prediction, structure-based drug design

## 1 | INTRODUCTION

The steadily growing number of experimentally solved and predicted protein structures and their availability in the Protein Data Bank (PDB),<sup>[1]</sup> SWISS-MODEL,<sup>[2]</sup> and AlphaFold Protein Structure Database<sup>[3]</sup> lay the grounds for data-driven approaches in structure-based drug design (SBDD).<sup>[4,5]</sup> There are many application areas for searching for similar protein binding sites. These include function and off-target prediction, protein classification, drug repurposing, and polypharmacology prediction (one drug addressing multiple targets).

Many tools have already been developed to predict and compare binding sites. Some overviews are provided elsewhere (protein pocket prediction,<sup>[6–10]</sup> binding site comparisons<sup>[11,12]</sup>). A recent extensive review by Eguida and Rognan<sup>[13]</sup> gives insights into state-of-the-art binding site analyses. Further binding site comparison tools were published recently (Table 1).

Binding site comparison tools can be divided into different groups by the output they produce (only similarity scores or also structure alignments), the binding site modeling (residue-, surface-, or interaction-based), and the used data structure for similarity calculation (e.g., graphs,

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2024 The Authors. *Archiv der Pharmazie* published by Wiley-VCH GmbH on behalf of Deutsche Pharmazeutische Gesellschaft.

**TABLE 1** An overview of additional binding site comparison tools published since the ProSPECCTs benchmark study from 2018.<sup>14</sup>

Year	Method	Benchmark	Identification	Modeling	Data structure	Alignment
2019	DeepDrug3D <sup>[15]</sup>	TOUGH-C1 <sup>[15]</sup>	Ligand	Interactions	3D points	No
2020	DeeplyTough <sup>[16]</sup>	Vertex <sup>[17]</sup> and ProSPECCTs <sup>[14]</sup> without the ROCS Structures data set <sup>[18]</sup>	Ligand or fpocket 2.0 <sup>[19]</sup>	Interactions	3D image	No
2020	ProCare <sup>[20]</sup>	Balanced Vertex <sup>[20]</sup>	VolSite <sup>[21]</sup>	Interactions	3D points	Yes
2021	PocketShape <sup>[22]</sup>	sc-PDB data set for method evaluation <sup>[23]</sup>	Ligand	Residues	Matrix	Yes
2021	Site2Vec <sup>[24]</sup>	ProSPECCTs without the ROCS Structures data set, APOCS3, <sup>[25]</sup> PLIC data set, <sup>[26]</sup> TOUGH-C1	Ligand	Residues	Histograms	No
2022	BindSiteS-CNN <sup>[27]a</sup>	TOUGH-C1, ProSPECCTs without the ROCS Structures data set	Ligand or SURFNET <sup>[28]</sup>	Surface	Graph	No
2023	TWN-RENCOD <sup>[29]b</sup>	Developer-built kinase data set	Ligand	Residues	Matrix	No

Note: Explanation of the column headers: Benchmark—data sets used for method benchmarking; Identification—binding site detection method; Modeling—binding site representation; Data Structure—data structure used for similarity calculation; Alignment—the ability to provide binding site alignments.

<sup>a</sup>The GitHub repository for this method is still under preparation: <https://github.com/Jing9558/BindSiteS-CNN>.

<sup>b</sup>This method is not publicly available.

grids, fingerprints, 3D points).<sup>[14]</sup> Due to the large number of tools and the different noncomparable evaluations based on differing benchmark data sets, a collection of data sets was developed to comprehensively and comparably assess binding site comparison performance in various application areas (ProSPECCTs).<sup>[18]</sup> Thereby, it aims to facilitate choosing a suitable tool for a specific application, which is a nontrivial task given the individual limitations of the tools. Benchmarking binding site comparison tools revealed that none of the evaluated tools showed an overall superiority. In consequence, choosing the appropriate tool depends on the application.

Many recently developed tools use machine learning (ML) and deep learning methods. Three of them, DeeplyTough,<sup>[16]</sup> BindSiteS-CNN,<sup>[27]</sup> and DeepDrug3D,<sup>[15]</sup> use convolutional neural networks. Site2Vec,<sup>[24]</sup> a mathematical enhancement of PocketMatch,<sup>[30]</sup> considers pairwise distances between representative points of binding sites, showing good overall performance for the ProSPECCTs<sup>[14,18]</sup> data sets. However, the applicability of these methods suffers from the lack of corresponding binding site alignments, which are fundamental for evaluating the results in the context of SBDD.

PocketShape<sup>[22]</sup> provides structure alignments by calculating residue assignments based on the Hungarian algorithm.<sup>[31]</sup> Since the authors did not evaluate the method's performance on standard benchmark data sets, we cannot compare it to the state of the art or assess its suitability for SBDD applications. In addition, its run time is much higher than that of the best-performing tools analyzed in earlier benchmark studies<sup>[14]</sup> (seconds vs. nano- and microseconds scale).

To find an optimal superposition of points sharing similar pharmacophoric and topological neighborhoods, ProCare<sup>[20]</sup> uses the point cloud registration concept. It was benchmarked on a balanced version<sup>[20]</sup> of the Vertex data set.<sup>[17]</sup> The authors show pairwise comparison run times in the seconds time scale. Its performance with an area under the receiver operating

characteristics (ROC) curve (AUC) of 0.811 was significantly lower than that of the best-performing tool, ProBiS, with 0.896.

TWN-RENCOD uses topological water networks obtained by short molecular dynamics simulations.<sup>[29]</sup> The aqueous environment in binding sites from these simulations is compared. The method was evaluated on a kinase data set comprising only 36 binding site pairs. Alignments were not reported.

To overcome the drawbacks of missing structure alignments, insufficiently comparable evaluations, a method choice dependent on the application field, and restricted availability, we present SiteMine, a new database- and structure-based alignment-providing binding site similarity search tool based on GeoMine.<sup>[10,32,33]</sup> SiteMine builds on the NAOMI library with numerous methods for handling and analyzing biomolecular structures<sup>[34]</sup> and small organic molecules.<sup>[35,36]</sup> GeoMine is a tool for searching user-defined 3D geometric patterns enhanced with textual and numerical filters within predicted and ligand-based small molecule binding sites. A fully automated workflow calculates all data, populating a classical relational database. Every binding site atom is converted into a search point and stored with its properties (source molecule type, element, atom type, protein residue type, solvent-exposed surface area, functional group, and secondary structure type), allowing to customize the search for 3D geometric patterns. The distances between all search point pairs below 15 Å are stored to create 3D geometric search patterns.

SiteMine relies on the TetraScan approach we designed for complex 3D shape-matching applications. In short, tetrahedral search patterns are processed by GeoMine to retrieve binding site matches. In addition to SiteMine, TetraScan is also used for searching geometrically similar protein-protein interfaces (submitted for publication). All proposed matches are subsequently scored and ranked. The best-scored match is superimposed by SiteMine.

Here, we present SiteMine in detail and assess its performance on the ProSPECCTs benchmark. We applied SiteMine on cathepsin L, a promising drug target for SARS-CoV-2 inhibition,<sup>[37]</sup> highlighting the applicability of SiteMine for drug repurposing.

## 2 | RESULTS AND DISCUSSION

All results on the ProSPECCTs data set will be summarized in the following. Furthermore, we evaluate SiteMine's performance on the *Balanced Vertex* data set and show the impact of using predicted pockets compared to ligand radius-defined binding sites. Finally, we showcase the applicability of SiteMine for the drug target cathepsin L.

### 2.1 | Run time

SiteMine's run times are determined for all-against-all comparisons (10,000) of the *Kahraman* data set.<sup>[38]</sup> The test system had Postgres 14.3 (default settings, `max_parallel_workers = 8`), an Intel(R) Core™ i5-8500 CPU with 3.00 GHz and 16 GB RAM installed (workstation). SiteMine was used as a single-core application with a multi-threaded database search. Table 2 shows the run time of SiteMine compared with other binding site comparison tools for the same data set. SiteMine Fast has the lowest average run time per comparison within the tools that model sites as 3D points (18 ms). In contrast, the Precise setting with a run time of 122 ms is still faster than the well-performing 3D point-based methods.

### 2.2 | Benchmark studies

The ProSPECCTs data sets were used to benchmark the Fast and Precise settings (see Section 4.8) of SiteMine (Table 3). To ensure a fair comparison for the *ROCS Structures* data set, pairs used for optimization (*Optimization Structures* data sets) were excluded. Therefore, all AUC and enrichment factor (EF) values were recalculated for the reduced data set with scoring tables produced for earlier publications.<sup>[14,18]</sup>

SiteMine Precise achieved the highest mean AUC with 0.835. With a difference of only 0.029 in the average AUC, SiteMine Fast is the third-best method. This difference can be attributed to performance disparity on the *Barelier*, *Decoy Structures Rational*, *Kahraman*, and *ROCS Structures* data sets. This performance loss regarding the AUC is less than 0.2 with the remaining data sets (*Decoy Structures Shape*, *Structures with Identical Sequences*, *NMR Structures*, and *Successful Applications*). Due to their high sequence similarity, these structures have strongly conserved structural binding sites. As a result, atom mappings are obtained even with a lower distance tolerance. Comparing both methods regarding the EFs (Supporting Information S1: Tables S1–S8), the trend of the observed AUC drop does not occur. A considerable difference was only found

for the *Barelier* data set for which SiteMine achieves the lowest AUC values. Compared to all other methods, it becomes clear that this is not a weakness of SiteMine but can be attributed to the design of the benchmark. The outcomes for this data set should not be over-interpreted, as the number of considered pairs is relatively small (62 pairs), in contrast to the other data sets (Table 8).

The performance differences between the SiteMine parameter sets are due to the optimization procedure. In particular, a higher filter number and more permissive tolerances increase the probability of finding atom mappings for binding site superposing and similarity score calculation. Going from Fast to Precise increases the computational cost (Table 2) but leads to significantly more structural superpositions, making it more likely to find weakly associated matches. This trend is more evident when examining the *NMR Structures* data set results (Supporting Information S1: Figure S1). Comparing SiteMine's score distributions for similar pairs, the lower whisker is 0.3 higher with the Precise setting.

Regarding the AUC (Table 3), two other tools stand out compared to SiteMine: SiteHopper<sup>[46]</sup> and KRIPO.<sup>[39]</sup> Although SiteEngine<sup>[49]</sup> and SMAP<sup>[48]</sup> show a high average AUC as well, their performance is already significantly lower, especially on data sets of similar binding pockets in unrelated proteins (*Kahraman* and *ROCS Structures* data set). Therefore, we focused on a comparison of SiteMine with SiteHopper and KRIPO.

SiteHopper is a surface-based binding site similarity tool. SiteHopper defines the binding site via a ligand-based radius. It calculates residue-based chemical properties annotated in a 3D shape calculated by two OpenEye toolkits: Shape<sup>[52]</sup> and Spicoli.<sup>[53]</sup> Alignments and similarities are calculated based on the physico-chemical and surface shape similarity. For the *Successful Applications* data set, the AUC values achieved with SiteMine are 0.12 and 0.14 higher in Fast and Precise mode, respectively. Also, the early enrichment for SiteHopper is inferior to that for both SiteMine settings. However, SiteHopper performs similarly to SiteMine on the *ROCS Structures* data set. Although there are no considerable differences in early enrichment (Supporting Information S1: Table S8), the AUC of SiteMine Fast is 0.05 lower. SiteMine Precise shows comparable performance to SiteHopper in terms of AUC and early enrichment. The performance similarities of both methods are remarkable, given that SiteHopper builds on the *ROCS*<sup>[54]</sup> 3D shape- and chemical feature-based ligand comparison application used to generate the *ROCS Structures* data set. For the *Barelier* data set, SiteHopper and SiteMine show almost identical performance. Only SiteMine Precise is superior regarding AUC and early enrichment (Supporting Information S1: Table S1). For both *Decoy Structures* data sets, SiteHopper's AUC is slightly higher than SiteMine's AUC. Regarding early EF for the *Decoy Structures Rational* data set, the SiteMine methods are equally well-performing (Supporting Information S1: Table S2).

For detecting minor differences, Spearman's Rho correlation coefficients were calculated using the *Decoy Structures* data sets by ranking the scores for different numbers (1–5) of introduced binding site mutations (Supporting Information S1: Table S9). While

TABLE 2 Run times of binding site comparison methods.

Method	Data basis	Preparation run time (s) (number of structures)	Comparison run time (s) (number of comparisons)	Total run time (s)	Average pairwise run time (s)
PocketMatch <sup>[30]</sup>	Distance lists	28.97*	0.28	29.25	0.00028
KRIPO <sup>[39]</sup>	Fingerprint	446.50	0.92	447.42	0.00092
RAPMAD <sup>[40]</sup>	Histogram	71.42 (100)	2.36 (8,281)	73.78	0.000285
FuzCav <sup>[41]</sup>	Fingerprint	399.88 (96)	5.59 (9,216)	405.47	0.000607
FuzCav (PDB)	Fingerprint	236.73 (96)	5.64 (9,216)	242.37	0.000612
TM-align <sup>[42]</sup>	Matrix	25.72*	65.96	91.68	0.006596
SiteMine Fast	3D points	169.56 (100)	186.51	369.09	0.018651
Shaper (PDB) <sup>[21]</sup>	3D points (grid)	181.16 (96)	364.42 (9,216)	545.58	0.039542
Shaper	3D points (grid)	384.21 (96)	367.21 (9,216)	751.42	0.039845
VolSite/Shaper	3D points (grid)	537.00 (76)	248.77 (5,776)	785.77	0.043070
ProBiS <sup>[43]</sup>	Graph	6.95	479.32	486.27	0.047932
VolSite/Shaper (PDB)	3D points (grid)	259.54 (57)	162.26 (3,249)	421.80	0.049942
TIFP <sup>[44]</sup>	Fingerprint	228.30 (77)	550.88 (5,929)	779.18	0.092913
TIFP (PDB)	Fingerprint	194.36 (47)	205.56 (2,209)	399.92	0.093056
SiteMine Precise	3D points	169.56 (100)	1,215.30	1,400.08	0.121530
Grim (PDB) <sup>[45]</sup>	Graph	169.33 (96)	1,714.49 (9,216)	1,883.82	0.186034
Grim	Graph	220.17 (95)	2,104.99 (9,025)	2,325.16	0.233240
IsoMIF <sup>[11]</sup>	Graph	752.83	2,561.44	3,314.27	0.256144
SiteHopper <sup>[46]</sup>	3D points	154.01	3,828.61	3,982.62	0.382861
Cavbase <sup>[47]</sup>	Graph	67.89 (100)	21,823.71 (8,281)	21,891.60	2.635396
SMAP <sup>[48]</sup>	Graph	1.69	42,346.74	42,348.43	4.234674
SiteEngine <sup>[49]</sup>	3D points	328.81	81,193.54	81,522.35	8.119354
SiteAlign <sup>[50]</sup>	Fingerprint	28.97*	286,326.41	286,355.38	28.632641

Note: The star (\*) denotes exemplary run times for separate preprocessing steps. The SiteMine rows are highlighted in light gray. The table and run times of the other methods are extracted from previous benchmark studies.<sup>[14]</sup> Note that computing times for SiteMine were recorded on different hardware.

SiteHopper's Combo score correlates better with the number of residue substitutions by similarly sized physicochemically diverse residues, SiteMine's score shows a better correlation with the number of residues substituted by differently sized residues. In this context, SiteMine is also one of the most sensitive tools regarding minor differences in the binding site. Regarding the *Structures with Identical Sequences* and *NMR Structures* data set, the early enrichment differences are negligible (Supporting Information S1: Tables S4 and S6). On the *Kahraman* data set, the SiteMine settings perform better than SiteHopper (Supporting Information S1: Table S5). In summary, we can show that SiteMine's performance is comparable and, for some application domains, even superior to SiteHopper.

KRIPO<sup>[39]</sup> defines binding sites via a ligand atom radius of 6 Å. An interaction fingerprint represents the binding site. It encodes residue interaction features and binned residue distances. A modified Tanimoto coefficient<sup>[55]</sup> is the similarity measure.

Concerning the *Kahraman* data set, the performance regarding the AUC of KRIPO is similar to both SiteMine settings. The early enrichment is slightly lower for SiteMine (Supporting Information S1: Table S5). For the *Decoy Structures*, *NMR Structures*, *Structures with Identical Sequences*, and the *Successful Applications* data set, SiteMine's performance is superior regarding AUC and EFs (Supporting Information S1: Tables S2-S4, S6, S7). For the *ROCS Structures* data set, the AUCs of SiteMine Fast and KRIPO are similar. A performance difference regarding the EFs is apparent: in contrast to SiteMine, the enrichment of similar pairs by KRIPO decreases with increasing percentages of screened data (Supporting Information S1: Table S8). KRIPO's fingerprint-based approach is faster than SiteMine, but binding site superpositions are not computed on the fly. Instead, they are calculated using a clique algorithm. Consequently, KRIPO harbors the disadvantage that the superposition does not necessarily correspond to the fingerprint-based similarity.

**TABLE 3** Overview of the SiteMine results and the tools of the benchmark studies.<sup>14,18</sup>

Method	Mean	Barelier <sup>[51]</sup>	Decoy Structures Rational	Decoy Structures Shape	Structures with Identical Sequences	Kahraman <sup>[38]</sup>	NMR Structures	Successful Applications	ROCS Structures
SiteMine Precise	0.835	0.61	0.69	0.72	1.00	0.78	1.00	0.91	0.97
SiteHopper	0.813	0.56	0.75	0.75	0.98	0.72	1.00	0.77	0.97
SiteMine Fast	0.806	0.56	0.65	0.71	1.00	0.74	0.98	0.89	0.92
KRIPO	0.794	0.73	0.60	0.61	0.91	0.76	0.96	0.85	0.93
SiteEngine	0.771	0.55	0.82	0.79	0.96	0.64	1.00	0.86	0.55
SMAP	0.766	0.68	0.76	0.65	1.00	0.62	1.00	0.86	0.56
SiteAlign	0.759	0.44	0.85	0.80	0.97	0.59	1.00	0.87	0.55
Shaper (PDB)	0.749	0.54	0.71	0.76	0.96	0.66	0.93	0.75	0.68
Shaper	0.746	0.54	0.71	0.76	0.96	0.65	0.93	0.75	0.67
TM-align	0.738	0.59	0.49	0.49	1.00	0.66	1.00	0.88	0.79
VolSite/Shaper	0.734	0.71	0.68	0.76	0.93	0.56	0.78	0.77	0.68
IsoMIF	0.733	0.62	0.59	0.59	0.77	0.75	0.70	0.87	0.97
FuzCav	0.720	0.67	0.69	0.58	0.94	0.55	0.99	0.77	0.57
FuzCav (PDB)	0.718	0.65	0.69	0.58	0.94	0.56	0.98	0.77	0.57
PocketMatch	0.714	0.51	0.59	0.57	0.82	0.66	0.96	0.82	0.78
Cavbase	0.711	0.55	0.65	0.64	0.98	0.60	0.87	0.82	0.58
VolSite/Shaper (PDB)	0.698	0.50	0.68	0.76	0.94	0.57	0.76	0.72	0.65
ProBiS	0.686	0.50	0.47	0.46	1.00	0.54	1.00	0.85	0.67
TIFP	0.680	0.55	0.66	0.66	0.66	0.71	0.91	0.71	0.58
Grim	0.665	0.45	0.55	0.56	0.69	0.69	0.92	0.70	0.76
RAPMAD	0.649	0.60	0.61	0.63	0.85	0.55	0.82	0.74	0.39
Grim (PDB)	0.633	0.45	0.57	0.56	0.62	0.61	0.85	0.64	0.76
TIFP (PDB)	0.598	0.56	0.56	0.57	0.55	0.54	0.78	0.66	0.56

Note: For each tool and data set, the area under curve (AUC) is given. The table is sorted according to the mean AUC for all data sets. The SiteMine methods are highlighted in light gray. The *Optimization Structures* data set pairs are excluded from the *ROCS Structures* data set for benchmarking all methods on this data set.

SiteMine is also applicable to compare predicted binding sites, representing a considerable advantage over SiteHopper and KRIPO. The *ROCS Structures* data set corresponds to common use cases, finding similar binding sites in unrelated proteins (e.g., off-target prediction). Also, IsoMIF<sup>[11]</sup> demonstrates remarkable performance but is slower than SiteMine. Considering the total of all data sets (AUC, EF, Spearman's Rho), SiteMine shows promising performance (Table 4).

### 2.3 | Comparison to ML tools evaluated on ProSPECCTs

Since their publication, the ProSPECCTs data sets have been used to evaluate three ML-based methods: DeeplyTough, Site2Vec, and

BindSiteS-CNN. Therefore, we can readily compare them to SiteMine (Table 5). Although the methods were not evaluated on the *ROCS Structures* data set, the second closest to a real-world application scenario, we can assess their general applicability to realistic use cases employing the *Successful Applications* data set.

Site2Vec has the highest mean AUC (0.87), while BindSiteS-CNN and DeeplyTough rank below SiteMine. The AUC values for the *Decoy Structures* data sets are lower for SiteMine. In contrast, SiteMine's AUC values are significantly higher for the *Successful Applications* data set than for the three ML-based methods. As this data set represents the most meaningful data set regarding SBDD studies, the poor performance of Site2Vec for this data set questions its applicability in SBDD.

Moreover, recent tools do not provide binding site alignments for proper visual inspections, restricting their usefulness for structure-based



TABLE 4 Criteria of importance for choosing a suitable binding site comparison method.

Method	Preparation (ease)	Preparation (completeness)	Application to predicted sites	Run time <sup>a</sup>	Definition <sup>b</sup>	Definition (ranking) <sup>b</sup>	Flexibility <sup>c</sup>	Properties (ranking) <sup>d</sup>	ROCS structures	Successful applications	Visualization
SiteMine	+	+	+	/	+	+	+	+	+	+	+
Cavbase	+	-	+	-	+	+	+	+	-	+	+
FuzCav	/	+	+	+	/	/	+	+	-	+	-
Grim	/	-	-	/	-	-	+	-	/	-	+
IsoMIF	+	+	+	/	-	-	-	-	+	+	+
KRIPO	+	+	-	+	/	/	+	+	/	+	+
PocketMatch	-	-	(+)	+	/	/	+	-	+	+	-
ProBIS	+	+	(+)	+	+	+	+	-	-	+	+
RAPMAD	+	-	+	+	-	-	-	+	-	-	-
VolSite/Shaper	/	-	+	/	+	/	-	+	-	+	+
SiteAlign	-	+	(+)	-	+	+	+	+	(+)	+	+
SiteEngine	+	+	-	-	+	/	+	+	-	+	+
SiteHopper	+	/	(+)	/	+	+	+	+	+	+	+
SMAP	+	+	(+)	-	+	+	+	+	-	+	+
TIFP	/	-	-	/	-	-	+	-	-	-	-
TM-align	-	+	(+)	/	+	+	+	n/a	+	+	+

Note: Besides its intermediate run time, SiteMine is superior to other tools investigated previously.<sup>[14,18]</sup> With respect to run time evaluation, “+”, “-”, “/”, “n/a”, “u”, “s” denote comparison algorithms that require several ns,  $\mu$ s, or s per comparison, respectively. With respect to the scoring, a “+” was assigned to tools if the intervals of upper and lower whiskers of active and inactive pairs do not overlap. A “/” was assigned to tools whose upper and lower quartiles for the pairs do not overlap. With respect to other factors, tools that were clearly outperformed by many other tools were assigned a “-”. The table was adapted based on earlier benchmark studies.<sup>[5]</sup>

<sup>a</sup>Kahnman data set.

<sup>b</sup>Structures with Identical Sequences data set.

<sup>c</sup>NMR Structures data set.

<sup>d</sup>Decoy Structures Rational & Shape data set.

**TABLE 5** Performance comparison of SiteMine with ML-based methods for the ProSPECCTs data sets.

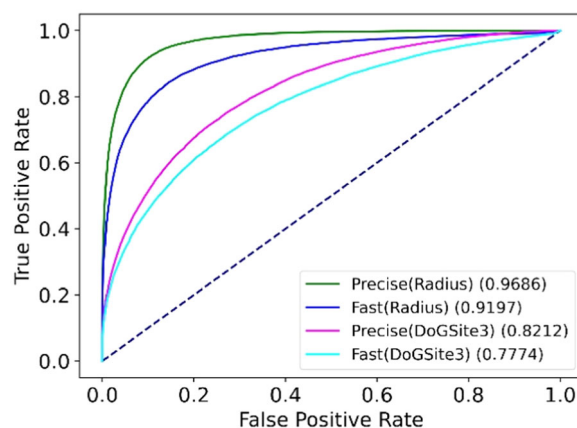
Method	Structures with Identical Sequences	NMR Structures	Decoy Structures Rational	Decoy Structures Shape	Kahraman	Barelier	Successful Applications	Mean
SiteMine Precise	1.00	1.00	0.69	0.72	0.78	0.61	0.91	0.82
SiteMine Fast	1.00	0.98	0.65	0.71	0.74	0.56	0.89	0.79
Site2Vec	1.00	1.00	0.99	0.99	0.86	0.53	0.66	0.87
BindSiteS-CNN	0.94	0.83	0.91	0.79	0.66	0.62	0.78	0.79
DeeplyTough	0.95	0.90	0.76	0.75	0.63	0.54	0.83	0.77

Note: The table shows the area under curve (AUC) values for the individual data sets. The results of the other methods were extracted from previous studies.<sup>[16,24,27]</sup>

studies. The number of applications that solely rely on similarity scores is considerably low,<sup>[12]</sup> highlighting that superpositions are indispensable for evaluating the results to assess their impact on SBDD projects. As shown in Table 5, the performances of ML-based comparison methods are inferior for data sets of active site pairs without sequential relationships (*Kahraman*, *Barelier*, *Successful Applications*). While detecting sequentially related binding sites can be regarded as solved, further developments of ML-based methods should focus on a good performance on structurally similar binding sites with low sequence similarity. Future efforts should also focus on reporting reliable binding site alignments.

## 2.4 | Evaluation of the impact of ligand radius-defined and predicted binding sites

Binding sites defined by ligands introduce a bias since the binding site description includes the ligand's exact location and size. Therefore, we evaluated the impact of predicted binding sites on SiteMine's performance on the ProSPECCTs benchmark data sets. To ensure the correctness of the compared binding site, DoGSite3's<sup>[57]</sup> new mode for detecting difficult ligand-occupied pockets was used. Here, ligand fragments are only used to bias the binding site grid if the ratio of the bound solvent-accessible surface area and the unbound solvent-accessible surface area is below 0.35 (maxSASLigandRatio). Using this feature, we ensure that we use the ligand-occupied pocket without biasing the binding site dimensions based on the ligand alone. The Supporting Information includes all AUC values and EFs for SiteMine on all ProSPECCTs data sets (Supporting Information S1: Table S10). SiteMine with predicted pockets performs slightly worse in terms of AUC for the *Successful Applications* and *NMR Structures* data sets. For both *Decoy Structures* data sets, a similar performance is observed. In contrast, the mean Spearman's Rho correlation coefficients decrease for the *Decoy Structures* data sets (Supporting Information S1: Tables S11 and S12) as mutations might lead to different binding site dimensions. SiteMine performs similarly with both types of binding site definition for the *Structures with Identical Sequences* data set.



**FIGURE 1** Receiver operating characteristics (ROC) curves for the ROCS Structures data set for SiteMine settings with both types of binding site definitions: ligand radius-based (Radius) and DoGSite3-defined (DoGSite3).

Using predicted pockets, SiteMine performs significantly poorer for the *Kahraman* and *ROCS Structures* data sets (see Figure 1). Both data sets have a substantial similarity: the similar binding site pairs contain similar or identical ligands. The similarity classification for the other data sets except *Barelier* relies on the protein instead of the ligand. Due to the ligand radius-based binding site definition, per se, the comparison is biased, focusing on the probably most similar parts of the sites: Similar or identical ligands in the *ROCS Structures* or the *Kahraman* data set show some similarity when evaluating size alone. The pockets are considerably larger when not selecting site residues based on the ligand. The increased binding site size leads to poor scoring performance, as the score is normalized by the larger binding site in terms of the number of solvent-exposed heavy atoms. In addition, SiteMine was optimized on a set of ligand-derived sites with similarly sized ligands.

However, one of the most common applications of binding site comparison is screening a database of ligand-based pockets against a predicted site to find potential ligands or off-targets, necessitating a high early enrichment rather than a promising overall performance. Given the comparison of the EFs for ligand radius-based and predicted pockets, SiteMine performs equally well in both scenarios.

## 2.5 | The Balanced Vertex data set

ProCare,<sup>[20]</sup> which is not ML-based, was not evaluated on the ProSPECCTs<sup>[14,18]</sup> data sets. Instead, the authors modified the Vertex<sup>[17]</sup> data set to create a balanced version with 676 pairs (338 similar and dissimilar). The Vertex data set, as the ROCS Structures data set, was developed based on the hypothesis that similar ligands bind to similar sites. Therefore, the similar site pairs were derived from their ligand-based similarity. Both data sets differ in the considered ligands for data set generation. In contrast to the ROCS Structures data set, the Vertex data set considers binding affinities. Due to the ambiguity of some binding site ligands for the Balanced Vertex<sup>[20]</sup> data set, we have revised the ligand identifiers of the binding site pairs (see Supporting Information S1: Table S13). Note that it is undocumented how the ambiguity was resolved by the other tools. ProCare compares VolSite-predicted<sup>[21]</sup> binding sites. For the Balanced Vertex<sup>[20]</sup> data set, SiteMine was benchmarked using DoGSite3-defined (maxSASLigandRatio = 0.35) and ligand radius-defined binding sites to allow a fair comparison (Table 6). Here, SiteMine with the Precise setting performs better than the Fast setting, as observed for the ProSPECCTs data sets regarding the AUC. Also, ligand radius-defined binding sites result in a higher AUC than DoGSite3-defined pockets. SiteMine performs better than ProCare except for the Fast setting with predicted pockets. ProCare's average pairwise run time of 2 s is several orders of magnitude slower than SiteMine's (see Table 2).

Looking at the performance of the analyzed tools for this data set, it meets the eye that methods performing only mediocly in previous studies perform best on this data set (PocketMatch and

**TABLE 6** Performance comparison for the *Balanced Vertex* data set.<sup>20</sup>

Method	AUC	Completeness (%)
SiteMine Precise (Radius)	0.906	98.5
SiteMine Precise (DoGSite3)	0.898	98.1
ProBiS	0.896	64.2
PocketMatch	0.895	99.4
SiteMine Fast (Radius)	0.874	98.5
KRIPO	0.862	95.2
SiteAlign	0.859	100.0
SiteMine Fast (DoGSite3)	0.846	98.1
FuzCav	0.831	100.0
ProCare	0.811	99.7
Shaper	0.774	99.7

Note: The results of the other methods were extracted from previous studies.<sup>[20]</sup> For SiteMine, the revised version was used (Supporting Information S1: Table S13).

Abbreviation: AUC, area under the receiver operating characteristics curve.

ProBiS), while more reliable tools (KRIPO and SiteAlign) show a poorer performance. This finding can be partially attributed to the high overall similarity of the pairs classified as similar, leading to a good performance of approaches relying solely on the pockets' residues sequence in a 7 Å radius and using sequence identity as the scoring measure (AUC of 0.9).<sup>[17]</sup> Furthermore, the Vertex data set includes pairs of structurally and functionally related protein pairs (e.g., protein kinases or phosphodiesterase enzymes) in both the active and inactive pairs, which might be caused by their selection relying solely on data available in the ChEMBL database.<sup>[58]</sup> In summary, we can conclude that the *Balanced Vertex* data set cannot reflect realistic scenarios for which elaborate binding site comparison tools are indispensable.

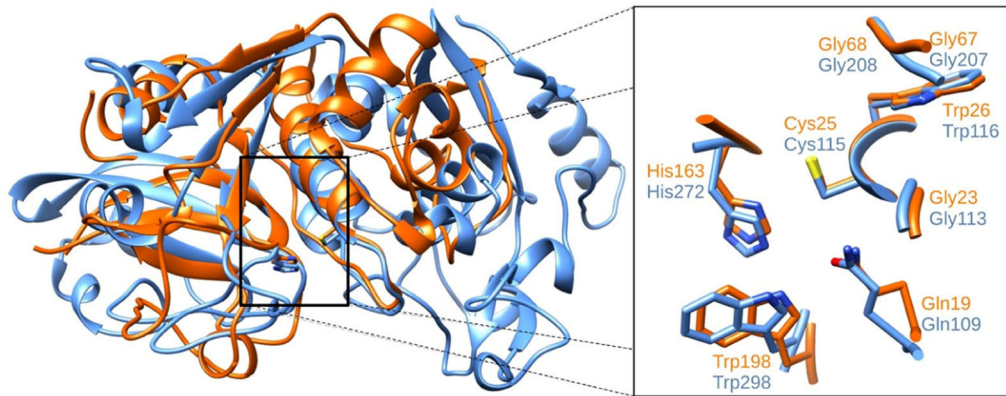
## 2.6 | Cathepsin L—searching for similar sites in the PDB

Cathepsins belong to the peptidase C1 family and play a role in the hydrolytic degradation of the extracellular matrix.<sup>[59]</sup> They also participate in apoptosis and antigen processing, as well as lysosomal recycling of cellular proteins. Cathepsin L, in particular, plays a pivotal role in the infection of human coronaviruses such as SARS-CoV and SARS-CoV-2 by facilitating their entry into the cell through proteolysis of the spike protein.<sup>[60]</sup> Inhibition of this protease can thus prevent infection, making it a target of interest for SBDD.<sup>[37,61]</sup>

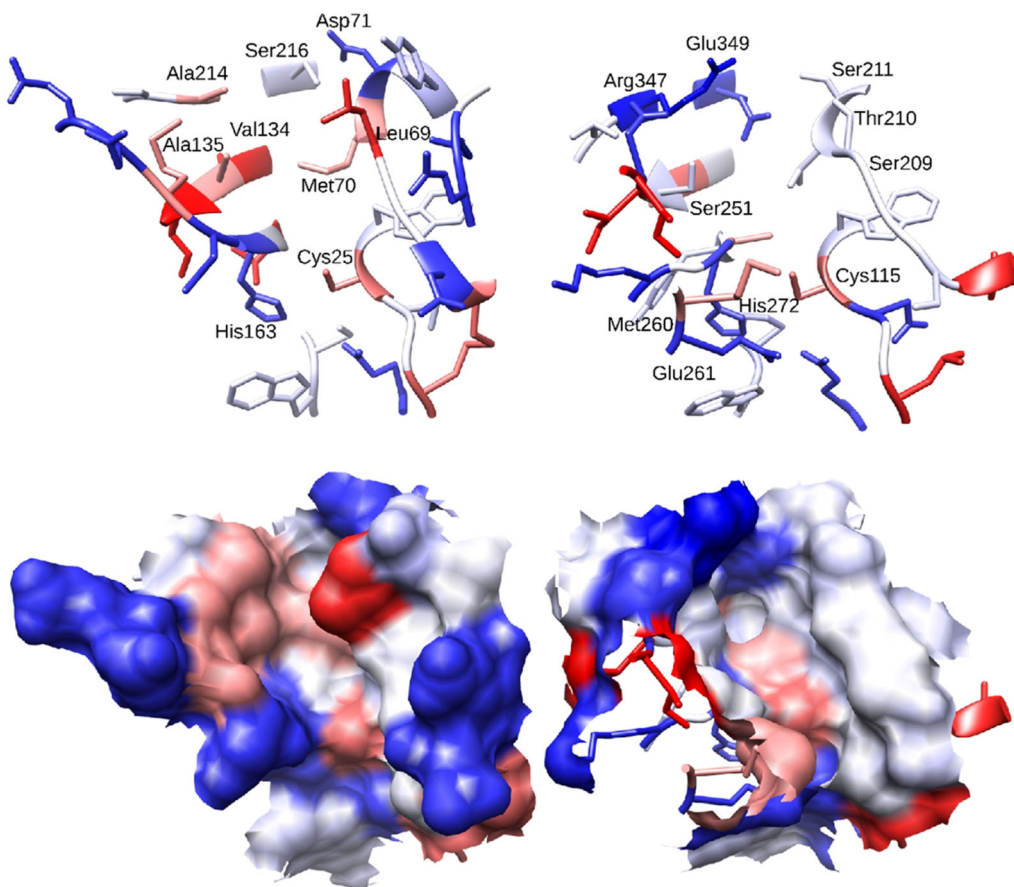
A sequence-culled PDB subset was created with PISCES<sup>[62]</sup> (see Supporting Information S1: Table S14 for parameters) to speed up the search. For this subset of 40,207 PDB entries, a database was built using ligand radius-defined binding sites. For searching, SiteMine's Fast setting was used. The query binding site was defined via the bound inhibitor (radius of 6.5 Å, ligand identifier: 424) of PDB entry 2xu1. Searching this database with 63,106 binding sites took 230 s single-threaded on a single desktop computer, corresponding to an average run time per comparison of about 3 ms.

We inspected the 30 top-scored (normalized by the larger binding site, Supporting Information S1: Table S15) binding site superpositions and mostly found papain-like proteases of a similar fold. Searching for nonobvious similarities in differently folded structures, we also inspected the top 30 superpositions of non-normalized scores (raw scores, Supporting Information S1: Table S16). On rank 27, we found the active site of human calpain 1 (PDB entry 1zcm). Interestingly, the rank within the normalized scores is much higher at 573 but still within the top 85% (Supporting Information S1: Figure S2).

The sequence identity is low (14.5%, EMBOSS Needle<sup>[63]</sup>), but both proteins belong to the same family of cysteine proteases. This similarity becomes more evident when examining the binding site superposition (Figure 2). It shows eight superimposed identical residues with similar side chain orientations. These residues are near the catalytic center of the reactive cysteines (cathepsin L—Cys25, calpain 1—Cys115).



**FIGURE 2** The SiteMine binding site superposition of human cathepsin L (orange, PDB entry 2xu1) and human calpain protease (blue, PDB entry 1zcm). Identical residues of chain A of the binding sites are shown. The image was created with UCSF Chimera.<sup>[64]</sup>



**FIGURE 3** SiteMine binding site alignment of cathepsin L (left, PDB entry 2xu1) and human calpain protease (right, PDB entry 1zcm). Top: residue arrangement. Bottom: the surface representation. The residues are color-coded according to the hydrophobicity scale of Kyte and Doolittle<sup>[65]</sup> in UCSF Chimera<sup>[64]</sup> and UCSF ChimeraX<sup>[66]</sup> (low hydrophobicity—blue, high hydrophobicity—red). The catalytic residues (His163 and Cys25, His272 and Cys115) in their hydrophobicity scale and further site residues are labeled.

Also, the pockets differ in several aspects (Figure 3). Regarding the binding site shape, in calpain 1, residues Met260 and Glu261 narrow the pocket moderately and cause a slight closure in the front part of the pocket (S1 pocket<sup>[67]</sup>). Furthermore, the

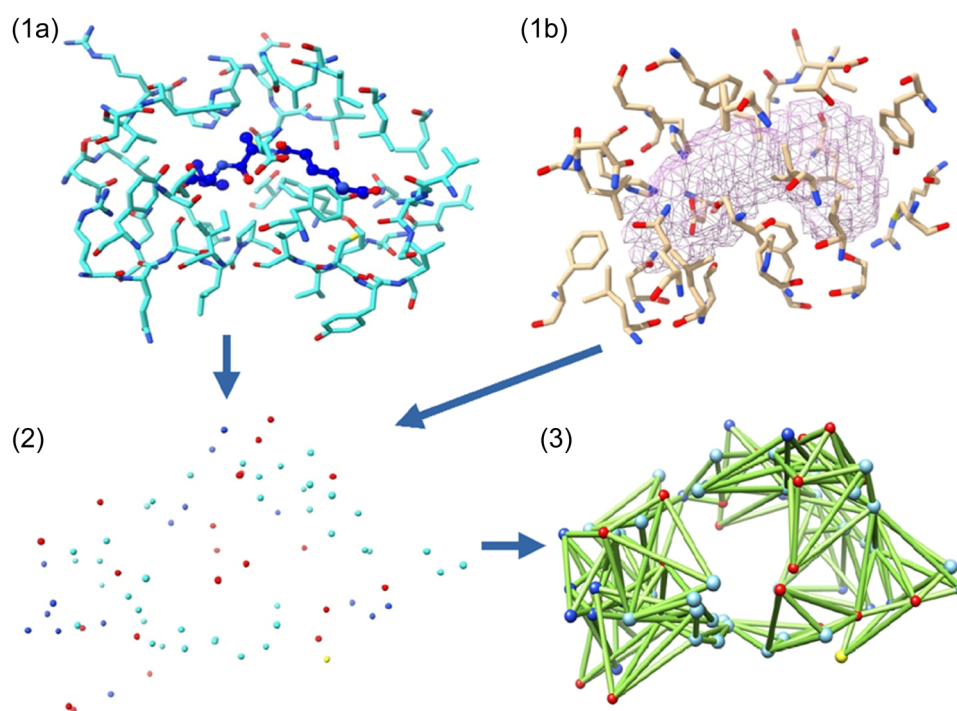
properties of some residues in the posterior (S2, S3 pocket<sup>[67]</sup>) part of the binding site differ. While cathepsin L is predominantly lipophilic (lipophilic: Ala135, Ala214, Leu69, Met70; hydrophilic: Ser216, and Asp71), calpain 1 has predominantly hydrophilic

residues (hydrophilic: Ser251, Arg347, Glu349, Ser209, Thr210, and Ser211).

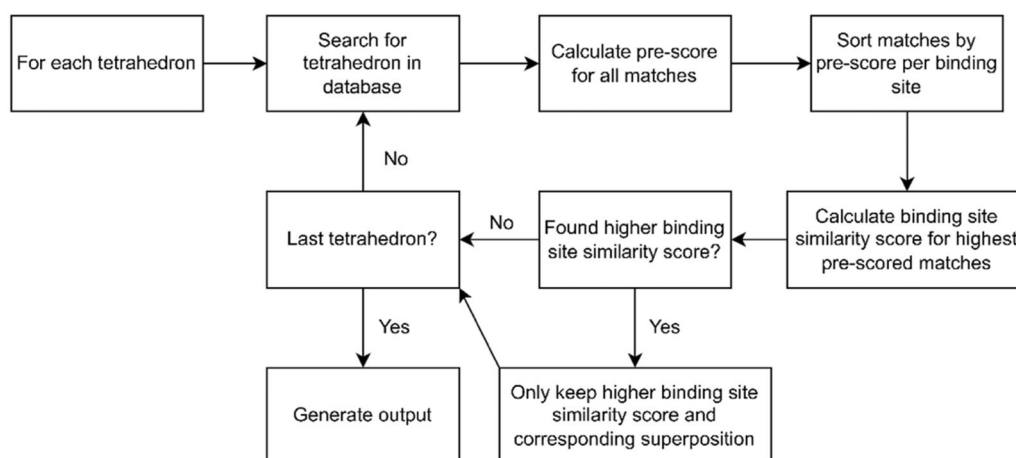
The identified binding site similarity assists in SBDD. On the one hand, the two binding sites have a high global similarity, meaning that inhibitors binding to calpain 1 could also bind to cathepsin L (drug repurposing, off-target prediction). On the other hand, selectivity is achievable by exploring the identified differences. One possibility would be to superpose available structures with bound inhibitors and derive the core and specificity-mediating fragments to design new, potentially more specific binders.

### 3 | CONCLUSION

Searching for similar protein binding sites can support several SBDD challenges, such as drug repurposing, analyzing protein–ligand and protein–protein complexes, and off-target or function prediction. According to the review by Eguida and Rognan,<sup>[13]</sup> almost 40 software tools have been developed in the past 20 years. However, only a few were evaluated based on unique benchmark sets<sup>[14,18]</sup> to determine strengths and weaknesses and, thus, their application domains. None of the comprehensively benchmarked tools showed a promising



**FIGURE 4** Binding site modeling with SiteMine. SiteMine supports using (1a) ligand radius-defined or (1b) predicted binding sites. The solvent-accessible atom selection results in an atom subset (2). Tetrahedra are built and selected to represent the query-binding site (3). Element-specific atom coloring: cyan/beige—carbon, red—oxygen, blue—nitrogen, yellow—sulfur. The image was created with UCSF ChimeraX.<sup>[66]</sup>



**FIGURE 5** Binding site comparison with SiteMine.

performance regarding critical criteria for reliable SBDD, that is, good performance in terms of AUC and early enrichment for all data sets, the possibility of comparing predicted binding sites, and reasonable run time to screen extensive collections of protein binding sites.

In this work, we introduced SiteMine, a new database-driven binding site comparison method providing similarity scores and the corresponding alignments. We evaluated it using the ProSPECCTs benchmark data sets, also comparing it to published tools. SiteMine is one of the best-performing tools for all data sets, demonstrating its broad applicability. In the run time comparison, SiteMine is slower than fingerprint-based methods but among the fastest for tools with comparable features regarding binding site modeling and the possibility of providing alignments. SiteMine is available for Linux, macOS, and Windows as part of the NAOMI ChemBio Suite (<https://uhh.de/naomi>) and is free for academic use and evaluation purposes. To enable screenings for similarity searches in huge databases, we established a second parameter set (Fast) in addition to the Precise setting. Therefore, we recommend the Fast setting for a large-scale similarity screening with subsequent Precise setting runs to improve scores and superposition on a promising selection.

Potential binders of novel detected binding sites can be predicted by screening for similar ligand-bound pockets, representing a frequent use case of automated binding site comparisons. These comparisons are particularly useful for binding sites in proteins with a low overall structural similarity to already known structures. We showed that our method performs reliably using both predicted and ligand-defined binding sites. We also realized a significant performance loss upon comparing differently sized binding sites, indicating the importance of adjusting the score normalization in screenings for similar sites with one query.

Given the rising number of developed binding site comparison tools, the scientific community might further benefit from even better-performing methods. However, it renders choosing a suitable tool infeasible without commonly used benchmark sets and unique evaluation pipelines. Current ML-based approaches gain attention, but their applicability suffers from the lack of binding site alignments. With SiteMine, we present a novel tool to the SBDD community that is easy to use, applicable to predicted sites, and shows promising performance regarding the most crucial quality criteria.

Despite SiteMine's comparably good run time within alignment-providing methods, a similarity search within the AlphaFold database with over 200 million structures seems challenging. This task becomes even more complex when using structure ensembles to consider protein flexibility.

SiteMine can be successfully applied for selectivity analyses and the discovery of novel targets for known drugs. In an application showcase, we used SiteMine to search for similar binding sites for cathepsin L. We found a high similarity to the active site of calpain 1. Thus, some inhibitors derived from calpain 1 might also bind cathepsin L, opening a potential avenue for drug repurposing. Similarly, calpain 1 should be considered a potential off-target when profiling cathepsin L binders for selectivity.

In summary, we hope the scientific community will benefit from using SiteMine in various SBDD projects and find the depicted similarity for cathepsin L and calpain 1 inspiring for searching for new inhibitors.

## 4 | EXPERIMENTAL

In the following, we describe and visualize the SiteMine comparison algorithm and outline tailor-made benchmarking data sets and the performance assessment (see also Figures 4 and 5). Subsequently, we provide details regarding the parameter optimization of SiteMine.

### 4.1 | Binding site definition

Predefined binding sites can easily be fetched from the GeoMine database. SiteMine also parses DoGSite3-predicted<sup>[57]</sup> binding sites or site atoms in a 6.5 Å radius of the ligand's heavy atoms. Alternatively, a custom binding site can be specified using residue IDs. The identification of interactions, protonation and tautomeric states, and hydrogen orientations generated by Protoss is described elsewhere.<sup>[68]</sup>

### 4.2 | Selecting search atoms

For 3D geometrical query generation, all solvent-accessible heavy atoms of all site residues, each aromatic ring center (His, Phe, Trp, and Tyr), and all side chain carbon atoms of hydrophobic residues (Ala, Ile, Leu, Lys, Met, Pro, and Val) are selected.

### 4.3 | Building and selecting tetrahedra

A series of search tetrahedra is constructed using the selected atoms as corners (Figure 4). As searching for all possible tetrahedra (counting to  $N^4$ , where  $N$  is the number of selected atoms) is prohibitive, we introduced an algorithm for tetrahedra selection. The algorithm aims for an equal distribution of tetrahedra across the binding site according to atom usage.

In the first step (Algorithm 1), tetrahedra fulfilling distance and properties constraints are created (L.2). Distances between search atoms corresponding to tetrahedron edge lengths have to be between 1 and 8 Å. The property constraints are the number of atom types representing the tetrahedron corners (see Section 4.8 for details). The user can adapt the distance values for specific application scenarios. Property constraints can be turned on or off.

All resulting tetrahedra are sorted in descending order by the sum of their edge lengths (L.3) to ensure that large tetrahedra with most atoms of minimum atom usage are preferred (L.9). Initially and during tetrahedra selection, the occurrence of every site atom as tetrahedron corner in selected tetrahedra is counted (atom usage count, L.4). The algorithm ensures that we always select tetrahedra with the maximum number of atoms as corners with minimal occurrence so far (L.9). Adding a tetrahedron to the selection list implies its deletion in the list of all tetrahedra and the update of the atom usage counts (L.8–12). The process terminates once the selected number of tetrahedra exceeds a user-defined

**Algorithm 1** Procedure of the tetrahedra selection algorithm

```

1: procedure SELECT_TETRAHEDRA(atoms, max, nofTetrahedra)
2:   allTetrahedra = createAllPossibleTetrahedra(atoms, max)
3:   sortTetrahedra(allTetrahedra)
4:   atomTetrahedraUsage = {atom1 = 0, atom2 = 0, ..., atomn = 0}
5:   selectedTetrahedra = ∅
6:   while |selectedTetrahedra| < nofTetrahedra and min(atomTetrahedraUsage) == 0 do
7:     for i = 4 down to 1 do
8:       for each tetrahedron ∈ allTetrahedra do
9:         if tetrahedron has i atoms with min(atomTetrahedraUsage) then
10:           selectedTetrahedra.add(tetrahedron)
11:           allTetrahedra.delete(tetrahedron)
12:           increase(atomTetrahedraUsage, tetrahedron)
13:         end if
14:       end for
15:     end for
16:   end while
17:   return selectedTetrahedra
18: end procedure

```

count (default: 30) if each atom already occurs in at least one tetrahedron (L.6), ensuring a complete binding site representation.

#### 4.4 | Filter building

The selected tetrahedra represent the 3D geometrical queries (called filters in the following) to search within the GeoMine database for atom mappings in binding sites. The atoms constitute search points annotated by their coordinates, atom types (acceptor, donor, acceptor & donor, aromatic, hydrophobic, anion, or cation), and solvent accessibility. The coordinates are only used to calculate superpositions in the match-processing step. Tetrahedra edges are translated to distance ranges (search point distances including relative tolerances, default: 20%).

#### 4.5 | Match processing

Found filter matches in the target binding sites result in atom mappings. Binding site superpositions are calculated by the C++ Eigen library<sup>[69]</sup> implementation of the Umeyama algorithm.<sup>[70]</sup> A prescore is computed by counting the query C $\alpha$  atom occupancy. An atom within a 6 Å radius (rounded average amino acid diameter 10.6 Å<sup>[71]</sup>) of a target C $\alpha$  atom is considered occupied. This prescore serves as superposition quality estimation. The highest prescored  $\sqrt{N}$  superpositions per binding site are chosen, where  $N$  is the number of query hits found for the binding site. This heuristic limitation does not influence the quality of the result while simultaneously reducing compute resources otherwise spent for more expensive similarity score calculations.

#### 4.6 | Binding site similarity scoring

For each target binding site superimposed on the query, a so-called SP-score consisting of a shape and a pharmacophore component is calculated. For each solvent-exposed atom (solvent-accessible surface > 0 Å<sup>2</sup>) in the query-binding site, neighboring solvent-exposed target atoms in a predefined radius of 1.5 Å are searched. If at least one atom is found, an atom pair is formed, and the shape score is increased by one. If more than one atom is found, the closest one is chosen to build the atom pair. The atom pair's similarity is evaluated by a pharmacophore-based scoring matrix (pharmacophore score, Table 7) and added up to the pharmacophore score. The shape and pharmacophore scores are equally weighted, summed up, and normalized to form the binding site similarity SP-score. For normalization, the score is divided by the maximum number of solvent-exposed atoms of the two compared binding sites. Among all possible superpositions, the one maximizing the SP-score is finally selected. The complete comparison process with SiteMine is summarized in Figure 5.

#### 4.7 | Benchmark data sets

The ProSPECCTs<sup>[14,18]</sup> data sets and the *Balanced Vertex*<sup>[20]</sup> data set are used for method evaluation (Table 8). ProSPECCTs aims to reveal the strengths and weaknesses of binding site similarity search tools.

We compared the performance of SiteMine and other binding site comparison methods evaluated in earlier benchmark studies.<sup>[14,18]</sup> Hence, the same evaluation metrics, that is, the AUC and the EF, are applied.

**TABLE 7** Scoring scheme for an atom pair according to its pharmacophore similarity.

	Acc/Don	Acc	Don	Aro	HyPhob	Ca	Pos/Don	Neg/Acc
Acc/Don	1	0.6	0.6	0	0	0	0.6	0.6
Acc		1	0	0	0	0	0	0.8
Don			1	0	0	0	0.8	0
Aro				1	0.8	0	0	0
HyPhob					1	0	0	0
Ca						1	0	0
Pos/Don							1	0
Neg/Acc								1

Abbreviations: Acc/Don, hydrogen bond acceptor and donor; Acc, hydrogen bond acceptor; Don, hydrogen bond donor; Aro, atom is part of an aromatic system; HyPhob, hydrophobic atom; Ca, alpha carbon atom; Pos/Don, positively charged hydrogen bond donor; Neg/Acc, negatively charged hydrogen bond acceptor.

**TABLE 8** Brief overview of the used data sets.

Data set name	Number of similar pairs	Number of dissimilar pairs	Evaluation purpose
Structures with Identical Sequences <sup>[14,18]</sup>	13,430	92,846	Influence of the binding site definition
NMR Structures <sup>[14,18]</sup>	7,729	100,512	Influence of the binding site flexibility
Decoy Structures Rational <sup>[14,18]</sup>	13,430	13,430	Differentiation of minor physicochemical changes
Decoy Structures Shape <sup>[14,18]</sup>	13,430	13,430	Differentiation of minor shape changes
Barelier <sup>[51]</sup>	19	43	Identification of unrelated binding site pairs with identical ligands in similar environments
Kahraman <sup>[38]</sup>	1,320	8,680	Recovery of sites with identical ligands and cofactors
Successful Applications <sup>[14,18]</sup>	115	56,284	Recovery of known similar binding site pairs
ROCS Structures <sup>[18]</sup>	15,339	56,179	Recovery of similar sites with similar ligands in similar conformations in sequentially unrelated site pairs
Optimization Structures	150	450	SiteMine's parameter optimization (subset of ROCS Structures)
Balanced Vertex <sup>[20]</sup> data set	338	338	Recovery of similar sites with ligands with similar binding affinities

## 4.8 | Parameter optimization

For filter and parameter optimization of SiteMine, a subset of the ROCS Structures data set<sup>[18]</sup> was created, named the *Optimization Structures* data set (Supporting Information S1: Table S17).

All ligands of the similar binding site pairs of the ROCS Structures data set were extracted as SD files and loaded in KNIME 4.3.3.<sup>[72]</sup> Their ECFP4 fingerprints were calculated using the CDK Fingerprints node. Next, a Tanimoto coefficient-based distance matrix was calculated for a k-Medoids clustering with a partition count of 150. This procedure was also applied to the ligands of the dissimilar binding site pairs using a partition count of 450. To compile a data set of 150 "active" and 450 "inactive" site pairs for parameter optimization, we extracted all respective pairs per clustered ligand (this ligand had to be in at least one binding site).

Finally, we randomly selected one pair not already chosen to represent a previously chosen ligand.

The search time of SiteMine is mainly influenced by the distances (tetrahedra edge lengths), their tolerance, the number of filters, and the search point properties (atom types).

To investigate the run time behavior of the filters composed of different point properties, we created filters with all possible property combinations and uniform edge lengths (4.5 Å with a tolerance of 3.5 Å representing a distance range between 1 and 8 Å). We found that filters became faster with increasing numbers of aromatic, anion, and cation points. The opposite was observed with increasing numbers of acceptor, donor, acceptor and donor, and hydrophobic points. The number of matches is inversely proportional to the run time (see Supporting Information S1: Table S18 for details). To find a compromise between optimum run time and performance, we derived the following rules:



**TABLE 9** Final parameter combinations resulting from the optimization.

Name	Minimum number of filters	Distance tolerance (%)	Maximum edge length (Å)
SiteMine Fast	30	20	8
SiteMine Precise	40	25	9

Filters must include at least one aromatic, anion, or cation point and two hydrophobic points at maximum. The latter rule limits the maximum number of hydrophobic points since these considerably contribute to the run time costs compared to acceptor, donor, and acceptor and donor points.

Using these rules, the remaining three parameters were optimized in a brute-force approach (Supporting Information S1: Table S19). The results of this parameter optimization can be found in Supporting Information S1: Table S20. We selected two parameter combinations based on the AUC, EFs, and run time: Fast and Precise (Table 9).

#### ACKNOWLEDGMENTS

The authors thank the whole development team of the NAOMI library for forming the basis of this work, as well as the members of our research group, Computational Molecular Design for code reviewing. This work was supported by DASHH (Data Science in Hamburg - Helmholtz Graduate School for the Structure of Matter) with the Grant-No. HIDSS-0002. Christiane Ehrt and Thorben Reim are funded by Data Science in Hamburg - Helmholtz Graduate School for the Structure of Matter (Grant-No. HIDSS-0002). Sebastian Günther and Alke Meents acknowledge financial support obtained from the Helmholtz Association through the projects FISCOV, SFragX and the Helmholtz Association Impulse and Networking funds InternLabs-0011 "HIR3X". Open Access funding enabled and organized by Projekt DEAL.

#### CONFLICTS OF INTEREST STATEMENT

ProteinsPlus and the NAOMI ChemBioSuite use some methods jointly owned by and/or licensed to BioSolveIT GmbH, Germany. Matthias Rarey is a shareholder of BioSolveIT GmbH. The other authors declare no conflicts of interest.

#### DATA AVAILABILITY STATEMENT

The data that supports the findings of this study are available in the Supporting Information of this article.

#### ORCID

Thorben Reim  <http://orcid.org/0009-0002-7712-8515>

Christiane Ehrt  <http://orcid.org/0000-0003-1428-0042>

Joel Graef  <http://orcid.org/0000-0001-8327-4936>

Sebastian Günther  <https://orcid.org/0000-0002-7329-6653>

Alke Meents  <https://orcid.org/0000-0001-6078-4095>

Matthias Rarey  <http://orcid.org/0000-0002-9553-6531>

#### REFERENCES

- [1] H. M. Berman, *Nucleic Acids Res.* **2000**, 28(1), 235. <https://doi.org/10.1093/nar/28.1.235>
- [2] A. Waterhouse, M. Bertoni, S. Bienert, G. Studer, G. Tauriello, R. Gumienny, F. T. Heer, T. A. P. de Beer, C. Rempfer, L. Bordoli, R. Lepore, T. Schwede, *Nucleic Acids Res.* **2018**, 46(W1), W296. <https://doi.org/10.1093/nar/gky427>
- [3] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Židek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis, S. Velankar, *Nucleic Acids Res.* **2022**, 50(D1), D439. <https://doi.org/10.1093/nar/gkab1061>
- [4] A. Volkamer, M. Rarey, *Future Med. Chem.* **2014**, 6(3), 319. <https://doi.org/10.4155/fmc.14.3>
- [5] A. V. Sadybekov, V. Katritch, *Nature* **2023**, 616(7958), 673. <https://doi.org/10.1038/s41586-023-05905-z>
- [6] A. Volkamer, A. Griewel, T. Grombacher, M. Rarey, *J. Chem. Inf. Model.* **2010**, 50(11), 2041. <https://doi.org/10.1021/ci100241y>
- [7] X. Zheng, L. Gan, E. Wang, J. Wang, *AAPS. J.* **2013**, 15(1), 228. <https://doi.org/10.1208/s122248-012-9426-6>
- [8] N. K. Broomhead, M. E. Soliman, *Cell Biochem. Biophys.* **2017**, 75(1), 15. <https://doi.org/10.1007/s12013-016-0769-y>
- [9] J. Liao, Q. Wang, F. Wu, Z. Huang, *Molecules* **2022**, 27(20), 7103. <https://doi.org/10.3390/molecules27207103>
- [10] J. Graef, C. Ehrt, K. Diedrich, M. Poppinga, N. Ritter, M. Rarey, *J. Med. Chem.* **2022**, 65(2), 1384. <https://doi.org/10.1021/acs.jmedchem.1c01046>
- [11] M. Chartier, R. Najmanovich, *J. Chem. Inf. Model.* **2015**, 55(8), 1600. <https://doi.org/10.1021/acs.jcim.5b00333>
- [12] M. Naderi, J. M. Lemoine, R. G. Govindaraj, O. Z. Kana, W. P. Feinstein, M. Brylinski, *Briefings Bioinf.* **2019**, 20(6), 2167. <https://doi.org/10.1093/bib/bby078>
- [13] M. Eguida, D. Rognan, *Int. J. Mol. Sci.* **2022**, 23(20), 12462. <https://doi.org/10.3390/ijms232012462>
- [14] C. Ehrt, T. Brinkjost, O. Koch, *PLoS Comput. Biol.* **2018**, 14(11), e1006483. <https://doi.org/10.1371/journal.pcbi.1006483>
- [15] L. Pu, R. G. Govindaraj, J. M. Lemoine, H. C. Wu, M. Brylinski, *PLoS Comput. Biol.* **2019**, 15(2), e1006718. <https://doi.org/10.1371/journal.pcbi.1006718>
- [16] M. Simonovsky, J. Meyers, *J. Chem. Inf. Model.* **2020**, 60(4), 2356. <https://doi.org/10.1021/acs.jcim.9b00554>
- [17] Y. C. Chen, R. Tolbert, A. M. Aronov, G. McGaughey, W. P. Walters, L. Meireles, *J. Chem. Inf. Model.* **2016**, 56(9), 1734. <https://doi.org/10.1021/acs.jcim.6b00118>
- [18] C. Ehrt, T. Brinkjost, O. Koch, *MedChemComm* **2019**, 10(7), 1145. <https://doi.org/10.1039/c9md00102f>
- [19] V. Le Guilloux, P. Schmidtke, P. Tuffery, *BMC Bioinformatics* **2009**, 10, 168. <https://doi.org/10.1186/1471-2105-10-168>
- [20] M. Eguida, D. Rognan, *J. Med. Chem.* **2020**, 63(13), 7127. <https://doi.org/10.1021/acs.jmedchem.0c00422>
- [21] J. Desaphy, K. Azdimousa, E. Kellenberger, D. Rognan, *J. Chem. Inf. Model.* **2012**, 52(8), 2287. <https://doi.org/10.1021/ci300184x>
- [22] S. Li, C. Cai, J. Gong, X. Liu, H. Li, *Proteins: Struct. Funct. Bioinf.* **2021**, 89(11), 1541. <https://doi.org/10.1002/prot.26176>
- [23] J. Desaphy, G. Bret, D. Rognan, E. Kellenberger, *Nucleic Acids Res.* **2015**, 43(Database issue), D399. <https://doi.org/10.1093/nar/gku928>
- [24] A. Bhadra, K. Yeturu, *Mach. Learn. Sci. Technol.* **2020**, 2(1), 015005. <https://doi.org/10.1088/2632-2153/abad88>
- [25] M. Gao, J. Skolnick, *Bioinformatics* **2013**, 29(5), 597. <https://doi.org/10.1093/bioinformatics/btt024>
- [26] P. Anand, D. Nagarajan, S. Mukherjee, N. Chandra, *Database* **2014**, 2014, bau029. <https://doi.org/10.1093/database/bau029>
- [27] O. B. Scott, J. Gu, A. W. E. Chan, *J. Chem. Inf. Model.* **2022**, 62(22), 5383. <https://doi.org/10.1021/acs.jcim.2c00832>

- [28] R. A. Laskowski, *J. Mol. Graphics* **1995**, 13(5), 323. [https://doi.org/10.1016/0263-7855\(95\)00073-9](https://doi.org/10.1016/0263-7855(95)00073-9)
- [29] K. E. Choi, A. Balupuri, N. S. Kang, *Comput. Struct. Biotechnol. J.* **2023**, 21, 425. <https://doi.org/10.1016/j.csbj.2022.12.014>
- [30] K. Yeturu, N. Chandra, *BMC Bioinformatics* **2008**, 9, 543. <https://doi.org/10.1186/1471-2105-9-543>
- [31] H. W. Kuhn, *Naval Res. Log. Quar.* **1955**, 2(1–2), 83. <https://doi.org/10.1002/nav.3800020109>
- [32] K. Diedrich, J. Graef, K. Schöning-Stierand, M. Rarey, *Bioinformatics* **2021**, 37(3), 424. <https://doi.org/10.1093/bioinformatics/btaa693>
- [33] T. Inhester, S. Bietz, M. Hilbig, R. Schmidt, M. Rarey, *J. Chem. Inf. Model.* **2017**, 57(2), 148. <https://doi.org/10.1021/acs.jcim.6b00561>
- [34] S. Urbaczek, A. Kolodzik, I. Groth, S. Heuser, M. Rarey, *J. Chem. Inf. Model.* **2013**, 53(1), 76. <https://doi.org/10.1021/ci300358c>
- [35] S. Bietz, T. Inhester, F. Lauck, K. Sommer, M. M. von Behren, R. Fährrolfes, F. Flachsenberg, A. Meyder, E. Nittinger, T. Otto, M. Hilbig, K. T. Schomburg, A. Volkamer, M. Rarey, *J. Biotechnol.* **2017**, 261, 207. <https://doi.org/10.1016/j.jbiotec.2017.06.004>
- [36] S. Urbaczek, A. Kolodzik, J. R. Fischer, T. Lippert, S. Heuser, I. Groth, T. Schulz-Gasch, M. Rarey, *J. Chem. Inf. Model.* **2011**, 51(12), 3199. <https://doi.org/10.1021/ci200324e>
- [37] C. P. Gomes, D. E. Fernandes, F. Casimiro, G. F. da Mata, M. T. Passos, P. Varela, G. Mastroianni-Kirsztajn, J. B. Pesquero, *Front. Cell. Infect. Microbiol.* **2020**, 10, 589505. <https://doi.org/10.3389/fcimb.2020.589505>
- [38] A. Kahraman, R. J. Morris, R. A. Laskowski, J. M. Thornton, *J. Mol. Biol.* **2007**, 368(1), 283. <https://doi.org/10.1016/j.jmb.2007.01.086>
- [39] D. J. Wood, J. Vlieg, M. Wagener, T. Ritschel, *J. Chem. Inf. Model.* **2012**, 52(8), 2031. <https://doi.org/10.1021/ci3000776>
- [40] T. Krotzky, C. Grunwald, U. Egerland, G. Klebe, *J. Chem. Inf. Model.* **2015**, 55(1), 165. <https://doi.org/10.1021/ci5005898>
- [41] N. Weill, D. Rognan, *J. Chem. Inf. Model.* **2010**, 50(1), 123. <https://doi.org/10.1021/ci900349y>
- [42] Y. Zhang, *Nucleic Acids Res.* **2005**, 33(7), 2302. <https://doi.org/10.1093/nar/gki524>
- [43] J. Konc, D. Janežič, *Bioinformatics* **2010**, 26(9), 1160. <https://doi.org/10.1093/bioinformatics/btq100>
- [44] G. Marcou, D. Rognan, *J. Chem. Inf. Model.* **2007**, 47(1), 195. <https://doi.org/10.1021/ci600342e>
- [45] J. Desaphy, E. Raimbaud, P. Ducrot, D. Rognan, *J. Chem. Inf. Model.* **2013**, 53(3), 623. <https://doi.org/10.1021/ci300566n>
- [46] J. Batista, P. C. Hawkins, R. Tolbert, M. T. Geballe, *J. Cheminf.* **2014**, 6(S1), P57. <https://doi.org/10.1186/1758-2946-6-s1-p57>
- [47] S. Schmitt, D. Kuhn, G. Klebe, *J. Mol. Biol.* **2002**, 323(2), 387. [https://doi.org/10.1016/s0022-2836\(02\)00811-2](https://doi.org/10.1016/s0022-2836(02)00811-2)
- [48] L. Xie, L. Xie, P. E. Bourne, *Bioinformatics* **2009**, 25(12), i305. <https://doi.org/10.1093/bioinformatics/btp220>
- [49] A. Shulman-Peleg, R. Nussinov, H. J. Wolfson, *J. Mol. Biol.* **2004**, 339(3), 607. <https://doi.org/10.1016/j.jmb.2004.04.012>
- [50] C. Schalon, J. S. Sargand, E. Kellenberger, D. Rognan, *Proteins Struct. Funct. Bioinf.* **2008**, 71(4), 1755. <https://doi.org/10.1002/prot.21858>
- [51] S. Barelier, T. Sterling, M. J. O'Meara, B. K. Shoichet, *ACS Chem. Biol.* **2015**, 10(12), 2772. <https://doi.org/10.1021/acschembio.5b00683>
- [52] S. F. OpenEye Scientific Software, NM. **2023**. *Shape Toolkit*. <http://www.eyesopen.com>
- [53] S. F. OpenEye Scientific Software, NM. **2023**. *Spicoli Toolkit*. <http://www.eyesopen.com>
- [54] S. F. OpenEye Scientific Software, NM. **2023**. *ROCS*. <http://www.eyesopen.com>
- [55] M. A. Fligner, J. S. Verducci, P. E. Blower, *Technometrics* **2002**, 44(2), 110. <https://doi.org/10.1198/004017002317375064>
- [56] C. Ehrhart, *The Impact of Binding Site Similarity on Hit Identification in Early Drug Discovery*, TU Dortmund University, Dortmund, Germany **2019**.
- [57] J. Graef, C. Ehrhart, M. Rarey, *J. Chem. Inf. Model.* **2023**, 63(10), 3128. <https://doi.org/10.1021/acs.jcim.3c00336>
- [58] B. Zdrzil, E. Felix, F. Hunter, E. J. Manners, J. Blackshaw, S. Corbett, M. de Veijs, H. Ioannidis, D. M. Lopez, J. F. Mosquera, M. P. Magarinos, N. Bosc, R. Arcila, T. Kizilören, A. Gaulton, A. P. Bento, M. F. Adasme, P. Monecke, G. A. Landrum, A. R. Leach, *Nucleic Acids Res.* **2023**, 52, D1180. <https://doi.org/10.1093/nar/gkad1004>
- [59] M. Novinec, B. Lenarčič, *BioMol. Concepts* **2013**, 4(3), 287. <https://doi.org/10.1515/bmc-2012-0054>
- [60] X. Ou, Y. Liu, X. Lei, P. Li, D. Mi, L. Ren, L. Guo, R. Guo, T. Chen, J. Hu, Z. Xiang, Z. Mu, X. Chen, J. Chen, K. Hu, Q. Jin, J. Wang, Z. Qian, *Nat. Commun.* **2020**, 11(1), 1620. <https://doi.org/10.1038/s41467-020-15562-9>
- [61] P. Y. A. Reinke, E. E. de Souza, S. Günther, S. Falke, J. Lieske, W. Ewert, J. Loboda, A. Herrmann, A. Rahmani Mashhour, K. Karničar, A. Usenik, N. Lindič, A. Sekirnik, V. F. Botosso, G. M. M. Santelli, J. Kapronezai, M. V. de Araújo, T. T. Silva-Pereira, A. F. S. Filho, M. S. Tavares, L. Flórez-Álvarez, D. B. L. de Oliveira, E. L. Durigon, P. R. Giaretta, M. B. Heinemann, M. Hauser, B. Seychell, H. Böhrer, W. Rut, M. Drag, T. Beck, R. Cox, H. N. Chapman, C. Betzel, W. Brehm, W. Hinrichs, G. Ebert, S. L. Latham, A. M. S. Guimarães, D. Turk, C. Wrenger, A. Meents, *Commun. Biol.* **2023**, 6(1), 1058. <https://doi.org/10.1038/s42003-023-05317-9>
- [62] G. Wang, R. L. Dunbrack Jr., *Bioinformatics* **2003**, 19(12), 1589. <https://doi.org/10.1093/bioinformatics/btg224>
- [63] F. Madeira, M. Pearce, A. R. N. Tivey, P. Basutkar, J. Lee, O. Edbali, N. Madhusoodanan, A. Kolesnikov, R. Lopez, *Nucleic Acids Res.* **2022**, 50(W1), W276. <https://doi.org/10.1093/nar/gkac240>
- [64] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, T. E. Ferrin, *J. Comput. Chem.* **2004**, 25(13), 1605. <https://doi.org/10.1002/jcc.20084>
- [65] J. Kyte, R. F. Doolittle, *J. Mol. Biol.* **1982**, 157(1), 105. [https://doi.org/10.1016/0022-2836\(82\)90515-0](https://doi.org/10.1016/0022-2836(82)90515-0)
- [66] T. D. Goddard, C. C. Huang, E. C. Meng, E. F. Pettersen, G. S. Couch, J. H. Morris, T. E. Ferrin, *Protein Sci.* **2018**, 27(1), 14. <https://doi.org/10.1002/pro.3235>
- [67] L. A. Hardegger, B. Kuhn, B. Spinnler, L. Anselm, R. Ecabert, M. Stihle, B. Gsell, R. Thoma, J. Diez, J. Benz, J. M. Plancher, G. Hartmann, D. W. Banner, W. Haap, F. Diederich, *Angew. Chem. Int. Ed.* **2011**, 50(1), 314. <https://doi.org/10.1002/anie.201006781>
- [68] S. Bietz, S. Urbaczek, B. Schulz, M. Rarey, *J. Cheminf.* **2014**, 6, 12. <https://doi.org/10.1186/1758-2946-6-12>
- [69] G. Gaël, J. Benoît, **2010**. *Eigen v3 (C++ library)*. <http://eigen.tuxfamily.org>
- [70] S. Umeyama, *IEEE. Trans. Pattern. Anal. Mach. Intell.* **1991**, 13(4), 376. <https://doi.org/10.1109/34.88573>
- [71] E. Calenoff, *ISRN Neurol.* **2012**, 2012, 1. <https://doi.org/10.5402/2012/851541>
- [72] S. Beisken, T. Meinel, B. Wiswedel, L. F. de Figueiredo, M. Berthold, C. Steinbeck, *BMC Bioinformatics* **2013**, 14, 257. <https://doi.org/10.1186/1471-2105-14-257>

## SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**How to cite this article:** T. Reim, C. Ehrhart, J. Graef, S. Günther, A. Meents, M. Rarey, *Arch. Pharm.* **2024**, e2300661. <https://doi.org/10.1002/ardp.202300661>

# Supporting Information

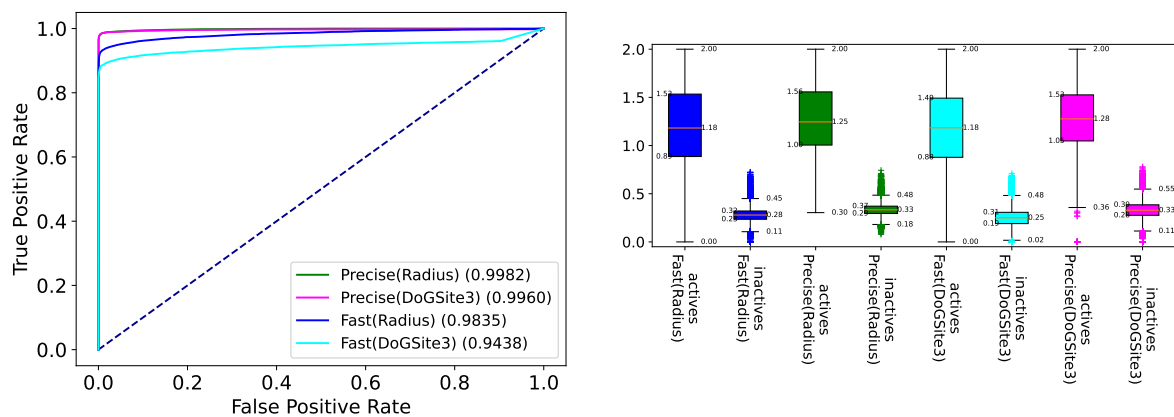
## SiteMine: Large Scale Binding Site Similarity Searching in Protein Structure Databases

Thorben Reim,<sup>†</sup> Christiane Ehart,<sup>†</sup> Joel Graef,<sup>†</sup> Sebastian Günther,<sup>‡</sup> Alke  
Meents,<sup>‡</sup> and Matthias Rarey<sup>\*,†</sup>

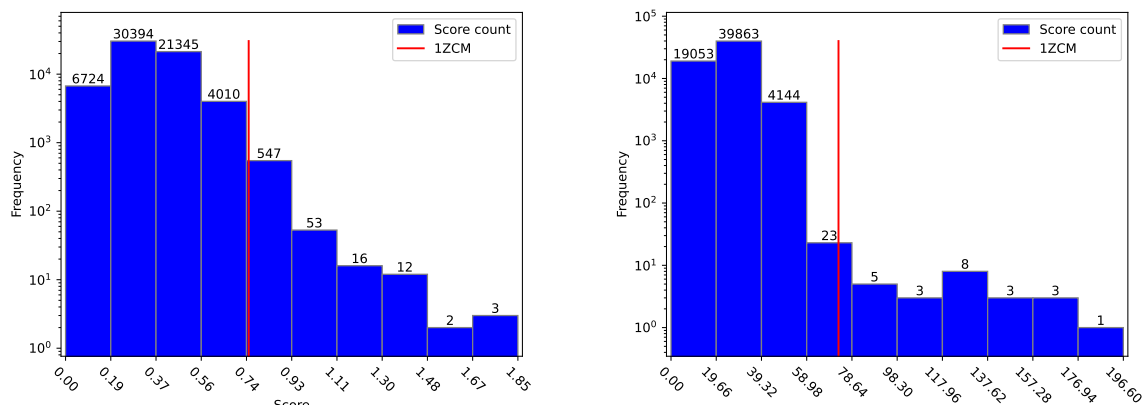
<sup>†</sup>*Universität Hamburg, ZBH - Center for Bioinformatics, Germany*

<sup>‡</sup>*Center for Free-Electron Laser Science, Deutsches Elektronen-Synchrotron DESY,  
Hamburg, Germany*

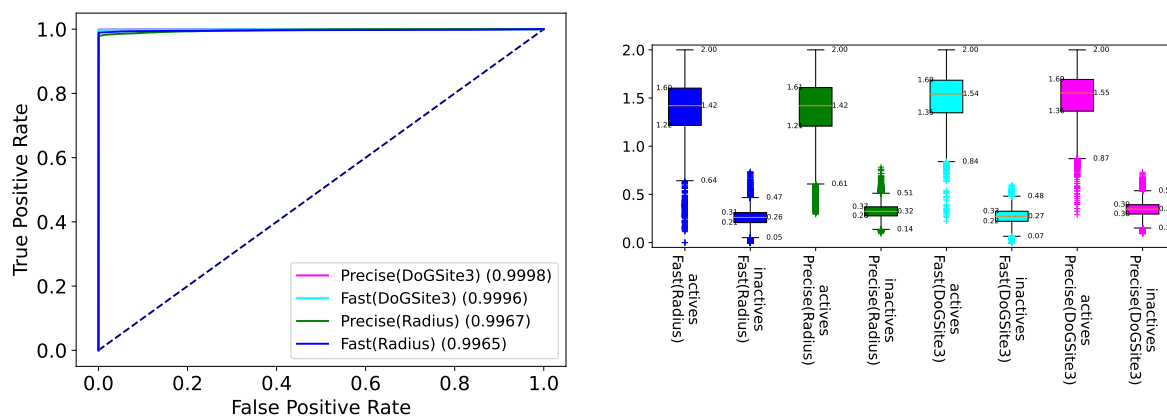
E-mail: [matthias.rarey@uni-hamburg.de](mailto:matthias.rarey@uni-hamburg.de)



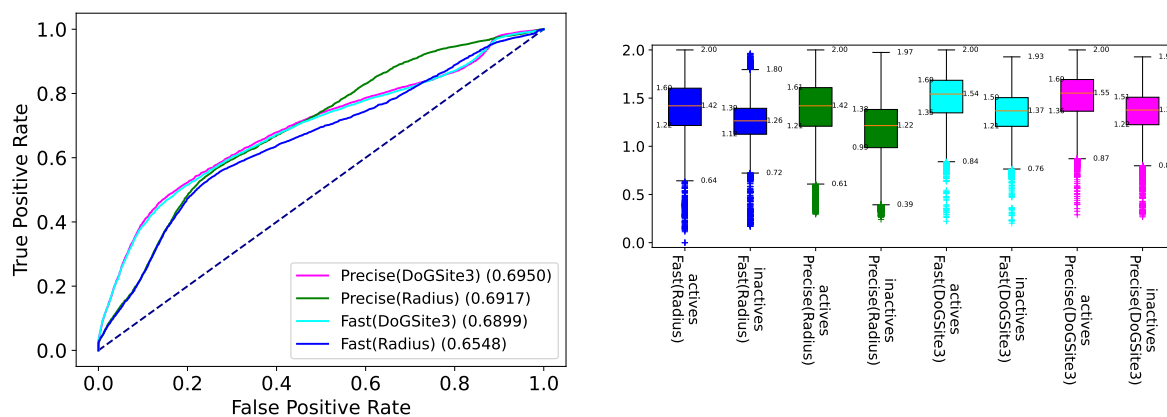
**Figure S1:** ROC curve (left) and boxplots (left) of the score distributions for the SiteMine settings (*Fast*, *Precise*) and binding site definition (ligand-radius and DoGSite3<sup>1</sup>) for the *NMR Structures* data set.



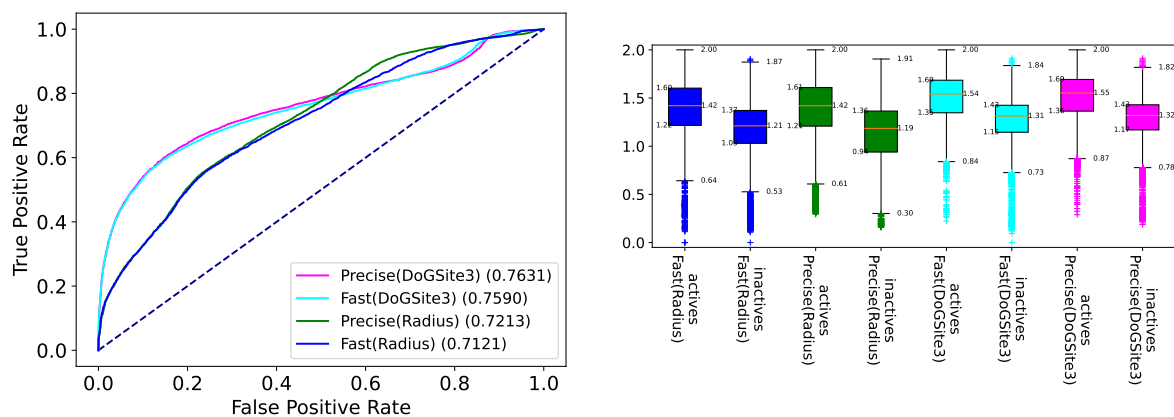
**Figure S2:** The similarity score distributions of normalized by the larger binding site (left) and non-normalize (right) scores for the search for the similar binding site of cathepsin L's (PDB entry 2xu1) active site. The vertical red line displays calpain 1's (PDB entry 1zcm) active site similarity score within the distribution.



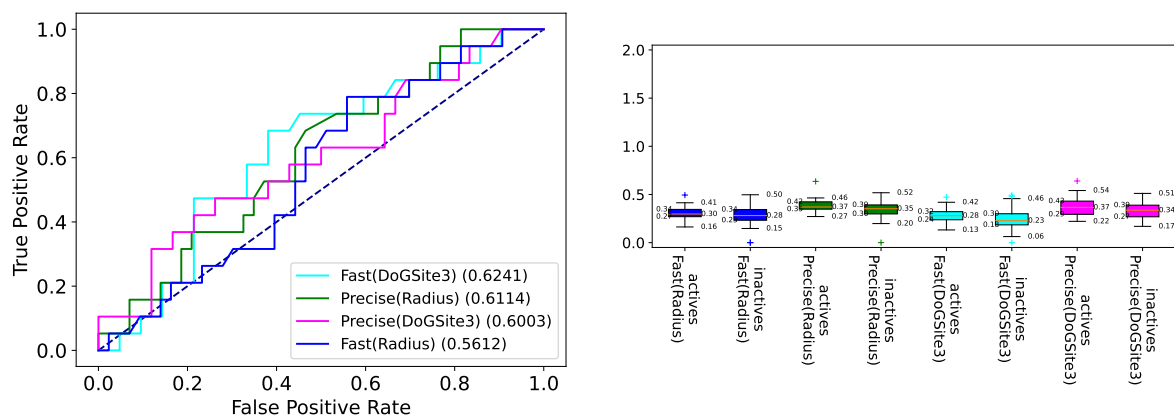
**Figure S3:** ROC curve (left) and boxplots (left) of the score distributions for the SiteMine settings (*Fast*, *Precise*) and binding site definition (ligand-radius and DoGSite3) for the *Structures with Identical Sequences* data set.



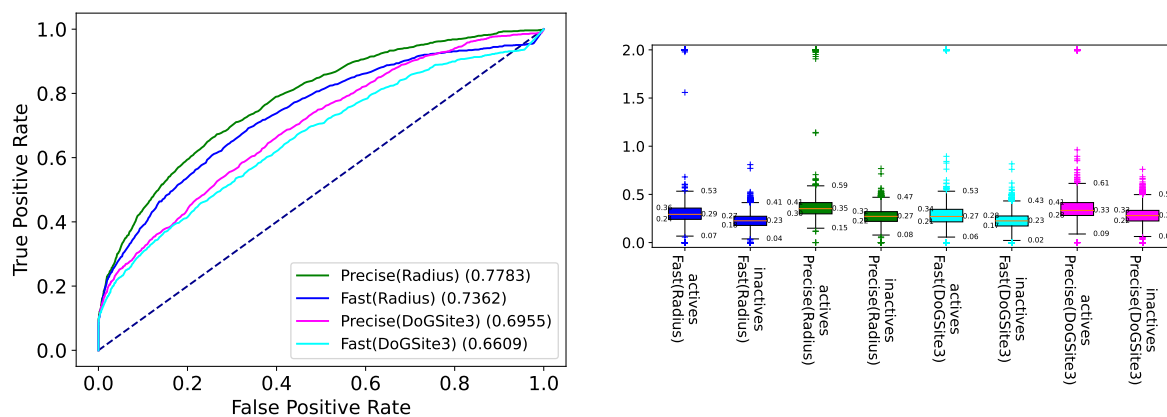
**Figure S4:** ROC curve (left) and boxplots (left) of the score distributions for the SiteMine settings (*Fast*, *Precise*) and binding site definition (ligand-radius and DoGSite3) for the *Decoy Structures Rational* data set.



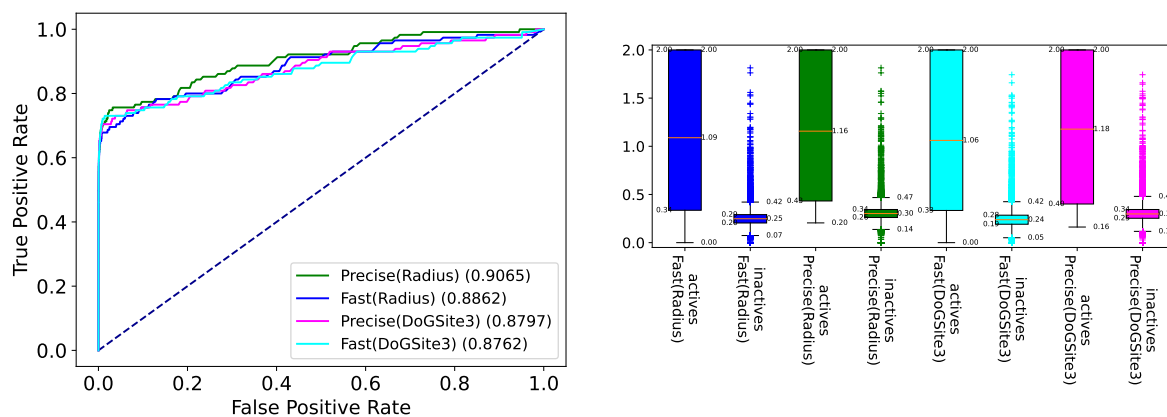
**Figure S5:** ROC curve (left) and boxplots (left) of the score distributions for the SiteMine settings (*Fast*, *Precise*) and binding site definition (ligand-radius and DoGSite3) for the *Decoy Structures Shape* data set.



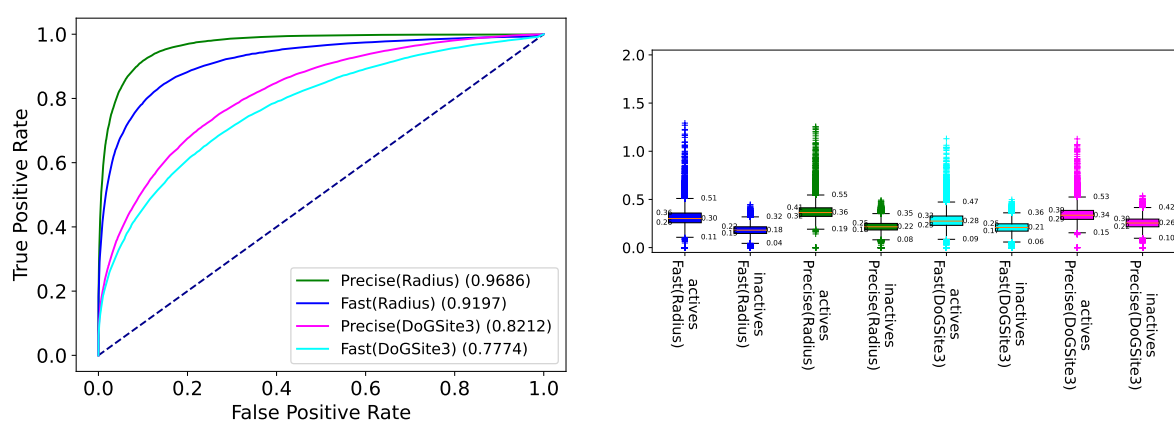
**Figure S6:** ROC curve (left) and boxplots (left) of the score distributions for the SiteMine settings (*Fast*, *Precise*) and binding site definition (ligand-radius and DoGSite3) for the *Barelier<sup>2</sup>* data set.



**Figure S7:** ROC curve (left) and boxplots (left) of the score distributions for the SiteMine settings (*Fast*, *Precise*) and binding site definition (ligand-radius and DoGSite3) for the *Kahraman*<sup>3,4</sup> data set.



**Figure S8:** ROC curve (left) and boxplots (left) of the score distributions for the SiteMine settings (*Fast*, *Precise*) and binding site definition (ligand-radius and DoGSite3) for the *Successful Applications* data set.



**Figure S9:** ROC curve (left) and boxplots (left) of the score distributions for the SiteMine settings (*Fast*, *Precise*) and binding site definition (ligand-radius and DoGSite3) for the *ROCS Structures* data set.



**Table S1:** Results for SiteMine (SP-score) and the benchmark study<sup>5</sup> tools for the *Barelier*<sup>2</sup> data set descending sorted by the enrichment factors (EF).

Method	EFs							AUC
	1.6%	8.1%	16.1%	32.3%	48.4%	64.5%	80.6%	
SMAP <sup>6</sup>	3.26	2.61	2.28	1.63	1.31	1.14	0.98	0.68
SiteMine <i>Precise</i>	3.26	1.63	1.31	1.14	1.26	1.14	1.11	0.61
VolSite/Shaper <sup>7</sup>	3.26	1.31	1.31	1.47	1.41	1.22	1.24	0.71
FuzCav <sup>8</sup>	3.26	1.31	0.98	1.31	1.31	1.22	1.24	0.67
FuzCav (PDB)	3.26	1.31	0.98	1.31	1.31	1.14	1.24	0.65
Shaper (PDB)	3.26	1.31	0.65	0.65	0.98	1.14	1.17	0.54
Shaper	3.26	1.31	0.65	0.65	0.98	1.06	1.17	0.54
TM-align <sup>9</sup>	3.26	0.65	1.31	1.14	0.98	1.06	1.17	0.59
Cavbase <sup>10,11</sup>	3.26	0.65	1.31	0.82	1.09	0.98	1.04	0.55
TIFP <sup>12</sup> (PDB)	3.26	0.65	0.98	1.31	0.98	0.82	0.65	0.56
TIFP	3.26	0.65	0.33	0.82	1.20	1.06	0.85	0.55
ProBiS <sup>13</sup>	3.26	0.65	0.33	0.16	0.11	0.08	0.52	0.50
SiteMine <i>Fast</i>	0.00	1.09	0.98	0.98	1.16	1.22	1.11	0.56
IsoMIF <sup>14</sup>	0.00	0.65	1.31	1.31	1.09	1.06	1.24	0.62
VolSite/Shaper (PDB)	0.00	0.65	0.98	0.98	0.98	0.90	0.72	0.50
PocketMatch <sup>15</sup>	0.00	0.65	0.98	0.98	0.98	0.73	0.59	0.51
SiteHopper <sup>16</sup>	0.00	0.65	0.65	1.14	1.09	1.14	1.11	0.56
SiteEngine <sup>17</sup>	0.00	0.65	0.65	0.98	1.09	1.06	1.11	0.55
RAPMAD <sup>18</sup>	0.00	0.00	0.33	1.14	1.09	1.14	1.17	0.60
Grim <sup>12</sup> (PDB)	0.00	0.00	0.33	0.82	0.98	0.82	0.65	0.45
Grim	0.00	0.00	0.33	0.65	0.87	0.98	0.85	0.45
SiteAlign <sup>19</sup>	0.00	0.00	0.33	0.65	0.76	0.98	1.04	0.44
KRIPO <sup>20</sup>	0.00	0.00	0.00	2.63	1.75	1.32	2.11	0.73

**Table S2:** Results for SiteMine (SP-score) and the benchmark study<sup>5</sup> tools for the *Decoy Structures Rational* data set descending sorted by the enrichment factors (EF).

Method	EFs							AUC
	0.1%	0.5%	1%	2%	3%	4%	5%	
SiteMine <i>Fast</i>	2.00	2.00	2.00	1.81	1.66	1.61	1.56	0.65
SiteMine <i>Precise</i>	2.00	2.00	2.00	1.78	1.64	1.61	1.59	0.69
SiteAlign	1.94	2.00	2.00	2.00	2.00	2.00	2.00	0.85
FuzCav	1.94	2.00	2.00	1.94	1.87	1.80	1.78	0.69
FuzCav(PDB)	1.94	2.00	2.00	1.94	1.87	1.80	1.78	0.69
SiteHopper	1.94	2.00	2.00	1.86	1.79	1.79	1.80	0.75
RAPMAD	1.94	2.00	2.00	1.72	1.70	1.70	1.69	0.61
VolSite/Shaper	1.94	2.00	2.00	1.68	1.66	1.66	1.68	0.68
VolSite/Shaper(PDB)	1.94	2.00	2.00	1.68	1.66	1.65	1.68	0.68
KRIPO	1.94	2.00	2.00	1.64	1.57	1.54	1.50	0.60
Shaper	1.94	2.00	2.00	1.63	1.61	1.64	1.66	0.71
Shaper(PDB)	1.94	2.00	2.00	1.62	1.61	1.64	1.64	0.71
IsoMIF	1.94	2.00	2.00	1.58	1.51	1.49	1.47	0.59
SiteEngine	1.94	1.98	1.95	1.83	1.61	1.55	1.53	0.82
Grim(PDB)	1.94	1.97	1.83	1.63	1.53	1.38	1.30	0.57
TIFP	1.71	1.95	1.97	1.62	1.54	1.54	1.51	0.66
ProBiS	1.71	1.53	1.21	1.00	1.00	1.03	0.99	0.47
Grim	1.64	1.30	1.23	1.34	1.42	1.38	1.38	0.55
Cavbase	1.41	1.65	1.36	1.05	1.13	0.97	1.03	0.65
SMAP	1.19	1.82	1.88	1.93	1.94	1.95	1.95	0.76
TIFP(PDB)	0.00	1.09	1.54	1.58	1.49	1.44	1.34	0.56
PocketMatch	0.00	0.00	0.89	1.45	1.63	1.64	1.58	0.59
TM-align	0.00	0.00	0.00	0.79	0.94	0.84	0.78	0.49

**Table S3:** Results for SiteMine (SP-score) and the benchmark study<sup>5</sup> tools for the *Decoy Structures Shape* data set descending sorted by the enrichment factors (EF).

Method	EFs							AUC
	0.1%	0.5%	1%	2%	3%	4%	5%	
SiteMine <i>Fast</i>	2.00	2.00	2.00	1.96	1.91	1.90	1.90	0.71
SiteMine <i>Precise</i>	2.00	2.00	2.00	1.96	1.89	1.87	1.87	0.72
SiteAlign	1.94	2.00	2.00	1.96	1.94	1.88	1.88	0.80
SiteHopper	1.94	2.00	2.00	1.92	1.88	1.86	1.84	0.75
VolSite/Shaper	1.94	2.00	2.00	1.83	1.78	1.76	1.77	0.76
VolSite/Shaper(PDB)	1.94	2.00	2.00	1.82	1.78	1.76	1.77	0.76
Shaper(PDB)	1.94	2.00	2.00	1.75	1.71	1.70	1.71	0.76
Shaper	1.94	2.00	2.00	1.75	1.70	1.71	1.71	0.76
RAPMAD	1.94	2.00	2.00	1.74	1.72	1.72	1.72	0.63
IsoMIF	1.94	2.00	2.00	1.69	1.63	1.60	1.58	0.59
KRIPO	1.94	2.00	2.00	1.58	1.42	1.36	1.32	0.61
SiteEngine	1.94	1.98	1.98	1.99	1.93	1.87	1.82	0.79
Grim(PDB)	1.94	1.77	1.82	1.59	1.58	1.38	1.30	0.56
Cavbase	1.71	1.62	1.63	1.08	1.17	1.05	0.99	0.64
ProBiS	1.64	1.71	1.27	0.89	0.95	0.99	0.95	0.46
Grim	1.64	1.33	1.36	1.43	1.43	1.39	1.38	0.56
FuzCav	1.49	1.91	1.95	1.79	1.56	1.50	1.50	0.58
FuzCav(PDB)	1.49	1.91	1.95	1.79	1.56	1.50	1.50	0.58
SMAP	1.19	1.85	1.92	1.94	1.90	1.87	1.82	0.65
TIFP	0.82	1.77	1.88	1.66	1.58	1.58	1.54	0.66
TIFP(PDB)	0.00	1.00	1.50	1.55	1.48	1.45	1.39	0.57
PocketMatch	0.00	0.00	0.83	1.42	1.61	1.64	1.58	0.57
TM-align	0.00	0.00	0.00	0.79	0.94	0.84	0.78	0.49

**Table S4:** Results for SiteMine (SP-score) and the benchmark study<sup>5</sup> tools for the *Structures with Identical Sequences* data set descending sorted by the enrichment factors (EF).

Method	EFs							AUC
	0.1%	0.5%	1%	2%	3%	4%	5%	
SiteMine <i>Fast</i>	7.91	7.91	7.91	7.91	7.91	7.91	7.91	1.00
SiteMine <i>Precise</i>	7.91	7.91	7.91	7.91	7.91	7.91	7.91	1.00
ProBiS	7.89	7.91	7.91	7.91	7.91	7.91	7.91	1.00
SMAP	7.89	7.91	7.91	7.91	7.91	7.91	7.91	1.00
TM-align	7.89	7.91	7.91	7.91	7.91	7.91	7.91	1.00
Cavbase	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.98
SiteHopper	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.98
SiteAlign	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.97
Shaper	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.96
Shaper(PDB)	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.96
SiteEngine	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.96
FuzCav	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.94
FuzCav(PDB)	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.94
VolSite/Shaper(PDB)	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.94
VolSite/Shaper	7.89	7.91	7.91	7.91	7.91	7.91	7.91	0.93
KRIPO	7.89	7.91	7.91	7.91	7.91	7.90	7.89	0.91
RAPMAD	7.89	7.91	7.91	7.91	7.79	7.45	6.87	0.85
IsoMIF	7.89	7.91	7.83	7.49	7.08	6.61	6.14	0.77
Grim	7.89	7.91	7.76	7.78	7.38	6.53	5.74	0.69
TIFP	7.89	7.22	5.37	4.37	4.00	3.71	3.40	0.66
PocketMatch	7.59	7.85	7.88	7.90	7.90	7.90	7.88	0.82
Grim(PDB)	6.25	6.76	5.51	4.47	4.04	3.59	3.31	0.62
TIFP(PDB)	1.64	5.51	4.10	2.88	2.37	2.16	2.05	0.55

**Table S5:** Results for SiteMine (SP-score) and the benchmark study<sup>5</sup> tools for the *Kahraman*<sup>3,4</sup> data set descending sorted by the enrichment factors (EF).

Method	EFs							AUC
	0.1%	0.5%	1%	2%	3%	4%	5%	
SiteMine <i>Precise</i>	7.58	7.58	7.50	6.22	5.49	4.99	4.70	0.78
SiteMine <i>Fast</i>	7.58	7.58	7.50	6.03	5.46	4.95	4.57	0.74
TIFP	6.52	6.96	6.96	6.03	5.04	4.29	4.02	0.71
KRIPO	6.52	6.96	6.96	5.87	5.07	4.62	4.22	0.76
SiteHopper	6.52	6.96	6.96	5.82	4.82	4.21	3.76	0.72
TM-align	6.52	6.96	6.96	5.76	4.64	3.89	3.57	0.66
IsoMIF	6.52	6.96	6.96	5.43	5.18	5.14	4.76	0.75
SiteEngine	6.52	6.96	6.96	5.38	4.35	3.64	3.17	0.64
PocketMatch	6.52	6.96	6.96	5.33	4.20	3.75	3.57	0.66
Shaper(PDB)	6.52	6.96	6.96	5.22	4.49	4.02	3.70	0.66
Shaper	6.52	6.96	6.96	5.11	4.35	3.94	3.48	0.65
SMAP	6.52	6.96	6.96	5.11	3.91	3.07	2.78	0.62
ProBiS	6.52	6.96	6.96	5.05	3.73	3.34	2.91	0.54
SiteAlign	6.52	6.96	6.96	4.89	3.80	3.10	2.70	0.59
Cavbase	6.52	6.96	6.96	4.89	3.51	3.23	2.89	0.60
RAPMAD	6.52	6.96	6.96	4.89	3.48	2.72	2.35	0.55
FuzCav(PDB)	6.52	6.96	6.96	4.89	3.33	2.88	2.43	0.56
FuzCav	6.52	6.96	6.96	4.89	3.33	2.83	2.35	0.55
VolSite/Shaper	6.52	6.96	6.52	4.13	3.15	2.74	2.41	0.56
Grim	6.52	6.96	6.20	5.87	5.36	4.73	4.39	0.69
VolSite/Shaper(PDB)	6.52	6.96	6.09	4.02	3.12	2.77	2.50	0.57
TIFP(PDB)	0.00	3.91	5.33	3.32	2.57	2.53	2.54	0.54
Grim(PDB)	0.00	2.39	4.67	3.75	3.44	2.66	2.13	0.61

**Table S6:** Results for SiteMine (SP-score) and the benchmark study<sup>5</sup> tools for the *NMR Structures* data set descending sorted by the enrichment factors (EF).

Method	EFs							AUC
	0.1%	0.5%	1%	2%	3%	4%	5%	
SiteMine <i>Precise</i>	14.00	14.00	14.00	14.00	14.00	14.00	14.00	1.00
SiteMine <i>Fast</i>	14.00	14.00	14.00	14.00	14.00	14.00	14.00	0.98
SiteAlign	13.97	14.00	14.00	14.00	14.00	14.00	14.00	1.00
SiteEngine	13.97	14.00	14.00	14.00	14.00	14.00	14.00	1.00
SiteHopper	13.97	14.00	14.00	14.00	14.00	14.00	14.00	1.00
SMAP	13.97	14.00	14.00	14.00	14.00	14.00	14.00	1.00
TM-align	13.97	14.00	14.00	14.00	14.00	14.00	14.00	1.00
FuzCav	13.97	14.00	14.00	14.00	14.00	14.00	13.96	0.99
FuzCav(PDB)	13.97	14.00	14.00	14.00	14.00	13.99	13.94	0.98
PocketMatch	13.97	14.00	14.00	14.00	14.00	13.94	13.50	0.96
Grim	13.97	14.00	14.00	14.00	13.99	13.83	12.58	0.92
Cavbase	13.97	14.00	14.00	14.00	13.99	13.14	12.26	0.87
VolSite/Shaper	13.97	14.00	14.00	14.00	13.96	13.61	12.31	0.78
KRIPO	13.97	14.00	14.00	14.00	13.94	13.61	12.95	0.96
VolSite/Shaper(PDB)	13.97	14.00	14.00	14.00	13.94	13.24	11.64	0.76
Shaper(PDB)	13.97	14.00	14.00	13.96	13.84	13.22	12.16	0.93
Shaper	13.97	14.00	14.00	13.96	13.84	13.16	12.05	0.93
RAPMAD	13.97	14.00	13.92	12.93	11.71	10.75	9.88	0.82
ProBiS	13.97	13.97	13.99	13.99	14.00	14.00	14.00	1.00
IsoMIF	13.97	13.92	13.90	13.28	11.81	10.24	8.92	0.70
Grim(PDB)	13.97	13.82	13.91	13.51	13.36	12.55	11.26	0.85
TIFP	13.97	13.64	13.05	11.74	10.80	9.56	8.38	0.91
TIFP(PDB)	5.95	12.39	12.63	11.24	9.85	8.80	7.77	0.78

**Table S7:** Results for SiteMine (SP-score) and the benchmark study<sup>5</sup> tools for the *Successful Applications* data set descending sorted by the enrichment factors (EF).

Method	EFs							AUC
	0.1%	0.5%	1%	2%	3%	4%	5%	
TM-align	434.78	135.65	68.70	37.83	26.38	19.78	15.83	0.88
PocketMatch	434.78	104.35	59.13	32.17	22.03	16.52	13.91	0.82
SiteAlign	426.09	116.52	60.00	30.87	20.87	16.52	13.39	0.87
KRIPO	426.09	106.09	56.52	30.87	20.58	16.30	13.04	0.85
SiteMine <i>Precise</i>	421.59	132.17	69.56	36.09	24.93	18.91	15.13	0.91
SiteMine <i>Fast</i>	421.59	128.69	67.82	33.91	23.19	17.39	14.26	0.89
Shaper(PDB)	417.39	100.87	53.04	26.96	18.84	14.13	11.30	0.75
Shaper	417.39	99.13	53.04	26.96	18.26	13.91	11.30	0.75
IsoMIF	408.70	107.83	60.00	30.87	21.16	15.87	13.04	0.87
VolSite/Shaper	408.70	107.83	55.65	27.83	18.84	14.35	11.65	0.77
FuzCav	408.70	88.70	44.35	25.65	18.84	15.00	12.00	0.77
FuzCav(PDB)	408.70	85.22	44.35	24.78	18.26	15.00	12.00	0.77
SiteHopper	400.00	114.78	60.00	30.43	20.58	15.43	12.35	0.77
VolSite/Shaper(PDB)	400.00	102.61	52.17	26.09	17.39	13.04	10.61	0.72
RAPMAD	400.00	83.48	41.74	20.87	13.91	10.43	8.70	0.74
ProBiS	382.61	137.39	70.43	36.96	24.64	18.48	14.78	0.85
SiteEngine	373.91	118.26	65.22	32.61	22.03	16.74	13.57	0.86
Cavbase	339.13	118.26	62.61	32.17	21.45	16.52	13.22	0.82
TIFP	330.43	69.57	36.52	19.13	13.33	10.43	8.70	0.71
SMAP	313.04	118.26	65.22	34.35	23.19	17.83	14.43	0.86
Grim	313.04	76.52	40.00	20.87	14.20	11.09	9.39	0.70
Grim(PDB)	52.17	62.61	31.30	17.39	12.17	9.13	7.65	0.64
TIFP(PDB)	0.00	64.35	33.91	16.96	11.88	9.35	7.83	0.66

**Table S8:** Results for SiteMine (SP-score) and the benchmark study<sup>5</sup> tools for the *ROCS Structures* data set descending sorted by the enrichment factors (EF).

Method	EFs							AUC
	0.1%	0.5%	1%	2%	3%	4%	5%	
SiteHopper	4.67	4.67	4.67	4.67	4.67	4.67	4.67	0.97
SiteMine <i>Fast</i>	4.67	4.67	4.67	4.67	4.65	4.63	4.58	0.92
SiteMine <i>Precise</i>	4.67	4.67	4.67	4.66	4.65	4.64	4.62	0.97
TM-Align	4.67	4.67	4.67	4.66	4.62	4.56	4.47	0.79
SMAP(RawScore)	4.67	4.67	4.25	3.13	2.46	2.08	1.84	0.56
ProBis(AlignmentScore)	4.67	4.61	4.29	3.56	3.23	2.91	2.63	0.52
ProBis(Zscore)	4.67	4.61	4.29	3.56	3.23	2.65	2.31	0.51
KRIPO	4.66	4.64	4.58	4.51	4.35	4.27	4.16	0.93
Cavbase	4.52	4.52	4.29	3.99	3.91	3.80	3.47	0.58
Grim	4.43	4.39	4.38	4.21	4.05	3.91	3.79	0.76
Shaper(Combo/PDB)	4.41	3.46	3.07	2.79	2.61	2.41	2.33	0.68
Shaper(Combo)	4.32	3.32	2.95	2.70	2.52	2.35	2.26	0.67
SiteEngine(Curvatur)	4.25	3.27	2.77	2.38	2.15	2.00	1.88	0.55
PocketMatch	4.00	3.84	3.76	3.71	3.57	3.41	3.30	0.78
Shaper(Color)	3.78	2.86	2.60	2.13	1.99	1.92	1.86	0.62
TIFP	3.75	3.56	3.11	2.64	2.27	2.03	1.89	0.58
Shaper(Color/PDB)	3.69	2.93	2.69	2.23	2.11	2.01	1.93	0.63
TIFP(PDB)	3.49	3.29	2.88	2.47	2.18	1.94	1.83	0.56
VolSite/Shaper(Combo)	3.02	3.25	3.06	2.78	2.62	2.51	2.43	0.68
Grim(PDB)	2.71	3.57	3.66	3.69	3.63	3.54	3.46	0.76
RAPMAD	2.59	0.97	0.68	0.62	0.61	0.62	0.60	0.39
VolSite/Shaper(Color)	2.23	2.29	2.23	2.05	2.01	1.98	1.95	0.63
VolSite/Shaper(Color/PDB)	1.84	2.12	2.10	2.01	1.96	1.97	1.98	0.65
SiteAlign(D3)	0.93	0.89	0.90	0.98	1.01	1.01	1.02	0.55
IsoMIF	0.72	1.03	2.77	3.65	3.96	4.10	4.20	0.97
Fuzcav(PDB)	0.00	0.50	0.72	0.96	1.10	1.18	1.18	0.57
Fuzcav	0.00	0.47	0.64	0.89	1.03	1.11	1.15	0.57



**Table S9:** Mean Spearman’s Rho correlation coefficients for the *Decoy Structures Rational* (left) and the *Decoy Structures Shape* (right) data set for SiteMine and the tools evaluated in an earlier benchmark study.<sup>5</sup>

Rank	Tool	Mean	Rank	Tool	Mean
1	SMAP	-0.884	1	SMAP	-0.904
2	SiteHopper	-0.872	2	<b>SiteMine Precise</b>	-0.888
3	SiteAlign	-0.830	3	<b>SiteMine Fast</b>	-0.880
4	FuzCav (PDB)	-0.759	4	SiteHopper	-0.864
5	FuzCav	-0.758	5	SiteEngine	-0.804
6	<b>SiteMine Precise</b>	-0.743	6	Cavbase	-0.735
7	<b>SiteMine Fast</b>	-0.730	7	Shaper (PDB)	-0.719
8	SiteEngine	-0.726	8	KRIPO	-0.698
9	VolSite/Shaper (PDB)	-0.724	9	Shaper	-0.690
10	VolSite/Shaper	-0.723	10	SiteAlign	-0.683
11	Cavbase	-0.707	11	VolSite/Shaper (PDB)	-0.663
12	Shaper (PDB)	-0.706	12	VolSite/Shaper	-0.662
13	KRIPO	-0.689	13	Grim	-0.624
14	Shaper	-0.677	14	FuzCav	-0.598
15	RAPMAD	-0.657	15	FuzCav (PDB)	-0.598
16	Grim	-0.620	16	IsoMIF	-0.580
17	Grim (PDB)	-0.555	17	Grim (PDB)	-0.553
18	IsoMIF	-0.544	18	RAPMAD	-0.545
19	TIFP (PDB)	-0.491	19	TIFP (PDB)	-0.504
20	TIFP	-0.473	20	TIFP	-0.490
21	PocketMatch	-0.471	21	PocketMatch	-0.476
22	ProBiS	-0.248	22	ProBiS	-0.236

**Table S10:** AUC values and enrichment factors for the ProSPECCTs data sets for both SiteMine settings (*Fast* and *Precise*, SP-score) comparing the performance of ligand-radius and DoGSite3-defined binding sites.

Data set	Method	AUC	EF						
			1.6%	8.1%	16.1%	32.2%	48.4%	64.5%	80.6%
Barelrier <sup>2</sup>	<i>Fast</i> (DoGSite)	0.62	0.00	0.64	1.28	1.44	1.39	1.20	1.09
	<i>Precise</i> (Radius)	0.61	3.26	1.63	1.31	1.14	1.26	1.14	1.11
	<i>Precise</i> (DoGSite)	0.60	3.21	1.28	1.61	1.44	1.18	1.04	1.03
	<i>Fast</i> (Radius)	0.56	0.00	1.09	0.98	0.98	1.16	1.22	1.11
Data set	Method	AUC	0.1%	0.5%	1%	2%	3%	4%	5%
Decoy Structures Rational	<i>Precise</i> (DoGSite)	0.69	2.00	2.00	2.00	1.94	1.87	1.84	1.82
	<i>Fast</i> (DoGSite)	0.69	2.00	2.00	2.00	1.93	1.87	1.84	1.82
	<i>Precise</i> (Radius)	0.69	2.00	2.00	2.00	1.78	1.64	1.61	1.59
	<i>Fast</i> (Radius)	0.65	2.00	2.00	2.00	1.81	1.66	1.61	1.56
Decoy Structures Shape	<i>Precise</i> (DoGSite)	0.76	2.00	2.00	2.00	1.99	1.98	1.98	1.98
	<i>Fast</i> (DoGSite)	0.76	2.00	2.00	2.00	1.99	1.98	1.98	1.97
	<i>Precise</i> (Radius)	0.72	2.00	2.00	2.00	1.96	1.89	1.87	1.87
	<i>Fast</i> (Radius)	0.71	2.00	2.00	2.00	1.96	1.91	1.90	1.90
Structures with Identical Sequences	<i>Precise</i> (DoGSite)	1.00	7.91	7.91	7.91	7.91	7.91	7.91	7.91
	<i>Fast</i> (DoGSite)	1.00	7.91	7.91	7.91	7.91	7.91	7.91	7.91
	<i>Precise</i> (Radius)	1.00	7.91	7.91	7.91	7.91	7.91	7.91	7.91
	<i>Fast</i> (Radius)	1.00	7.91	7.91	7.91	7.91	7.91	7.91	7.91
Kahraman <sup>3,4</sup>	<i>Precise</i> (Radius)	0.78	7.58	7.58	7.50	6.22	5.49	4.99	4.70
	<i>Fast</i> (Radius)	0.74	7.58	7.58	7.50	6.03	5.46	4.95	4.57
	<i>Precise</i> (DoGSite)	0.70	7.65	7.65	7.65	6.25	5.19	4.54	4.13
	<i>Fast</i> (DoGSite)	0.66	7.65	7.65	7.65	5.90	4.82	4.17	3.83
NMR Structures	<i>Precise</i> (DoGSite)	1.00	14.00	14.00	14.00	14.00	14.00	14.00	14.00
	<i>Precise</i> (Radius)	1.00	14.00	14.00	14.00	14.00	14.00	14.00	14.00
	<i>Fast</i> (Radius)	0.98	14.00	14.00	14.00	14.00	14.00	14.00	14.00
	<i>Fast</i> (DoGSite)	0.94	14.00	14.00	14.00	14.00	14.00	14.00	14.00
Successful Applications	<i>Precise</i> (Radius)	0.91	421.59	132.17	69.56	36.09	24.93	18.91	15.13
	<i>Fast</i> (Radius)	0.89	421.59	128.69	67.82	33.91	23.19	17.39	14.26
	<i>Precise</i> (DoGSite)	0.88	437.66	131.83	70.37	35.19	23.76	18.25	14.61
	<i>Fast</i> (DoGSite)	0.88	437.66	133.56	71.24	36.49	24.34	18.25	14.61
ROCS Structures	<i>Precise</i> (Radius)	0.97	4.67	4.67	4.67	4.66	4.65	4.64	4.62
	<i>Fast</i> (Radius)	0.92	4.67	4.67	4.67	4.67	4.65	4.63	4.58
	<i>Precise</i> (DoGSite)	0.82	4.62	4.62	4.60	4.43	4.28	4.12	3.96
	<i>Fast</i> (DoGSite)	0.78	4.62	4.62	4.60	4.38	4.15	3.98	3.80

**Table S11:** The Spearman’s Rho correlation coefficients for the *Decoy Structures Rational* data set for comparing the crystal structure to the mutated ones (one to five randomly chosen mutations).

PDB-ID.chain	Ligand radius-defined		DoGSite3-defined	
	SiteMine <i>Fast</i>	SiteMine <i>Precise</i>	SiteMine <i>Fast</i>	SiteMine <i>Precise</i>
1kmv.A	-0.88	-0.89	-0.77	-0.79
1odm.A	-0.73	-0.74	-0.69	-0.70
2qwx.A	-0.69	-0.70	-0.59	-0.60
3f17.A	-0.67	-0.67	-0.62	-0.64
3rm2.H	-0.68	-0.86	-0.76	-0.75
3t10.A	-0.86	-0.83	-0.72	-0.71
3u5l.A	-0.67	-0.67	-0.64	-0.63
3u9w.A	-0.70	-0.69	-0.75	-0.75
4bfz.A	-0.63	-0.63	-0.68	-0.69
4buu.A	-0.73	-0.70	-0.71	-0.75
4ca7.A	-0.73	-0.70	-0.69	-0.69
4fpt.A	-0.79	-0.84	-0.84	-0.85
Mean	-0.73	-0.74	-0.71	-0.71

**Table S12:** The Spearman’s Rho correlation coefficients for the *Decoy Structures Shape* data set for comparing the crystal structure to the mutated ones (one to five randomly chosen mutations).

PDB-ID.chain	Ligand radius-defined		DoGSite3-defined	
	SiteMine <i>Fast</i>	SiteMine <i>Precise</i>	SiteMine <i>Fast</i>	SiteMine <i>Precise</i>
1kmv.A	-0.97	-0.97	-0.85	-0.88
1odm.A	-0.93	-0.89	-0.93	-0.90
2qwx.A	-0.86	-0.88	-0.73	-0.73
3f17.A	-0.86	-0.82	-0.76	-0.72
3rm2.H	-0.64	-0.74	-0.71	-0.71
3t10.A	-0.91	-0.90	-0.73	-0.72
3u5l.A	-0.86	-0.87	-0.91	-0.91
3u9w.A	-0.84	-0.84	-0.84	-0.81
4bfz.A	-0.94	-0.95	-0.88	-0.87
4buu.A	-0.92	-0.93	-0.76	-0.75
4ca7.A	-0.91	-0.91	-0.78	-0.78
4fpt.A	-0.95	-0.96	-0.93	-0.92
Mean	-0.88	-0.89	-0.82	-0.81

**Table S14:** The sequence-culled PISCES<sup>22</sup> parameters.

Parameter	Value
Maximum pairwise percent sequence identity	100%
Minimum resolution	0.0
Maximum resolution	2.0
Maximum R-value	0.25
Minimum chain length	40
Maximum chain length	10000
Include X-ray entries	yes
Include cryo-EM entries	yes
Include NMR entries	yes
Include chains with chain breaks	yes
Include chains with missing residues due to disorder?	yes
Resulting number of chains	42797

**Table S15:** The top 30 scored (normalized by the larger site) binding sites for the search for the bound inhibitor (ligand three-letter-code: 424) radius-defined active site of cathepsin L (2xu1). The ligand ID consists of the residue name - chain ID - residue sequence number, as annotated in the PDB file.

Rank	PDB ID	SP-Score	Pharmacophore	Shape	Ligand ID
1	2XU3	1.86	0.90	0.95	XU3-A-1221
2	7QGW	1.78	0.86	0.92	RN2-A-309
3	7QGW	1.77	0.85	0.92	RN2-B-510
4	3I06	1.62	0.79	0.84	QL2-A-220
5	3IUT	1.50	0.71	0.79	KB2-A-300
6	6YCG	1.46	0.67	0.80	TCK-B-304
7	6P4E	1.46	0.70	0.76	GES-B-307
8	1M6D	1.45	0.71	0.74	MYP-B-2280
9	3QNS	1.43	0.64	0.79	GOL-A-355
10	5A24	1.43	0.67	0.76	E64-A-1224
11	2F7D	1.42	0.68	0.75	NOQ-A-600
12	4X6H	1.41	0.64	0.77	3XT-A-301
13	2OKX	1.39	0.63	0.76	GOL-A-3015
14	1VSN	1.39	0.66	0.73	NFT-A-283
15	2P86	1.39	0.67	0.72	VS1-A-300
16	3M1T	1.36	0.64	0.71	GOL-A-275
17	7NX1	1.32	0.57	0.75	BTB-A-405
18	4BS6	1.30	0.60	0.69	JG7-B-1341
19	2OKX	1.29	0.61	0.69	GOL-B-3016
20	1M6D	1.29	0.62	0.67	MYP-A-1280
21	3IOQ	1.28	0.60	0.68	E64-A-301
22	4X6H	1.27	0.56	0.71	I37-A-302
23	7EOD	1.27	0.47	0.80	GOL-B-301
24	6NPS	1.27	0.60	0.67	GOL-A-1004
25	6PTZ	1.20	0.42	0.78	GOL-A-510
26	7QBM	1.19	0.56	0.63	A0U-A-301
27	5OKA	1.18	0.56	0.62	GOL-B-502
28	5OKA	1.18	0.56	0.62	GOL-A-501
29	3ZQK	1.16	0.51	0.65	NAG-C-4677
30	5E8E	1.16	0.48	0.68	PCA-B-1


**Table S16:** The top 30 scored (non-normalized, raw score) binding sites for the search for the bound inhibitor (ligand three-letter-code: 424) radius-defined active site of cathepsin L (2xu1). The ligand ID consists of the residue name - chain ID - residue sequence number, as annotated in the PDB file.

Rank	PDB ID	SP-Score	Pharmacophore	Shape	Ligand ID
1	2XU3	196.6	95.6	101.0	XU3-A-1221
2	7QGW	158.4	76.4	82.0	RN2-A-309
3	2P86	158.2	76.2	82.0	VS1-A-300
4	7QGW	157.4	75.4	82.0	RN2-B-510
5	1VSN	146.0	69.0	77.0	NFT-A-283
6	2F7D	139.4	66.4	73.0	NOQ-A-600
7	3IUT	138.4	65.4	73.0	KB2-A-300
8	4X6H	137.0	62.0	75.0	3XT-A-301
9	6P4E	130.0	62.0	68.0	GES-B-307
10	1M6D	128.8	62.8	66.0	MYP-B-2280
11	3I06	128.0	62.0	66.0	QL2-A-220
12	1M6D	125.0	60.0	65.0	MYP-A-1280
13	3IOQ	120.4	56.4	64.0	E64-A-301
14	1AEC	118.6	57.6	61.0	E64-A-219
15	4BS6	118.0	55.0	63.0	JG7-B-1341
16	5A24	117.0	55.0	62.0	E64-A-1224
17	7QBM	115.6	54.6	61.0	A0U-A-301
18	6YCG	106.8	48.8	58.0	TCK-B-304
19	4X6H	93.0	41.0	52.0	I37-A-302
20	1THE	91.6	41.6	50.0	0E6-A-901
21	6YI7	91.0	42.0	49.0	ORW-A-403
22	2DCC	88.0	41.0	47.0	77B-A-770
23	6AY2	86.8	40.8	46.0	C1G-A-301
24	1THE	77.0	35.0	42.0	0E6-B-911
25	2R9F	74.8	35.8	39.0	K2Z-A-367
26	6AY2	73.8	33.8	40.0	C1G-B-301
27	1ZCM	72.8	31.8	41.0	C1N-A-1115
28	3U2M	67.8	24.8	43.0	FAD-A-301
29	1J8Q	67.0	23.0	44.0	FMN-A-149
30	7TJA	64.4	20.4	44.0	DBV-K-101

# Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, den 6. März 2024

A handwritten signature in black ink, appearing to read 'Joel Graef', is written over a horizontal line.

Joel Graef