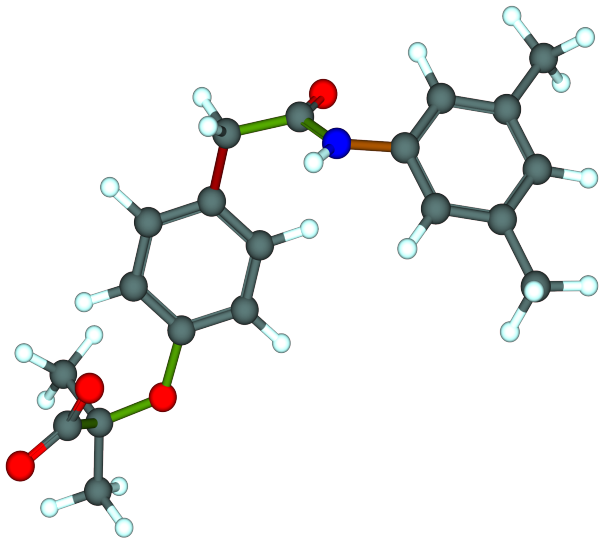
Wissensbasierte Analyse von Konformationen in kleinen Molekülen



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

Christin Schärfer
aus Hamburg

Hamburg, April 2013

Korrigierte Fassung

Gutachter:

Prof. Dr. Matthias Rarey
JProf. Dr. Tobias Schwabe
Prof. Dr. Holger Gohlke

Tag der Disputation: 15.10.2013

Kurzfassung

Viele der Methoden im computergestützten Wirkstoffdesign, wie zum Beispiel *Docking*, *Shape Matching* oder *Pharmacophore Searching*, benutzen Konformationen, um die Flexibilität von Molekülen zu beschreiben. Das zugrunde liegende Konformationsmodell hat dabei einen wesentlichen Einfluss auf die Ergebnisse der Anwendungen, weshalb der Analyse und Generierung von Konformationen eine besondere Bedeutung zukommt. Die Konformationsräume kleiner Moleküle können mit Hilfe von Expertenwissen beschrieben werden. Die meisten Anwendungen im computergestützten Wirkstoffdesign arbeiten mit Datenbanken, in denen Millionen von Molekülen gespeichert sind, was eine manuelle Beschreibung der Konformationsräume unmöglich macht. Es existieren bereits mehrere Methoden zur Konformationsgenerierung, von denen viele jedoch noch immer nicht zu optimalen Ergebnissen führen. Das anhaltende Interesse an diesem Thema in der Literatur zeigt, dass hier ein Bedarf an weiteren Verbesserungen besteht.

In dieser Arbeit wird ein neues wissensbasiertes Konformationsmodell vorgestellt, welches sowohl zur Analyse als auch zur Generierung von Konformationen eingesetzt werden kann. Das Modell basiert auf einer Sammlung von Torsionsregeln. Jede dieser Regeln beschreibt eine rotierbare Bindung und ihre bevorzugten Torsionswinkel, welche aus experimentellen Daten abgeleitet wurden. Dabei decken die spezifischen Torsionsregeln bereits etwa 96% des für die Medizinalchemie relevanten Konformationsraums ab.

Das Modell kann mit Hilfe des *TorsionAnalyzers*, einem graphischen Softwarewerkzeug zur Analyse von Molekülkonformationen, angezeigt und bearbeitet werden. Anschließend wird das neue Modell in *CONFECT*, einer neuen Methode zur Generierung von Molekülkonformationen benutzt, um Konformationsensembles zu erzeugen. Beim Vergleich mit anderen Methoden liefert CONFECT vergleichbare Ergebnisse bei der Reproduktion der bioaktiven Konformation, benötigt dafür aber weniger Zeit und eine kleinere Ensemblegröße.

Zum Vergleich von Konformationen, wird häufig der relative RMSD verwendet. Die Vorteile des RMSD sind seine universelle Einsetzbarkeit, seine Objektivität und seine einfache und automatisierbare Berechnung. Allerdings hat der RMSD-Vergleich gravierende Nachteile. Der *TFD*, ein neues Maß zum Vergleich von Konformationen überwindet diese Nachteile, behält dabei aber die Vorteile des RMSD.

Abstract

Many applications in computer aided drug design, e.g. docking, pharmacophore searching and 3D-QSAR, use conformations to adequately represent the conformational flexibility of a molecule. The underlying conformational model has a major impact on the results of these applications, which makes the generation of conformations a central task in computer aided drug design. The conformational space for a single molecule can easily be described manually by using the expert knowledge of a computational chemist. For modelling applications involving millions of compounds, however a manual approach is not feasible. The problem of generating low-energy conformations is not new and there are several methods and tools described in the literature. However, there are still several issues in conformation generation that remain not optimally solved.

A new knowledge-based conformation model is presented in this thesis. The model can either be used to analyse or to generate small molecule conformations. It is based on a set of torsion rules, each describing a rotatable bond and its preferred torsion angles, derived from experimental data. The set of specific torsion rules already covers about 96% of conformational space relevant for medicinal chemistry.

The new conformation model can be explored and modified with the *TorsionAnalyzer*, an interactive graphical software tool which can also be used to analyse small molecule conformations. *CONFECT*, an new method to generate small molecule conformations uses the model to generate ensembles of relevant conformations. In comparison with other methods, CONFECT performs equally well in reproducing the bioactive conformation, but requires less time and smaller ensembles.

Objectivity, intuitive interpretation, general applicability, and its easy, automated calculation make the relative RMSD the measure of choice for comparing small molecule conformations. However, there are some significant weaknesses in RMSD comparisons. The *TFD*, a novel measure to compare conformations of small molecules, overcomes major limitations of RMSD while retaining its advantages.

Danksagung

An dieser Stelle gilt mein herzlicher Dank allen, die mich während meiner Promotionszeit unterstützt und zum guten Gelingen dieser Arbeit beigetragen haben.

Prof. Dr. Matthias Rarey danke ich für die Möglichkeit, mich mit einem sehr spannenden Thema in meiner Dissertation zu beschäftigen und meine Arbeit auf nationalen und internationalen Konferenzen zu präsentieren. Des Weiteren möchte ich mich bei ihm für die intensive fachliche Betreuung, die sehr gute Unterstützung und vor allem für seinen motivierenden Infovortrag, der mich überhaupt erst zum Studium der Bioinformatik gebracht hat, bedanken.

Ich bedanke mich auch bei JProf. Dr. Tobias Schwabe und Prof. Dr. Holger Gohlke für die Begutachtung meiner Dissertationsschrift.

Ich möchte mich bei der Firma F. Hoffmann-La Roche Ltd. für die Finanzierung des Projektes, das dieser Dissertation zugrunde liegt, bedanken. Mein besonderer Dank gilt Tanja Schulz-Gasch für die tolle Zusammenarbeit, das Korrekturlesen dieser Dissertation, ihre große Hilfsbereitschaft und für die vielen netten Abende im Fünfschilling und der Sushi-Bar. Ich danke außerdem Martin Stahl, Wolfgang Guba und den weiteren Mitarbeitern der Abteilung Discovery Chemistry für interessante Diskussionen, das Testen meiner Software und die gute Zusammenarbeit.

Bei den Mitarbeitern der Firma BioSolveIT, besonders bei Christian Lemmen, möchte ich mich für die Hilfe bei Problemen mit der Softwarebibliothek, für spannende Diskussionen und für die Bereitstellung der TorsionAnalyzer-Webseite bedanken.

Mein herzlicher Dank gilt allen Mitgliedern und ehemaligen Mitgliedern der Arbeitsgruppe Algorithmisches Molekulares Design für ein tolles Arbeitsklima, anregende Gespräche und nette Doppelkopfrunden. Ich danke Hans-Christian Ehrlich für die Bereitstellung des SMARTS-Matchers, Lennart Heinzerling für die Bereitstellung des Optimierers, Tobias Lippert, Robert Fischer, Adrian Kolodzik und Sascha Urbaczek für die Bereitstellung der NAOMI-Bibliothek, Therese Inhester für die Bereitstellung von coord3d, Angela Henzler für das intensive Testen von CONFECT und Birte Seebeck, die netteste Bürokollegin die man sich nur wünschen kann.

Ich möchte mich auch bei allen anderen Mitarbeitern des ZBH für ein angenehmes Arbeitsumfeld bedanken. Vor allem möchte ich mich bei Dirk Willrodt für leckeren

Kuchen und nette Mittagspausen und bei Jörn Adomeit und Christian Rhein für die Unterstützung bei technischen Problemen bedanken.

Mein besonderer Dank gilt Sascha Steinbiß für das Korrekturlesen dieser Arbeit, seine unendliche Geduld und dafür, dass er mich gerade in letzter Zeit so lecker bekocht hat. Danke, dass du immer für mich da bist!

Der größte Dank gebührt meinen Eltern. Danke, dass ihr mich in allem unterstützt habt und immer für mich da seid. Ohne euch wäre ich niemals so weit gekommen!

Inhaltsverzeichnis

Kapitel 1

1 Einleitung

1.1. Motivation

Der Mensch hat sich über Jahrtausende Wissen angeeignet, wie er bestimmte Krankheitsbilder, zum Beispiel Wunden, Schmerzen und Entzündungen, mit Pflanzen oder Pflanzenextrakten behandeln kann. Heute ist das Wissen über die Vorgänge im menschlichen Körper detaillierter. So ist zum Beispiel bekannt, welche Proteine an der Blutgerinnung und Wundschließung beteiligt sind, und wie diese mit Hilfe von Wirkstoffen aktiviert oder gehemmt werden können. Die Entwicklung neuer Medikamente ist jedoch nicht einfacher geworden, da sich mit der Zeit auch die Krankheitsbilder verändert haben. Es geht heute nicht mehr nur darum, Blutungen zu stillen und Wunden zu schließen, sondern auch darum, komplexe psychische (Depressionen, Burnout, Autismus), chronische (Krebs, Diabetes, Multiple Sklerose) oder Infektionskrankheiten (HIV, Hepatitis, Tuberkulose, Malaria, Influenza) zu behandeln. Resistenzen von Viren und Bakterien gegenüber bereits vorhandenen Wirkstoffen stellen eine enorme Herausforderung dar. Als Gegenmaßnahme müssen neue Medikamente entwickelt werden, da die bisherigen zunehmend ihre Wirkung verlieren.

Heute werden bei der Entwicklung neuer Medikamente in einem ersten Schritt die biologischen Prozesse im Körper, die zu der Krankheit führen, analysiert. Diese Prozesse sind in der Regel nicht isoliert zu betrachten, sondern vielmehr sind sie in ein Netzwerk (*pathway*) von Einzelreaktionen eingebunden. Idealerweise lassen sich hierbei eine oder mehrere kritische

Reaktionen innerhalb des Netzwerkes identifizieren, welche dann als Angriffspunkt (*Target*) für einen neuen Wirkstoff genutzt werden können. Dieser Angriffspunkt ist zumeist ein Protein, dessen Funktion mit Hilfe des Wirkstoffes entweder inhibiert, aktiviert oder moduliert werden soll. Damit der Wirkstoff die gewünschte biologische Wirkung erzielt, muss er spezifisch an das Target binden. Nach dem vereinfachten *Schlüssel-Schloss-Prinzip* bedeutet dies, dass der Wirkstoff (*Schlüssel*) die richtige Größe und Form haben muss, um optimal in die Bindetasche des Proteins (*Schloss*) zu passen. Für die biologische Aktivität zwischen Protein und Wirkstoff ist es zudem wichtig, dass die physiko-chemischen Eigenschaften komplementär sind, damit spezifische Wechselwirkungen ausgebildet werden können.

Ein häufig genutzter Ansatz bei der Suche nach Startstrukturen für die Entwicklung neuer Medikamente ist die Suche (*high-throughput screening*) in sogenannten *Substanz-Bibliotheken*. Hierzu wird für jedes Molekül aus der Bibliothek ($> 10^6$ Moleküle) überprüft, ob es mit dem Target wechselwirkt. Zeigt eines der Moleküle eine biologische Aktivität, wird es näher charakterisiert und eventuell weiterentwickelt. Typische Substanz-Datenbanken decken nur einen Bruchteil der Gesamtheit der wirkstoffähnlichen Moleküle ($\approx 10^{60}$ Moleküle) ab, weshalb eine erschöpfende Suche auf konventionelle Weise nicht durchführbar ist. Zudem stehen nicht alle möglichen wirkstoffähnlichen Moleküle synthetisiert zur Verfügung, um im Screening getestet zu werden. Einen effizienteren Ansatz zur Suche nach Startstrukturen bietet das *computergestützte Wirkstoffdesign*. Bei sogenannten *virtual Screenings* [1, 2] wird in Molekül-Datenbanken nach *Hits* oder *Leitstrukturen* gesucht, die als Ausgangspunkte für die Entwicklung neuer Wirkstoffe benutzt werden können. Leitstrukturen besitzen die biologische Wirkung auf dem Target und ein Profil, dass eine Weiterentwicklung erfolgreich erscheinen lässt [3]. Eine Teildisziplin der medizinischen Arzneimittelforschung, die in den letzten Jahrzehnten an Bedeutung gewonnen hat, ist das rationale *Wirkstoffdesign*. Hierbei werden die zu testenden Leitstrukturen nicht zufällig einer Substanz-Datenbank entnommen, sondern gezielt mit Hinblick auf ihren spezifischen Bindungspartner entwickelt.

Viele der Anwendungen im computergestützten Wirkstoffdesign, wie zum Beispiel *Docking* [4–6], *Shape Matching* [7] oder *Pharmacophore Searching* [8] sind auf die dreidimensionale Struktur von Molekülen angewiesen, da diese die chemischen, biologischen und physikalischen Eigenschaften eines Moleküls bestimmen. Die Struktur eines Moleküls im Raum ist dabei nicht starr, sondern flexibel. Unterschiedliche räumliche Anordnungen der Molekülatome werden *Konformationen* genannt [9]. Das zugrunde liegende Konformationsmodell hat einen wesentlichen Einfluss auf die Ergebnisse

im Wirkstoffdesign, weshalb der *Konformationsanalyse* (die Untersuchung von Konformationen [10]) eine besondere Bedeutung zukommt. So hängen Verfahren wie zum Beispiel Docking oder Shape Matching stark von der Qualität der Konformationen ab. Es existieren bereits Programme [11–27] zur Konformationsgenerierung, die entweder eine stochastische (zufällige Suche, Simulationsverfahren) oder deterministische Methode (systematische Suche, wissensbasierte Ansätze) benutzen. Für das Durchsuchen großer Substanz-Datenbanken haben sich die deterministischen Methoden, vor allem die wissensbasierten Ansätze, als effektiv erwiesen [9, 10, 28].

Die Generierung von relevanten Konformationen für das computergestützte Wirkstoffdesign ist ein unterschätztes Problem. Viele der existierenden Methoden führen noch immer nicht zu optimalen Ergebnissen. Die anhaltende Präsenz dieses Themas in der Literatur zeigt, dass Konformationsgenerierung nach wie vor ein wichtiger Forschungsaspekt im Wirkstoffdesign ist und ein Bedarf an weiteren Verbesserungen besteht [8, 29].

Im Rahmen dieser Dissertation wurde ein neues wissensbasiertes Konformationsmodell entwickelt. Bei der Entwicklung wurde darauf geachtet, dass das Modell für den Benutzter transparent und erweiterbar ist. Das Modell kann sowohl zur Analyse als auch zum Generieren von Konformationen eingesetzt werden. Im Kontext dieser Arbeit wurde zudem ein neues Maß zum Vergleich von Konformationen entwickelt.

1.2. Projektbeschreibung

Das dieser Dissertation zugrunde liegende Projekt ist ein Kooperationsprojekt zwischen der Universität Hamburg und der Firma F. Hoffmann-La Roche Ltd. mit dem Ziel eine neues Programm zur Generierung von Konformationen zu entwickeln. Im ersten Schritt wurden die bevorzugten Konformationen von Molekülen anhand von zwei in der Medizinalchemie relevanten Datenbanken analysiert. Das Resultat dieser Analyse ist ein wissensbasierter, auf experimentellen Daten gestützter Regelsatz, der im Anschluss zur Parametrisierung eines neuen Konformationsmodells verwendet wurde. Das Modell sollte dabei transparent für den Benutzer sein, damit für die Konformationserzeugung die Entscheidung des Algorithmus für bzw. gegen bestimmte Konformationen nachvollziehbar bleibt. Ein weiterer Fokus lag auf der Erweiterbarkeit des Modells.

Die Ergebnisse dieser Arbeit entstanden in enger Zusammenarbeit mit der Abteilung Discovery Chemistry, insbesondere mit Tanja Schulz-Gasch. Die

Evaluierung der neu entwickelten Methoden wurde auf Grund der Auslegung des Projektes und der Erfahrungen von Tanja Schulz-Gasch auf dem Gebiet der Medizinalchemie und dem molekularen Wirkstoffentwurf gemeinsam von der Autorin dieser Arbeit und Tanja Schulz-Gasch durchgeführt.

1.3. Aufbau der Dissertation

Die vorliegende Arbeit wurde in der Arbeitsgruppe Algorithmisches Molekulares Design am Zentrum für Bioinformatik der Universität Hamburg in der Zeit von Oktober 2008 bis März 2013 durchgeführt. Die Arbeit wurde bereits in drei Postern und einem Vortrag auf wissenschaftlichen Konferenzen veröffentlicht. Des weiteren entstanden im Laufe des Projektes zwei Publikationen [30, 31], eine weitere ist in Vorbereitung [32]. In diesen Publikation wurden die Methoden hauptsächlich von der Autorin dieser Arbeit und die Evaluierung und Auswertung gemeinschaftlich von der Autorin dieser Arbeit und Tanja Schulz-Gasch beschrieben. Die restlichen Teile der Publikationen wurden von den jeweiligen Autoren gemeinsam bearbeitet. Die Beschreibungen der Methoden wurde teilweise aus den Publikationen in diese Arbeit übernommen und dann erweitert, bzw. näher erläutert. Es wurde außerdem ein Teil der Evaluierung und Auswertung aus den Publikationen in diese Arbeit übernommen.

In Kapitel 2 werden Grundbegriffe der Konformationsanalyse eingeführt und deren Ziel näher erläutert. Zudem werden zwei Datenbanken zur Analyse von Konformationen vorgestellt.

Kapitel 3 gibt einen grundlegenden Überblick über die bestehenden Ansätze und Verfahren zum Vergleich, zur Analyse und zur Generierung von Konformationen sowie über vorhandene Software.

In Kapitel 4 werden die im Rahmen dieser Arbeit entwickelten Methoden zur Analyse und Generierung von Molekülkonformationen näher beschrieben. Das Kapitel gliedert sich dabei in vier Teile. Im ersten Teil wird der *TFD*, eine neue Methode zum Vergleich von Konformationen beschrieben. Im zweiten Teil wird das Konzept der *Torsionsbibliothek*, welche sowohl für die Analyse als auch für die Generierung von Konformationen verwendet wird, erläutert. Der dritte Teil beschäftigt sich mit *CONFECT*, einer neuen wissensbasierten Methode zur Generierung von Konformationen. Der letzte Teil des Kapitels behandelt den *TorsionAnalyzer*, ein graphisches Softwarewerkzeug zur Analyse von Molekülkonformationen.

Kapitel 5 beschreibt die Datensätze und Methoden zur Evaluierung der in Kapitel 4 erwähnten Methoden. Die Evaluierung des TFD, der Torsionsbibliothek und der Konformationsgenerierung erfolgt dabei jeweils getrennt voneinander. Die Ergebnisse der einzelnen Evaluierungen werden dann in Kapitel 6 vorgestellt und diskutiert.

Kapitel 7 enthält eine Zusammenfassung der Ergebnisse der gesamten Arbeit sowie einen Ausblick auf Verbesserungsmöglichkeiten der einzelnen Methoden.

Im Anhang sind weitere Details über die Verwendung der entstandenen Software (A) und deren Implementierung (B), das XML-Schema der Torsionsbibliothek (C) sowie eine Liste der Publikationen, Vorträge und Poster, die aus dieser Arbeit hervorgegangen sind (D), enthalten.

Kapitel 2

2 Analyse von Konformationen

Dieses Kapitel führt die Grundbegriffe der Konformationsanalyse und zwei Datenbanken mit experimentell bestimmten 3D-Strukturen ein.

2.1. Grundbegriffe

Die *Konformationen* eines Moleküls werden üblicherweise beschrieben als „die Anordnungen der Atome des Moleküls im Raum die nur durch Rotation um Einfachbindungen ineinander umgewandelt werden können“ [10]. Die Gesamtheit aller möglichen Konformationen eines Moleküls wird häufig auch als *Konformationsraum* bezeichnet.

Eine Rotation um eine Einfachbindung wird durch den von vier aufeinander folgenden kovalent gebundenen Atomen definierten *Torsionswinkel* ϕ beschrieben (siehe Abbildung 2.1). ϕ ist dabei der Winkel, den sich Atom (4) gegen den Uhrzeigersinn um die Achse drehen muss, welche durch Atom (2) und (3) gebildet wird, um in einer Ebene mit den Atomen (1), (2) und (3) zu liegen. Die Betrachtungsreihenfolge der Atome, 1-4 oder 4-1, hat dabei keinen Einfluss auf die Berechnung von ϕ.

Konformationen die sich in ihren Torsionswinkeln unterscheiden, weisen typischerweise auch unterschiedliche Energien auf. Ein bekanntes Beispiel dafür sind die Energieunterschiede bei den gestaffelten und ekliptischen Konformationen von Butan (siehe Abbildung 2.2). Die unterschiedlichen

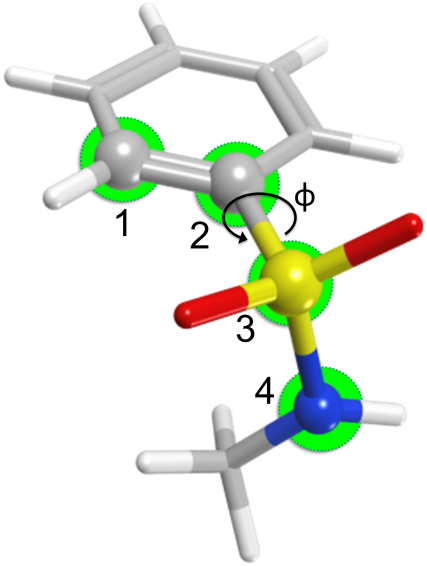


Abbildung 2.1.: Der Torsionswinkel ϕ wird durch 4 aufeinander folgende kovalent gebundene Atome definiert.

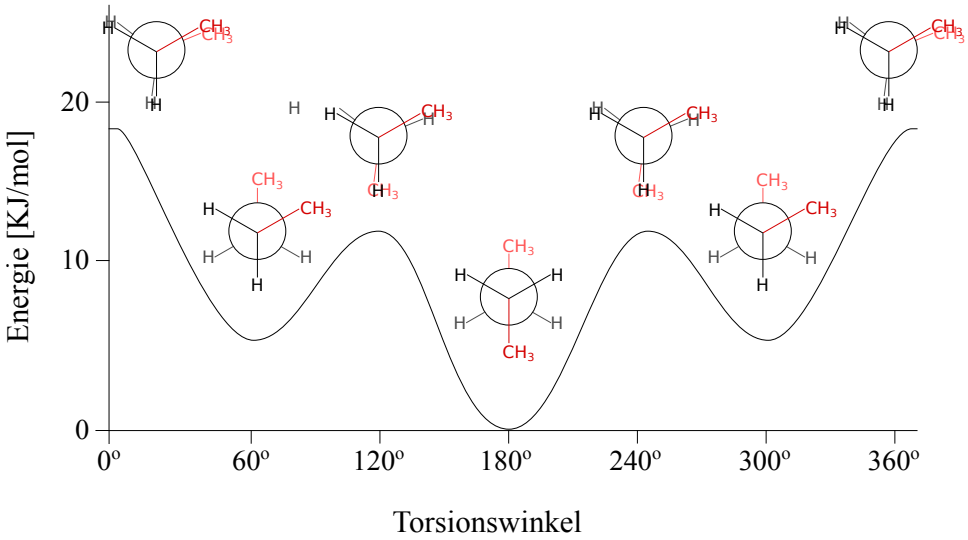


Abbildung 2.2.: Energie der unterschiedlichen Butan-Konformationen

Energien aller möglichen Konformationen eines Moleküls bilden eine (multidimensionale) *Energielandschaft* mit einem globalen und in den meisten Fällen mehreren lokalen Minima (siehe Abbildung 2.3) [9].

Welche Konformation bevorzugt wird, hängt von den Interaktionen des Moleküls mit seiner Umgebung ab. Die beobachteten Konformationen ein und

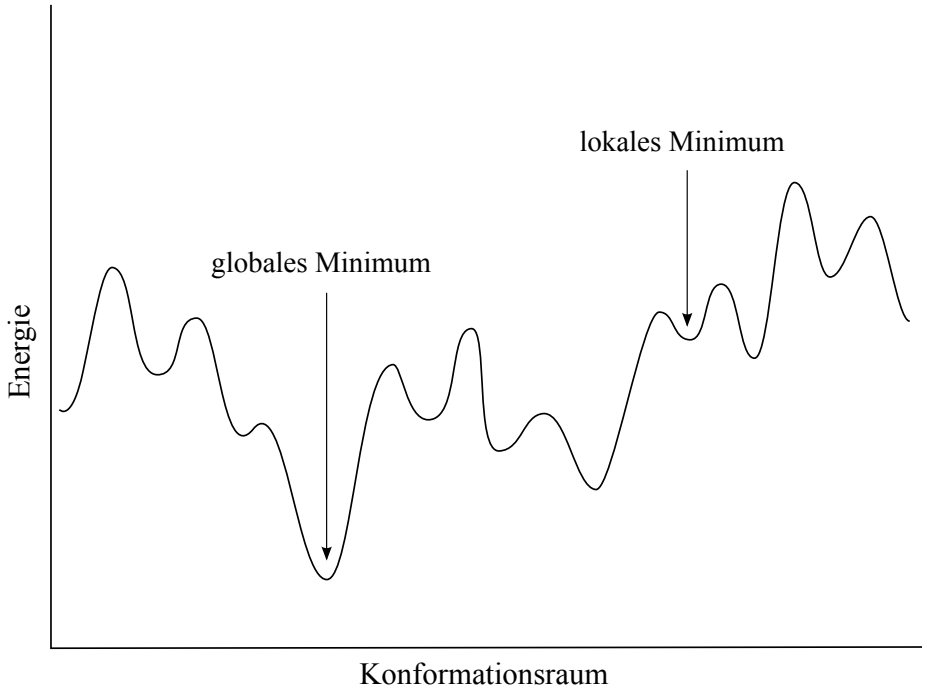


Abbildung 2.3.: Energielandschaft aller möglichen Konformationen eines Moleküls

des selben Moleküls können sich stark unterscheiden, je nach dem ob sich das Molekül in der Gasphase, im Lösungsmittel, in einer Kristallpackung oder in der Bindetasche des Proteins befindet [9]. Die für das Wirkstoffdesign relevante Konformation ist die sogenannte *bioaktive* Konformation.

2.2. Ziel der Konformationsanalyse

Das Ziel der Konformationsanalyse ist die Erzeugung eines Ensembles von Konformationen, die geeignete Kandidaten für eine bioaktive Form darstellen [9,33]. Die naheliegende, naive Vorstellung, dass eine Konformation mit minimaler Energie in der Praxis ausreicht, ist nicht haltbar, da laut verschiedener Studien bioaktive Konformationen nicht zwingend solche mit global minimaler Energie sind. Vielmehr entsprechen sie in vielen Fällen nicht einmal einem der lokalen Energieminima [34–39]. Des Weiteren hat ein Teil der Studien gezeigt, dass einige Moleküle wie zum Beispiel AMP/ADP/ATP oder Methotrexat an verschiedene Proteine binden und dabei unterschiedliche bioaktive Konformationen einnehmen [38,39]. Zur Auswahl der Konformationen für das Ensemble gibt es zwei Strategien: entweder wird eine

Menge möglichst diverser d.h. untereinander möglichst unähnlicher Konformationen (siehe Abschnitt 3.1), oder eine Menge von Konformationen mit besonders niedriger Energie gesucht [40]. Bei einem diversen Ensemble wird versucht, den Konformationsraum durch das gewählte Ensemble möglichst gut zu repräsentieren. Der Vorteil eines diversen Ensembles ist, dass die Wahrscheinlichkeit sehr hoch ist, Konformationen im Ensemble zu haben, die sehr ähnlich zu den bioaktiven Konformationen sind. Der Nachteil eines diversen Ensembles ist allerdings, dass häufig auch viele Konformationen dabei sind, die energetisch ungünstig sind. Zur Auswahl der Konformationen mit niedriger Energie wird meist eine Energie-Obergrenze θ festgesetzt. Alle Konformationen mit Energie $< \theta$ werden dann in das Ensemble aufgenommen. Der Nachteil bei diesem Ansatz zur Ensembleerzeugung besteht darin, dass durch die Beschränkung auf energiearme Konformationen eventuell bioaktive Konformationen außerhalb der Energieminima nicht berücksichtigt werden.

2.3. Datenbanken zur Konformationsanalyse

Zwei wichtige Datenbanken in der Medizinalchemie sind die vom Cambridge Crystallographic Data Centre (CCDC) angebotene Cambridge Structural Database (CSD) [41] und die vom Research Collaboratory for Structural Bioinformatics (RCSB) verwaltete Protein Data Bank (PDB) [42]. Diese beiden Datenbanken eignen sich ideal für die Analyse von bevorzugten Konformationen und Protein-Ligand-Interaktionen [31, 43–49] und werden im Folgenden kurz vorgestellt.

2.3.1. CSD

Die seit 1965 bestehende CSD beinhaltet aktuell über 600.000 Kristallstrukturen für kleine organische und metallorganische Verbindungen bei einem jährlichen Zuwachs von 40.000 neuen Strukturen [50]. Die 3D-Strukturen der Moleküle wurden anhand von Röntgenbeugung [51] oder Neutronenstreuung [52] aufgeklärt und von einem Expertenteam aus Chemikern und Kristallographen verifiziert. Jeder Eintrag enthält zusätzlich zur 3D-Struktur weitere Informationen zu den chemischen und physikalischen Eigenschaften des Moleküls. Die CSD ist kommerziell und nicht frei zugänglich.

2.3.2. PDB

Im Gegensatz zur CSD ist die PDB öffentlich. Sie enthält zur Zeit fast 90.000 3D-Strukturen von großen biologischen Molekülen (darunter ca. 65.000 Proteine mit gebundenen kleinen Molekülen). Die Strukturen stammen aus unterschiedlichen Organismen wie zum Beispiel Bakterien, Hefen, Pflanzen, Tieren und Menschen und wurden mit Röntgenbeugung, Kernspinresonanzspektroskopie [53] oder Elektronenmikroskopie aufgeklärt [54]. Die PDB startete 1971 am Brookhaven National Laboratory und wurde 1998 vom RCSB übernommen. Der jährliche Zuwachs an neuen Strukturen steigt stetig an. So sind zum Beispiel 1992 gerade mal etwa 200, 2002 schon über 3000 und 2012 fast 9000 neue Strukturen hinzugekommen [55].

Kapitel 3

3 Bestehende Ansätze und Verfahren

Dieses Kapitel beschreibt bestehende Ansätze und Verfahren zum Vergleich, zur Analyse und zur Generierung von Konformationen. Es enthält zudem einen Überblick über vorhandene Software.

3.1. Vergleich von Konformationen

Im folgenden Abschnitt werden bestehende Ansätze zum Vergleich von Konformationen näher beschrieben. Methoden zum Vergleich von Konformationen sind für verschiedene Aufgaben bei der Konformationsanalyse notwendig. Ein Anwendungsfall ist die Bewertung der Ergebnisse von Programmen zur Generierung von Konformationen, ein anderer die Entfernung von Duplikaten und die Erstellung möglichst diverser Konformationsensemble.

Bei der Evaluierung von Methoden zur Konformationsgenerierung wird häufig untersucht, in wie weit die Methode in der Lage ist, bioaktive Konformationen zu reproduzieren, wie viele Konformationen dabei generiert werden und wie lange die Methode dafür gebraucht hat. Um zu bewerten, ob eine bioaktive Konformation reproduziert wurde, wird berechnet, wie ähnlich die generierten Konformationen der bioaktiven Konformation sind.

Um Duplikate zu entfernen oder um ein möglichst diverses Konformationsensemble zu erzeugen, wird gemessen, wie ähnlich eine neu generierte Konformation den bereits generierten Konformationen des Ensembles ist [9, 40]. Bei der Erstellung eines diversen Ensembles wird häufig ein Parameter, wie

z.B. ein Grenzwert eines Abstands- oder Ähnlichkeitsmaßes, angegeben, der festlegt, wann eine Menge von Konformationen als äquivalent betrachtet wird. Mit Hilfe dieses Parameters lässt sich dann die Diversität des Ensembles kontrollieren.

3.1.1. RMSD

Der *RMSD (root mean square deviation)* wird in vielen verschiedenen Forschungsgebieten wie zum Beispiel Meteorologie, Wirtschaftswissenschaften, Chemie- oder Bioinformatik verwendet. In der Chemieinformatik wird der RMSD oft benutzt, um Konformationen miteinander zu vergleichen. Der RMSD zwischen zwei Konformationen berechnet sich dabei nach der folgenden Formel:

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N} d_i^2}{N}}$$

wobei N die Anzahl der Atome ist und d_i die Distanz zwischen den Koordinaten von Atom i in beiden Konformationen. Beim Vergleich von Konformationen muss zwischen dem *absoluten* und dem *relativen* RMSD unterschieden werden [56]. Beim absoluten RMSD wird die Distanzen zwischen den entsprechenden Atome gemessen, ohne die Koordinaten durch Translation oder Rotation der Moleküle zu verändern. Der relative RMSD benötigt eine zusätzliche Überlagerung der beiden Moleküle vor der eigentlichen RMSD-Berechnung. Die Moleküle werden dabei so überlagert, dass die Distanzen zwischen den entsprechenden Atomen minimal sind [57]. Zum Vergleich von Konformationen wird meist der relative RMSD verwendet.

Um zu beurteilen, wie ähnlich eine Konformation der bioaktiven Konformation ist, werden die Konformationen anhand der folgenden RMSD-Werte klassifiziert [27,58]:

- $RMSD < 1,0$Å: Die Konformation ist sehr ähnlich zur bioaktiven Konformation.
- $RMSD < 1,5$Å: Die Konformation ist der bioaktiven Konformation ähnlich, weicht aber in einigen Teilen leicht ab.
- $RMSD < 2,0$Å: Die Konformation ist der bioaktiven Konformation zwar immer noch recht ähnlich, weicht aber in einigen Teilen stark ab.
- $RMAD > 2,0$Å: Die Konformation ist der bioaktiven Konformation zu unähnlich.

Die Vorteile des RMSD sind seine universelle Einsetzbarkeit, seine Objektivität und seine einfache und automatisierbare Berechnung [59]. Der RMSD hat allerdings auch einige gravierende Nachteile. Zum einen lässt sich aus dem RMSD keinerlei Information über die Qualität der Bindung der Konformation an das Protein ableiten [56]. Es kann zum Beispiel sein, dass sich eine Konformation lokal zwar sehr gut mit der Referenzstruktur überlagern lässt, aber durch die Betrachtung der globalen Überlagerung einen schlechten RMSD hat. Die Konformation wird dann aufgrund des schlechten RMSD verworfen, obwohl vielleicht gerade nur der lokal gut überlagerte Teil wichtig für die Bindung an das Protein ist. Zum anderen hängt der RMSD stark von der Größe des Moleküls ab. Zum Beispiel haben kleine, kompakte Moleküle oft sehr kleine RMSD-Werte, sogar wenn ihre Atome willkürlich verteilt werden [56, 60]. Die Abhängigkeit von der Größe des Moleküls und die Tatsache, dass der RMSD nicht normalisiert ist, ist besonders problematisch, wenn RMSD-Werte über einen großen Datensatz gemittelt werden. Sehr große und flexible Moleküle können zum Beispiel einen hohen durchschnittlichen RMSD aufweisen und dadurch das Gesamtergebnis dominieren bzw. verzerren [56].

3.1.2. IBAC

IBAC (Interaction-Based Accuracy Classification) [59] bewertet Konformationen anhand des Vorhandenseins von für die Bindung an ein Zielprotein relevanten Interaktionen. Dazu wird zuerst die Kristallstruktur auf für die Bindung relevante Interaktionen wie zum Beispiel Wasserstoffbrücken, Salzbrücken und hydrophobe Kontakte untersucht. Anschließend werden die Interaktionen der Konformation gezählt und mit denen der Kristallstruktur verglichen. Zum Schluss wird die Konformation nach folgenden Kriterien klassifiziert:

- *correct:* Korrekte Orientierung und Konformation, alle wichtigen Interaktionen sind vorhanden
- *nearly correct:* Fast korrekte Orientierung und Konformation, wichtige Interaktionen fehlen
- *incorrect:* Konformationen, die weder *correct* noch *nearly correct* sind

Die Methode liefert eine gute Bewertung von Konformationen, da im Gegensatz zum RMSD die Interaktion mit dem Protein berücksichtigt wird.

Allerdings lässt sich die Methode nicht automatisieren, da sie auf visuelle Inspektion der Konformationen angewiesen ist. Des Weiteren ist die Methode nur auf Protein-Ligand-Komplexe anwendbar.

3.1.3. RSR

Die 3D-Koordinaten der Kristallstruktur eines Moleküls entsprechen nicht den ursprünglichen experimentellen Daten, sondern sind eher eine subjektive Interpretation der Elektronendichte [61, 62]. Bei der Interpretation der Elektronendichte entsteht also ein Modell des Moleküls und je besser die Auflösung ist, mit der die Elektronendichte gemessen wurde, desto genauer wird auch das Modell. Allerdings kann auch bei hoher Auflösung die Bestimmung eines individuellen Schweratoms zwischen 0,1Å und 0,5Å von der eigentlichen Position abweichen [63]. Dieser Fehler muss beim Vergleich von Atomkoordinaten mit berücksichtigt werden.

Der *RSR (Real Space R-factor)* [64] versucht dieses Problem zu umgehen, indem statt der Koordinaten die experimentell bestimmte Elektronendichte verwendet wird. Dabei wird gemessen, wie gut eine Konformation in die experimentell bestimmte Elektronendichte passt ($RSR_{Kristallstruktur}$), in dem diese mit einer aus der Konformation berechneten Elektronendichte verglichen wird ($RSR_{Konformation}$). Der *RSR* für eine Konformation n berechnet sich dann nach der folgenden Formel:

$$RSR_n = \frac{RSR_{Konformation}}{RSR_{Kristallstruktur}}$$

Ein $RSR_n < 1,7$ wird als *success* und ein $RSR_n \geq 1,7$ als *failure* klassifiziert.

Der Vorteil von RSR ist, dass die experimentellen Daten durch die Benutzung der Elektronendichte wesentlich realistischer repräsentiert werden und so Ungenauigkeiten bei der Einpassung in die Elektronendichte umgangen werden. Allerdings kann die Methode nur angewendet werden, wenn die experimentell bestimmte Elektronendichte vorhanden ist. Außerdem ist die Methode abhängig von der Auflösung, mit der die Elektronendichte gemessen wurde: ist diese zu hoch, ist der RSR zu sensitiv; ist die Auflösung zu niedrig, ist der RSR zu ungenau.

3.1.4. GARD

GARD (Generally Applicable Replacement of RMSD) [60] bewertet das Alignment zwischen den Atomen einer Referenzstruktur und den Atomen einer ihrer Konformationen. Die Bewertung des Alignments von zwei Atomen basiert auf ihrer geometrischen Distanz und einer Gewichtung bezüglich ihrer Relevanz für die Bindung an ein Protein. Das Alignment von zwei Konformationen wird mit der folgenden Formel berechnet:

$$GARD = \frac{\sum_{i=1}^{N} \delta_i \omega_i}{\sum_{i=1}^{N} \omega_i}$$

wobei N die Anzahl der Atome, δ_i die Bewertung des Alignments und ω_i das Gewicht von Atom i ist. GARD ist auf einen Wert zwischen 0 und 1 normalisiert, wobei 0 für das schlechteste und 1 für das beste Alignment steht. Die Gewichtung der Atome bzw. der funktionellen Gruppen wurde aus der statistischen Analyse von häufig an einer Bindung beteiligten funktionellen Gruppen abgeleitet.

Ein Vorteil von GARD gegenüber dem RMSD ist die Berücksichtigung funktioneller Gruppen beim Vergleich von zwei Konformationen. Des Weiteren kann die Gewichtungsfunktion ausgetauscht und die Berechnung einfach automatisiert werden. Die Entscheidung, ob eine funktionelle Gruppe wichtig für die Bindung an das Protein ist, hängt allerdings immer vom Protein ab. Ein und die selbe Gruppe kann wichtig für die Bindung an ein bestimmtes Protein, aber völlig unwichtig für die Bindung an ein anderes Protein sein. Um dies korrekt einschätzen zu können, müsste der Protein-Ligand-Komplex vorliegen.

3.1.5. TanimotoCombo

ROCS (Rapid Overlay of Chemical Structures) [65] ist ein kommerzielles Programm um sehr schnell die Form zweier Moleküle miteinander zu vergleichen. ROCS basiert auf der Idee, dass Moleküle eine ähnliche Form aufweisen, wenn sich ihre Volumina sehr gut überlagern lassen, und dass jede Abweichung von einer idealen Überlagerung ein Maß für ihre Unähnlichkeit ist. Das Volumen der Moleküle wird dabei durch Gaussfunktionen repräsentiert [7]. Bei der Bewertung der Überlagerung kann zusätzlich zum Vergleich der Form ein Vergleich der chemischen Eigenschaften berücksichtigt werden.

Die Bewertung kann dann entweder nach der Ähnlichkeit der Form (*TanimotoShape*), der chemischen Eigenschaften (*TanimotoColor*) oder einer Kombination von beiden (*TanimotoCombo*) vorgenommen werden. Die Bewertung durch TanimotoShape oder TanimotoColor ist auf einen Wert zwischen 0 und 1 und die Bewertung durch TanimotoCombo auf einen Wert zwischen 0 und 2 normalisiert. Je höher der Wert, desto besser die Überlagerung und desto ähnlicher sind die Moleküle.

ROCS ist ebenfalls für den Vergleich von Konformationen geeignet. Allerdings lässt sich die Methode nur aufwendig implementieren, da sie nicht komplett publiziert wurde.

3.2. Methoden zur Konformationsgenerierung

Im folgenden Abschnitt werden die gängigsten und am häufigsten benutzten Methoden zur Analyse des Konformationsraumes von kleinen Molekülen vorgestellt. Die Methoden lassen sich generell in die folgenden sechs Kategorien einteilen [9, 43]:

- *Systematische Suche*,
- *Wissensbasierte Ansätze*,
- *Zufällige Suche*,
- *Evolutionäre Algorithmen*,
- *Distance-Geometry*
- *Simulationsverfahren*

3.2.1. Systematische Suche

Bei der systematischen Suche werden Konformationen generiert, indem allen rotierbaren Bindungen eines Moleküls systematisch Torsionswinkelwerte zugewiesen werden. Der einfachste und älteste Algorithmus für die Systematische Suche, *grid search*, funktioniert wie folgt [43]:

1. Identifizierung aller rotierbaren Bindungen (Bindungslängen und Bindungswinkel bleiben starr)
2. Systematische Rotation der Bindungen von 0° bis 360°in Inkrementen einer konstanten Größe

3. Energieminimierung

Der Algorithmus stoppt, wenn alle möglichen Kombinationen von Torsionswinkeln generiert wurden. Wird das Inkrement entsprechend klein gewählt, ist dieser Algorithmus der einzige, der mit absoluter Sicherheit das globale Energieminimum bzw. die bioaktive Konformation finden kann [28], es werden dabei allerdings auch viele hochenergetische Konformationen generiert [28,43](siehe Abbildung 3.1). Ein größeres Problem ist die *kombinatorische Explosion*. Die Anzahl der generierten Konformationen wächst exponentiell mit der Anzahl der rotierbaren Bindungen [9,33,43]. Die Anzahl der generierten Konformationen K berechnet sich dabei folgendermaßen [43]:

$$K = \prod_{i=1}^{N} \frac{360}{\theta_i}$$

wobei N die Anzahl der rotierbaren Bindungen ist und θ_i das gewählte Torsionswinkel-Inkrement für Bindung i. Für ein Molekül mit drei rotierbaren Bindungen und einem Inkrement von 30° für jede rotierbare Bindung würden beispielsweise 1.782 Konformationen generiert werden. Bei sechs rotierbaren Bindungen würden fast drei Millionen Konformationen generiert werden, so dass auch die beste Implementierung der systematischen Suche ab einer bestimmten Anzahl rotierbarer Bindungen nicht mehr praktisch anwendbar ist [40].

Die Anzahl an generierten Konformationen lässt sich einschränken, indem bestimmte Konformationen, z.B. solche mit besonders hoher Energie, frühzeitig ausgeschlossen werden. Eine einfache Methode dies zu erreichen ist die Verwendung einer *Tiefensuche* in Verbindung mit *Pruning* [33]. Dazu wird zuerst bestimmt, in welcher Reihenfolge die einzelnen Torsionswinkel eingestellt werden. Die erste Konformation wird generiert, indem für jede rotierbare Bindung der erste Torsionswinkelwert eingestellt wird. Dann wird für die letzte rotierbare Bindung der nächste Torsionswinkel eingestellt, um die zweite Konformation zu generieren. Wenn auf diese Weise alle Torsionswinkel für die letzte rotierbare Bindung eingestellt wurden, wird zur vorletzten rotierbaren Bindung gewechselt und so weiter. Am Ende entsteht ein *Suchbaum*, bei dem die inneren Knoten Teilkonformationen und die *Blattknoten* „fertige“ Konformationen repräsentieren (siehe Abbildung 3.2). Wenn jetzt bereits bei einer Teilkonformation k Probleme auftauchen, wie zum Beispiel überlappende Atome, die zu einer hohen Energie führen, dann können alle im Teilbaum mit k als Wurzel liegenden Knoten verworfen werden (*Pruning*). Dabei ist es wichtig, dass bei den Molekülteilen, die zu dem Problem führen, die relative Lage der Molekülteile zueinander

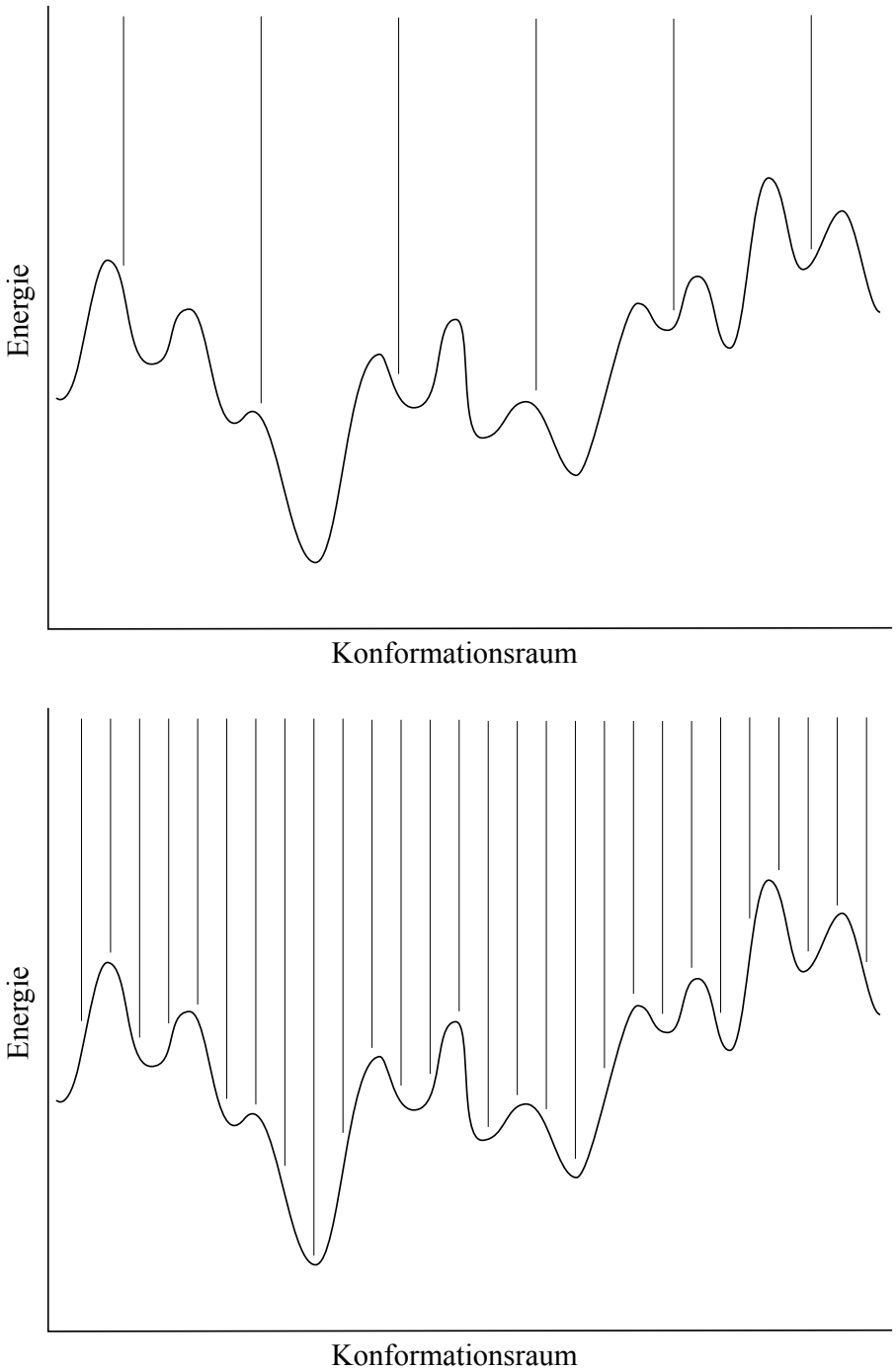


Abbildung 3.1.: Abtastung des Konformationsraumes beim *grid search*-Algorithmus mit einem großen Inkrement (oben) und einem kleinen Inkrement (unten).

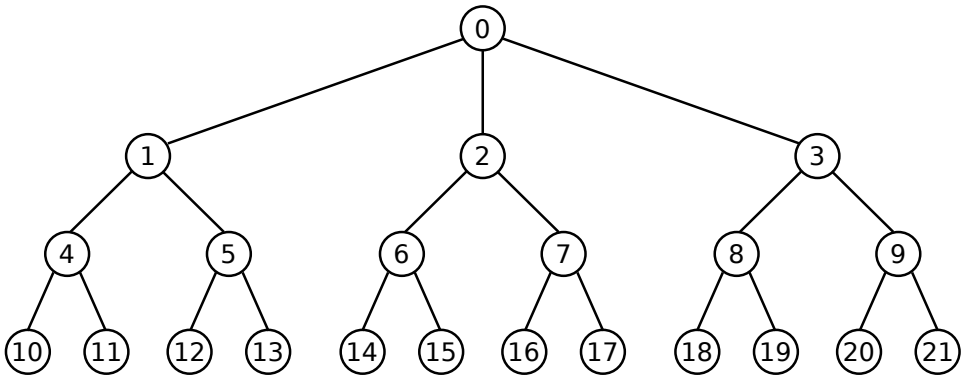


Abbildung 3.2.: Suchbaum der bei der Konformationsgenerierung mit Tiefensuche entsteht. In diesem einfachen Beispiel gibt es drei rotierbare Bindungen, von denen die erste drei mögliche Torsionswinkel und die anderen beiden zwei mögliche Torsionswinkel haben, woraus sich 12 mögliche Konformationen ergeben. Jeder Knoten repräsentiert einen Zustand in dem einer, zwei oder drei Torsionswinkel zugewiesen wurden. Die Reihenfolge in der die Knoten während der Tiefensuche durchgegangen werden ist: 0, 1, 4, 10, 4, 11, 4, 1, 5, 12, 5, 13, 5, 1, 0, 2, 6, 14, 6, 15, 6, 2, 7, 16, 7, 17, 7, 2, 0, 3, 8, 18, 8, 19, 8, 3, 9, 20, 9, 21. Dieses Beispiel wurde aus [33] übernommen.

nicht durch spätere Torsionswinkeleinstellungen verändert und das Problem somit eventuell wieder aufgelöst wird.

Eine weitere Methode, um die Menge der generierten Konformationen einzuschränken, ist die Löschung von redundanten Konformationen, d.h. Konformationen die sich sehr ähnlich sind. Die Ähnlichkeit wird dabei meistens mit Hilfe des RMSD bestimmt (siehe Abschnitt 3.1.1).

Bis jetzt wurde nur beschrieben, wie die systematische Konformationsgenerierung für azyklische Moleküle funktioniert. Der oben beschriebene Algorithmus kann allerdings auch auf Ringsysteme angewendet werden [9,43]. Dazu wird zuerst in jedem Ring eine Ringbindung entfernt, um ein „pseudoazyklisches“ Molekül zu erhalten, welches dann wie ein azyklisches Molekül behandelt werden kann. Um sicherzustellen, dass am Ende wieder korrekte Ringe gebildet werden, müssen verschiedene intramolekulare Parameter überprüft werden. Der wichtigste Parameter ist dabei das Ringschluss-Kriterium: Der Abstand der beiden Atome der entfernten Bindung muss innerhalb des Bereichs einer Bindungslänge liegen, so dass der Ring auch wieder geschlossen werden kann.

Die systematische Suche ist immer ein Kompromiss zwischen der Größe des Inkrements, das heißt der Granularität der Abtastung des Konformations-

raums und der damit verbundenen Anzahl an generierten Konformationen, und der Laufzeit [43]. Wird das Inkrement zu klein gewählt, ist die Konformationsgenerierung sehr zeitintensiv und es werden zu viele energetisch ungünstige Konformationen generiert. Wird das Inkrement zu groß gewählt, ist die Laufzeit zwar kürzer, aber es könnten bioaktive Konformationen eventuell nicht aufgezählt werden.

3.2.2. Wissensbasierte Ansätze

Bei den wissensbasierten Ansätzen wird aus experimentellen Daten oder theoretischen Untersuchungen gewonnenes Wissen zur Konformationsgenerierung eingesetzt. Das Wissen wird dabei entweder explizit (zum Beispiel durch Regeln) oder implizit (zum Beispiel durch Template mit erlaubten Ringkonformationen) eingesetzt [9].

Mit Hilfe von wissensbasierten Ansätzen kann das Problem der *kombinatorischen Explosion* teilweise eingeschränkt werden [43]. Werden für eine rotierbare Bindung in einer bestimmten chemischen Umgebung zum Beispiel nur zwei verschiedene Torsionswinkel beobachtet, dann kann daraus eine Regel abgeleitet werden, die die Anzahl der möglichen Torsionswinkel für diese Bindung bei der systematischen Suche von vornherein auf die beiden beobachteten Torsionswinkel begrenzt [22].

Ein weiterer Ansatz ist der Fragmentbasierte Ansatz. Dabei wird das Molekül zuerst in Fragmente zerlegt. Anschließend werden Konformationen der Fragmente zu Molekülkonformationen zusammengebaut. Da es gewöhnlicherweise weniger Kombinationen von Fragmentkonformationen gibt, als Torsionswinkelkombinationen, wird erwartet, dass dieser Ansatz effizienter als die systematische Suche ist [43]. Die möglichen Konformationen für bestimmte Fragmente können ebenfalls aus experimentellen Daten oder theoretischen Untersuchen abgeleitet werden und als sogenannte *Template* in einer Konformationsbibliothek gespeichert werden. Dieser Ansatz eignet sich auch für die Generierung von Ringkonformationen. Zum Beispiel benutzt das Programm CORINA [66] eine Bibliothek mit Ringtemplaten zur Generierung von Konformationen für kleine und mittelgroße Ringe.

Wissensbasierte Methoden bieten gegenüber der systematischen Suche nicht nur den Vorteil, dass das Problem der *kombinatorischen Explosion* eingeschränkt wird sondern auch, dass Konformationen generiert werden, die mit beobachteten Daten übereinstimmen. Zudem steigt die Zahl der Strukturen in chemischen Datenbanken wie der CSD oder der PDB stetig an, so dass

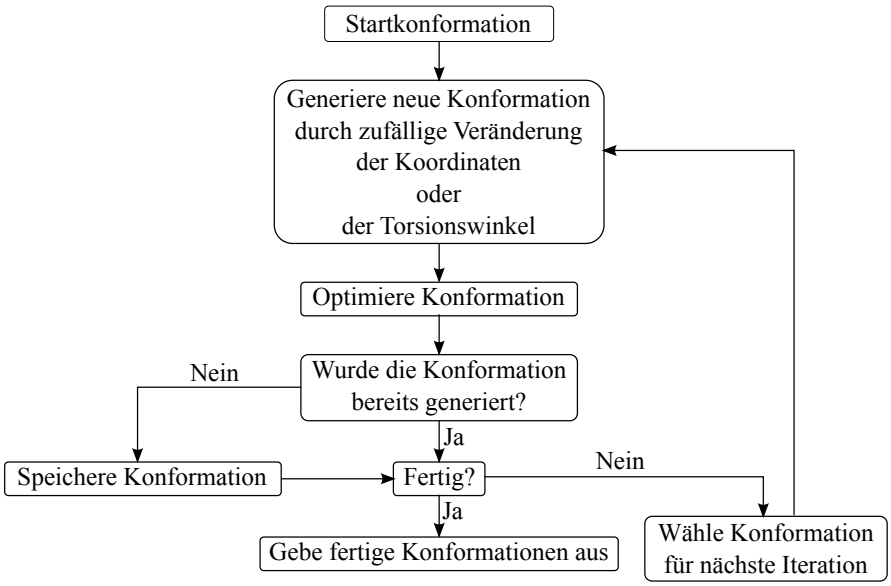


Abbildung 3.3.: Übersicht der Schritte zur Konformationsgenerierung bei der zufälligen Suche.

die Ableitung des Wissens immer besser und detaillierter wird (siehe auch Abschnitt 2.3).

3.2.3. Zufällige Suche

Im Gegensatz zur systematischen Suche, bei der Konformationen durch systematisches Ändern von Torsionswinkeln generiert werden, werden bei der zufälligen Suche Konformationen generiert, indem wiederholt entweder die kartesischen Koordinaten oder die Torsionswinkel eines Moleküls zufällig verändert werden [9, 43]. Bei jeder Iteration wird eine Konformation aus den bisher generierten Konformationen ausgewählt und zufällig verändert. Die Auswahl der zu verändernden Konformation kann dabei entweder zufällig erfolgen, oder es wird die in der vorherigen Iteration generierte Konformation verwendet [33]. Bei der ersten Iteration wird die Konformation des Eingabemoleküls verwendet. Anschließend wird die neu entstandene Konformation optimiert und mit allen bereits generierten Konformationen verglichen. Wenn die Konformation bisher noch nicht gefunden wurde, wird sie gespeichert. Dies wird so lange wiederholt, bis entweder eine bestimmte Anzahl an Iterationen erreicht wurde, oder keine neue Konformationen mehr gefunden werden kann (siehe auch Abbildung 3.3).

Ein Vorteil der zufälligen Suche ist, dass von einer Iteration zur nächsten komplett unterschiedliche Regionen des Konformationsraums untersucht werden können [9]. Sie ist außerdem besser geeignet, Konformationen für sehr flexible Moleküle zu erzeugen [40]. Allerdings ist die zufällige Suche im Gegensatz zur systematischen Suche nicht deterministisch, so dass auch bei langer Laufzeit nicht garantiert werden kann, dass das globale Minimum bzw. alle bioaktiven Konformationen gefunden werden. Zudem erhöht sich mit zunehmender Laufzeit auch die Wahrscheinlichkeit, dass redundante Strukturen (Duplikate) generiert werden [43].

3.2.4. Evolutionäre Algorithmen

Evolutionäre Algorithmen (EA) orientieren sich an den Grundprinzipien der biologischen Evolution und versuchen die optimale Lösung für ein Problem zu finden [9, 28, 43]. Am Anfang wird eine zufällige *Population* von möglichen Lösungen des Problems erzeugt. Die Mitglieder der Population werden anhand einer *Fitnessfunktion* bewertet. Anschließend werden die Mitglieder mit der besten Bewertung durch zufällige *Mutation* oder *Rekombination* verändert. Die übrigen Mitglieder der Population werden verworfen (*Selektion*) Die Population verändert sich so mit der Zeit und entwickelt sich zu einer Population mit besseren Lösungen.

Bei der Konformationsgenerierung entspricht die Start-Population einer Menge von zufällig generierten Konformationen des Eingabemoleküls. Die Torsionswinkelwerte jeder rotierbaren Bindung eines Mitglieds werden als *Chromosom* kodiert. Als Fitnessfunktion kann zum Beispiel die interne Energie benutzt werden. Aus der Start-Population werden dann mehrere Paare von Chromosomen ausgewählt und durch Mutation oder Rekombination verändert, um neue Konformationen für eine neue Population zu erzeugen. Dies wird solange wiederholt, bis eine vorgegeben Anzahl an Schritten erreicht ist, oder der Prozess konvergiert [43].

Ein Problem evolutionärer Algorithmen ist, dass wenige *fitte* Individuen zu einer frühzeitigen Konvergenz führen können, oder dass es andersherum zu lange dauert, bis eine optimale Lösung gefunden wird [28]. Durch die zufälligen Mutationen und Rekombinationen kann außerdem nicht sichergestellt werden, dass für ein bestimmtes Eingabemolekül immer die gleichen Konformationen generiert werden und dass das globale Minimum bzw. alle bioaktiven Konformationen gefunden werden. Verglichen mit der systematischen Suche sind evolutionäre Algorithmen allerdings wesentlich besser geeignet um Konformationen für sehr flexible Moleküle zu erzeugen

[28,43]. Ein weiterer Vorteil ist, dass nach jeder Iteration (auch bereits nach der ersten) eine Menge an Konformationen extrahiert werden kann, da immer eine Population mit den bisher besten Lösungen vorliegt [43].

3.2.5. Distance-Geometry

Der Distance-Geometry Ansatz ist ein weit verbreitetes Verfahren zur Generierung von Konformationen und gehört zu den sogenannten numerischen Methoden. Diese beruhen auf umfangreichen numerischen Optimierungsverfahren, die oft eine sehr lange Laufzeit haben [9]. Bei der Distance Geometry wird die Konformation eines Moleküls nicht durch die Kartesischen Koordinaten beschrieben, sondern anhand der Distanzen zwischen allen Paaren von Atomen des Moleküls. Die Distanzen werden dabei in einer *Distanzmatrix* gespeichert. Konformationen werden auf dieser Basis generiert, indem zufällig Distanzmatrizen generiert werden und diese anschließend wieder in Kartesische Koordinaten umgewandelt werden. Der entscheidende Punkt dabei ist, dass es nicht möglich ist, willkürliche Distanzen zuzuweisen. Die Distanzen zwischen Atome hängen von einander ab und viele Kombinationen sind geometrisch nicht möglich [43].

Die Konformationsgenerierung kann bei der Distance Geometry in vier Schritte eingeteilt werden [43]:

1. Berechnung einer Matrix mit Ober- und Untergrenzen für jede paarweise Atomdistanz des Eingabemoleküls.
2. Zuweisung von zufälligen Werte für jede Distanz innerhalb der vorgegebenen Grenzen.
3. Umwandlung der neuen Distanzmatrix in Kartesische Koordinaten.
4. Optimierung der neu generierten Konformation.

Die Ober- und Untergrenzen für die Atomdistanzen können aus experimentellen Daten abgeleitet werden. So können Konformationen erzeugt werden, die mit beobachteten Konformationen übereinstimmen [9]. Ein weiterer Vorteil ist, dass die Berechnung der Ringkonformationen bereits komplett im Algorithmus enthalten ist und daher keine separate Berechnung benötigt wird. Da die Zuweisung von neuen Distanzwerten zufällig passiert, hat die Methode der Distance Geometry die gleichen Nachteile wie die zufällige Suche. Es können auch hier redundante Konformationen generiert werden und es kann auch hier nicht sichergestellt werden, dass das globale Minimum bzw. alle bioaktiven Konformationen gefunden werden.

3.2.6. Simulationsverfahren

Die Simulation ist ein Verfahren, bei dem ein System mit seinen dynamischen Prozessen modelliert wird, um zu neuen Erkenntnissen zu gelangen, die auf die Realität übertragbar sind. Die wichtigsten Simulationsverfahren zur Generierung von Konformationen sind *Moleküldynamik (MD)*, *Monte-Carlo-Simulation* und *Simulierte Abkühlung (Simulated Annealing)* [9].

Bei der Moleküldynamik werden zeitabhängige Bewegungen und Konformationsänderungen eines Moleküls basierend auf den Newtonschen Gesetzen und unter vorgegebener Temperatur simuliert. Das Ergebnis einer Simulation ist eine *Trajektorie*, die beschreibt, wie sich die Positionen und Geschwindigkeiten der Atome mit der Zeit verändern. In festgelegten Abständen wird jeweils eine Konformation aus der Trajektorie extrahiert und anschließend energieminimiert. Bei der Generierung von Konformationen wird häufig eine sehr hohe, physikalisch unrealistische Temperatur verwendet, so dass das System in der Lage ist unterschiedliche Regionen der Energielandschaft zu erkunden und nicht in einem lokalen Minimum stecken bleibt [43].

Die Monte-Carlo-Simulation funktioniert ähnlich wie die zufällige Suche. Allerdings fehlt der Optimierungs-Schritt und für jede neu erzeugte Konformation wird anhand des *Metropolis-Kriteriums* entschieden, ob sie akzeptiert oder verworfen wird [9,43].

Bei der simulierten Abkühlung startet das System mit einer hohen Anfangstemperatur und wird anschließend mit Hilfe von MD in ein Temperaturgleichgewicht gebracht. Am Ende wird das System herunter gekühlt, wobei Konformationen mit niedriger Energie wahrscheinlicher werden. Bei einer Temperatur von 0 K sollte das System beim globalen Minimum angelangt sein [9].

Bei allen drei Verfahren kann nicht garantiert werden, dass das globale Minimum bzw. die bioaktiven Konformationen gefunden werden. Außerdem basieren die Verfahren auf aufwendigen numerischen Berechnungen, was eine sehr hohe Laufzeit zur Folge hat [9,28].

3.2.7. Generierung einer initialen 3D-Struktur

Die meisten der in diesem Abschnitt beschriebenen Verfahren benötigen eine initiale 3D-Struktur als Startkonformation, welche einen starken Einfluss auf die Generierung des Konformationsensembles haben kann. Abhängig

vom Startpunkt können bei den Simulationsverfahren zum Beispiel unterschiedliche Trajektorien entstehen. Bei einigen Trajektorien kann die Suche aufgrund unüberwindbarer Energiebarrieren in einem lokalen Minimum stecken bleiben und somit zu früh konvergieren. Je nach Startstruktur können also unterschiedliche Konformationen generiert werden. Systematische Methoden können durch die Wahl von Bindungslängen und insbesondere Bindungswinkeln beeinflusst werden. Ungünstig gewählte Bindungswinkel können zum Beispiel dazu führen, dass Atome schneller überlappen wodurch eine zu hohe Energie für die Teilkonformation entsteht, die dann verworfen wird [40].

Die Generierung einer 3D-Struktur aus einem 2D-Molekülgraphen ist ein ähnlich komplexes Problem wie die Generierung eines Konformationsensembles und die Methoden überlappen teilweise. Methoden zur Generierung einer initialen 3D-Struktur legen dabei nicht unbedingt Wert darauf das globale Minimum oder die bioaktive Konformation zu finden, sondern es wird eher versucht eine „vernünftige“ und energiearme Konformation zu generieren [40].

3.3. Software zur Konformationsgenerierung

Innerhalb der letzten 30 Jahre wurden verschiedene Ansätze und Programme zur Generierung von 3D-Strukturen und Konformationsensembles entwickelt. Den Anfang machten Ende der 80er Jahre unter anderem die beiden Programme *CONCORD* [67] und *CORINA* [66], welche mit Hilfe eines wissensbasierten Ansatzes eine initiale 3D-Struktur erzeugen. Später folgten dann eigenständige Programme zur Generierung von Konformationsensembles wie zum Beispiel *MIMUMBA* [22], *ROTATE* [26], *CEASAR* [13] und *TCG* [27]. Die meisten dieser Programmen benutzten einen der im vorherigen Abschnitt beschriebenen Ansätze zur Generierung von Konformationen. Neuere Methoden benutzten teilweise auch eine Kombination der Ansätze (zum Beispiel *Frog2* [19, 20]) oder bestehen aus mehreren Modulen, die sich in den jeweils verwendeten Methode unterscheiden (zum Beispiel *MOE* [23]). Eine Übersicht verschiedener Programme zur Konformationsgenerierung ist in Tabelle 3.1 zusammengefasst. Da die Methoden bereits ausführlich im vorherigen Abschnitt beschrieben wurden, werden im folgenden nur die drei häufig genutzten Programme *Catalyst* [11], *OMEGA* [24] und *ConfGen* [16] und das am Zentrum für Bioinformatik entwickelte Programm zur Generierung von initialen 3D-Strukturen, *coord3d* [68], näher beschrieben.

Tabelle 3.1.: Übersicht verschiedener Programme zur Generierung von Konformationen. Die alphabetische Auflistung ist nicht vollständig.

Name	Methode	Referenz
Balloon	Evolutionärer Algorithmus	[12]
Catalyst	Systematische Suche, Distance Geometry	[11]
CEASAR	Systematische Suche	[13]
CONAN	Systematische Suche	[14]
CONCORD	Systematische Suche, Wissensbasierter Ansatz	[67]
Confab	Systematische Suche	[15]
ConfGen	Wissensbasierter Ansatz	[16]
Confort	Systematische Suche	[17]
CORINA	Systematische Suche, Wissensbasierter Ansatz	[66]
Cyndi	Evolutionärer Algorithmus	[18]
Frog2	Wissensbasierter Ansatz, Monte-Carlo-Simulation	[19,20]
MacroModel	Systematische Suche, Zufällige Suche	[21]
MIMUMBA	Systematische Suche, Wissensbasierter Ansatz	[22]
MOE	Systematische Suche, Zufällige Suche	[23]
OMEGA	Wissensbasierter Ansatz	[24]
RDKit	Distance Geometry	[25]
ROTATE	Wissensbasierter Ansatz	[26]
TCG	Wissensbasierter Ansatz	[27]

Eine sehr gute aktuelle Übersicht und nähere Beschreibung von Programmen zur Konformationsgenerierung findet sich in [8]. Eine Übersicht über freie Programme zur Konformationsgenerierung findet sich in [69].

3.3.1. Catalyst

Das Programm Catalyst [11] der Firma Accelrys besitzt zwei verschiedene Modi – *fast* und *best* – zur Generierung von Konformationen, welche sich in ihrer zugrunde liegenden Methode unterscheiden.

Beim *fast*-Modus werden die Ringsysteme und die azyklischen Teile des Moleküls getrennt voneinander behandelt. Für die Ringsysteme wird eine Bibliothek mit vordefinierten Ringkonformationen (Templaten) benutzt. Für die azyklischen Teile des Moleküls wird eine modifizierte systematische Suche (auch *quasi-exhaustive search* genannt) verwendet, bei der Torsionswinkel anhand eines *fuzzy grid* eingestellt werden. Im nächsten Schritt werden die generierten Konformationen mit einem modifizierten CHARMM-

Kraftfeld [70] optimiert, was dafür sorgt, dass Torsionswinkel nur innerhalb eines bestimmten Bereichs verändert und Duplikate ausgeschlossen werden. Im letzten Schritt wird die Anzahl der Konformationen mit Hilfe einer simplen Heuristik reduziert, wobei darauf geachtet wird, ein möglichst diverses Konformationsensemble zu erhalten. Der *fast*-Modus ist schnell und eignet sich besonders gut, um Konformationen für eine sehr große Menge von Molekülen zu erzeugen. [8,9,58,71].

Beim *best*-Modus wird ein Distance-Geometry-Ansatz benutzt, um den Konformationsraum gründlicher abzusuchen als beim *fast*-Modus. Auch hier wird das modifizierte CHARMM-Kraftfeld zur Optimierung benutzt. Eine *poling* [72] genannte Methode wird eingesetzt, um Konformationen zu generieren die zwar weit von einem lokalen Energieminimum entfernt sind, aber eine ähnliche Energie zueinander haben. Mit Hilfe dieser Methode lassen sich Regionen des Konformationsraums mit niedriger Energie absuchen und Konformationen generieren, die nicht in einem lokalen Energieminimum liegen. Der *best*-Modus ist zwar laufzeitintensiver als der *fast*-Modus, kann dafür aber besser bioaktive Konformationen reproduzieren [8,58,71].

3.3.2. OMEGA

OMEGA [24] benutzt einen wissensbasierten Ansatz zur Generierung eines Konformationsensembles. Der Algorithmus gliedert sich dabei in drei Phasen:

1. Zusammenbau einer initialen 3D-Struktur aus einer Fragmentbibliothek. Die Bibliothek wurde aus einer großen Sammlung kommerziell erhältlicher Moleküle erstellt und enthält ein oder mehrere Konformationen pro Fragment (eine Konformation für azyklische Fragmente und starre Ringe; mehrere Konformationen für flexible Ringe). Das Eingabemolekül wird dabei nach den gleichen Regeln fragmentiert, die auch bei der Erstellung der Fragmentbibliothek verwendet wurden.

2. Systematische Generierung von Konformationen anhand einer Torsionsbibliothek. Die Torsionsbibliothek enthält eine hierarchische Sammlung von Torsionsregeln, die so geordnet sind, dass jeder rotierbaren Bindung eines Moleküls mindestens eine der Regeln zugeordnet werden kann. Jede Torsionsregel enthält eine Liste von Torsionswinkeln, die aus der Analyse von Kristallstrukturen und aus Kraftfeldberechnungen (MMFF94 [73]) abgeleitet wurden. Konformationen mit überlappenden Atomen werden verworfen.

3. Zusammenstellung eines Konformationsensembles. Dazu werden die im vorherigen Schritt generierten Konformationen mit Hilfe einer kraftfeldbasierten Bewertungsfunktion (modifiziertes MMFF94) nach absteigender Bewertung sortiert. Beginnend mit der am besten bewerteten Konformation werden alle schlechter bewerteten Konformationen, deren Abstand zur am besten bewerteten Konformation unterhalb eines vom Benutzer definierten RMSD liegt, verworfen. Dieser Prozess wird solange mit der nächsten Konformation in der geordneten Liste wiederholt, bis entweder nur noch eine bestimmte Anzahl an Konformationen übrig ist, oder nur noch Konformationen mit einer Bewertung oberhalb eines vorher definierten Wertes übrig sind.

Evaluierungen haben gezeigt, dass OMEGA in der Lage ist, schnell Konformationsensembles zu generieren, und gute Ergebnisse bei der Reproduktion von bioaktiven Konformationen erreicht [8,24,58].

3.3.3. ConfGen

ConfGen [16], basiert ebenfalls auf einem wissensbasierten Ansatz und wurde ursprünglich entwickelt um Konformationen für das Docking-Programm Glide [74,75] zu erzeugen. Die Generierung von Konformationen erfolgt in drei Schritten:

1. Identifizierung der variablen Molekülteile. In diesem Schritt werden rotierbare Bindungen, flexible Ringe und invertierbare Stickstoffatome identifiziert. Für flexible Ringe wird in einer Bibliothek mit von MacroModel [21] vorberechneten Ringkonformationen nach passenden Konformationen gesucht.

2. Generierung von Konformationen. Für jede rotierbare Bindung wird mit einer modifizierten Version des OPLS_2001-Kraftfelds [76,77] ein Torsionspotenzial berechnet. Die Energieminima der Potenziale werden anschließend zur Einstellung der Torsionswinkel benutzt. Konformationen mit zu hoher Energie oder überlappenden Atomen werden verworfen. Die Konformationen werden so sortiert, dass die eher räumlich ausgestreckten Konformationen weiter oben in der Liste stehen.

3. Auswahl und Optimierung der Konformationen. Im letzten Schritt werden Konformationen aussortiert, die ungewollte elektrostatische Eigenschaften, polare Kontakte oder eine hohe lokale Konzentration von Schweratomen haben. Aus der sortierten Liste wird eine vom

Benutzer vorgegebene Anzahl von Konformationen extrahiert und mit dem OPLS_2005-Kraftfeld [78] optimiert. Duplikate oder Konformationen, die sich zu ähnlich sind (auf Grundlage des RMSD), werden ebenfalls verworfen.

ConfGen besitzt verschiedene Modi zum Generieren von Konformationen. Evaluierungen haben gezeigt, dass ConfGen einerseits gut in der Lage ist die bioaktive Konformation zu reproduzieren und anderseits schnell kleine Konformationsensembles für eine sehr große Menge von Molekülen erzeugen kann [16,79]. Im Vergleich mit dem *best*-Modus von Catalyst erzielt ConfGen ähnlich gute Ergebnisse bei der Reproduktion der bioaktiven Konformation, ist dabei aber eine Größenordnung schneller als Catalyst [79].

3.3.4. Nachteile bestehender Programme

Alle drei vorgestellten Programme sind zwar in der Lage relativ schnell sinnvolle Konformationsensembles zu erzeugen, allerdings ist die Generierung der Konformationen nicht transparent, dass heißt für den Benutzer ist nicht nachvollziehbar, warum bestimmte Konformationen generiert werden. Des Weiteren kann der Benutzter durch Parametereinstellungen zwar Einfluss auf die Konformationsgenerierung nehmen, das zugrunde liegende Konformationsmodell kann aber zumindest bei Catalyst und ConfGen nicht angepasst werden. Bei OMEGA lassen sich zwar eigene Torsionsregeln hinzufügen, aber die eigentliche Torsionsbibliothek kann nicht verändert werden.

3.3.5. coord3d

Im Gegensatz zu Catalyst, OMEGA und ConfGen, die ein Konformationsensemble generieren, nutzt das von Therese Inhester im Rahmen ihrer Masterarbeit entwickelte Programm coord3d [68] einen wissensbasierten Ansatz, um eine initiale 3D-Struktur zu erzeugen. Dazu wird das Eingabemolekül zuerst in Ringsysteme und azyklische Teile aufgeteilt. Das Eingabemolekül kann dabei entweder als 2D- oder 3D-Struktur vorliegen. Wenn das Eingabemolekül bereits 3D-Koordinaten besitzt, werden diese verworfen. Im nächsten Schritt werden dann Bindungslängen und Bindungswinkel zugewiesen. Generell werden die Werte für azyklische Bindungslängen aus der Summe der kovalenten Radien [80] der benachbarten Atome abgeleitet. Weicht die so abgeleitete Bindungslänge um mehr als 0,05Å von

experimentell beobachteten Bindungslängen ab, wird eine detailliertere Klassifizierung anhand der Valenzzustände und Ladung der beteiligten Atome vorgenommen. Die Bindungswinkel für azyklische Bindungen sind auf dem VSEPR-Modell basierte idealisierte Werte. Wie bei den Bindungslängen werden Bindungswinkel, die mehr als 5% von experimentell beobachteten Daten abweichen noch einmal überarbeitet. Für Torsionswinkel nicht rotierbarer Bindungen wird der durchschnittliche beobachtete Wert aus experimentell bestimmten Strukturen verwendet. Die Torsionswinkel rotierbarer Bindungen werden dann so eingestellt, dass Atome nicht überlappen und eine räumlich gestreckte Konformation entsteht. Die Koordinaten für Ringsysteme werden separat generiert (siehe Abschnitt 4.3.2) und anschließend transformiert, so dass die neu generierte Ringkonformation korrekt mit dem azyklischen Teil verbunden werden kann. Im letzten Schritt werden zwei verschiedene Strategien angewandt, um überlappende Atome zu beseitigen. Diese basieren auf der Veränderung aufeinander folgender Torsionswinkel.

Vergleiche mit CORINA haben gezeigt, dass coord3d zwar langsamer als CORINA ist, dafür aber wesentlich weniger 3D-Strukturen mit überlappenden Atomen erzeugt [68].

3.4. Software zur Konformationsanalyse

Bei der Konformationsanalyse werden häufig die beiden Programme *ConQuest* [81] und *Mogul* [82,83] verwendet, welche im folgenden kurz beschrieben werden.

3.4.1. ConQuest

ConQuest [81] ist das Standardprogramm um in der CSD nach Strukturen zu suchen und Informationen über die Moleküle zu erhalten. Die CSD kann dabei anhand verschiedener Kriterien (wie zum Beispiel Molekülname, Molekülformel oder Literaturreferenzen) durchsucht werden. Es können auch Substrukturen zur Anfrage benutzt werden, wobei chemische Eigenschaften wie zum Beispiel Ladung oder Hybridisierung oder geometrische Eigenschaften wie zum Beispiel Bindungs- oder Torsionswinkel berücksichtigt werden können. Die gefundenen Moleküle können entweder in 2D oder 3D durchgesehen und statistisch analysiert werden. Durch die vielen verschiedenen Möglichkeiten erlaubt ConQuest dem Benutzer, flexible und individuelle Anfragen zu stellen. Diese müssen einerseits präzise genug

definiert sein, um die richtigen Information zu finden, aber andererseits flexibel genug sein, um ausreichend Daten für statistische Analysen zu erhalten.

3.4.2. Mogul

Das Programm Mogul [82,83] ermöglicht die Analyse von Bindungslängen, Bindungswinkeln, Torsionswinkeln und Ringkonformationen eines Eingabemoleküls anhand bevorzugter Geometrien von Molekülen in der CSD. Dazu wird jedem Fragment des Eingabemoleküls eine Menge von Schlüsselwerten zugewiesen, die die Umgebung des Fragments beschreiben. Danach wird mit Hilfe eines Suchbaums nach allen Fragmenten in der CSD mit identischen Schlüsselwerten gesucht. Wenn die Suche zu wenig Treffer liefert, wird anhand von *backtracking* und einem Ähnlichkeitsmaß nach Fragmenten mit ähnlichen Schlüsselwerten gesucht. Die gefundenen Fragmente werden anschließend auf geometrische Eigenschaften untersucht und die Ergebnisse in Form von Statistiken wie zum Beispiel Durchschnittswerte, Standardabweichung und als Histogramme angegeben. Die Analyse von Molekülen kann dabei entweder über eine graphische Oberfläche oder über die Kommandozeile erfolgen [82,84].

Die Analyse von Konformationen in Mogul ist schnell und vor allem über die graphische Oberfläche nach einer gewissen Einarbeitungsphase relativ einfach. Ein Problem ist allerdings die automatische Definition der chemischen Umgebung der untersuchten Fragmente. Für den Benutzer ist dabei nicht ersichtlich, welche Umgebung für die Suche verwendet wird. Des Weiteren können bei der Analyse eines Moleküls nur die Moleküle der CSD durchsucht werden. Es ist nicht möglich eigene Datensätze anzugeben.

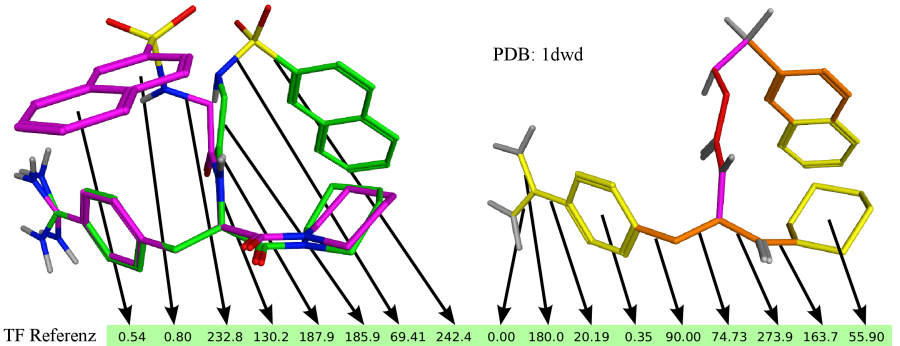
Kapitel 4

4 Methoden

In diesem Kapitel werden die im Rahmen dieser Arbeit entwickelten Methoden zur Analyse und Generierung von Molekülkonformationen näher beschrieben. Das Kapitel gliedert sich dabei in vier Teile. Im ersten Teil wird der *TFD*, eine neue Methode zum Vergleich von Konformationen beschrieben. Im zweiten Teil wird das Konzept der *Torsionsbibliothek*, welche sowohl für die Analyse als auch für die Generierung von Konformationen verwendet wird, vorgestellt und erläutert. Der dritte Teil beschäftigt sich mit *CONFECT*, einer neuen wissensbasierten Methode zur Generierung von Konformationen. Im letzten Teil wird der *TorsionAnalyzer*, ein graphisches Softwarewerkzeug zur Analyse und Generierung von Molekülkonformationen, vorgestellt.

4.1. Torsion-Fingerprint-Deviation

Die *Torsion-Fingerprint-Deviation* (*TFD*) [30] ist eine Methode zum Vergleich von Molekülkonformationen. Da sich Konformationen eines Moleküls hauptsächlich in ihren Torsionswinkeln unterscheiden, kann man diese nutzen, um Konformationen eines Moleküls zu beschreiben und miteinander zu vergleichen. Dabei beschreibt der *Torsion-Fingerprint (TF)* eines Moleküles die Torsionswinkel aller seiner Bindungen und Ringe mit Hilfe von numerischen Werten. Die Differenz (TFD) zwischen zwei TFs, also dem TF einer Referenz-Konformation und dem TF einer vorhergesagten Konformation des selben Moleküls, berechnet sich wie folgt (eine genauere Beschreibung der einzelnen Schritte erfolgt im weiteren Verlauf dieses Kapitels):



TF Referenz	0.54	0.80	232.8	130.2	187.9	185.9	69.41	242.4	0.00	180.0	20.19	0.35	90.00	74.73	273.9	163.7	55.90	
TF Konformation	0.54	0.80	150.2	144.7	179.9	270.1	270.0	89.99	0.00	180.0	0.00	0.35	90.00	63.62	275.1	179.9	54.21	
Differenz	0.00	0.00	82.60	14.50	8.00	84.20	159.4	152.5	0.00	0.00	20.19	0.00	0.00	11.11	1.20	16.22	1.69	
Normalisierte Differenz	0.000	0.000	0.459	0.080	0.045	0.468	0.886	0.847	0.000	0.000	0.112	0.000	0.000	0.062	0.006	0.090	0.009	
Gewichtung	0.011	0.175	0.892	0.635	0.892	1.000	0.635	0.359	0.000	0.000	0.001	0.050	0.162	0.359	0.359	0.162	0.050	Summe 5.742
Gewichtete Differenz	0.000	0.000	0.409	0.051	0.040	0.468	0.563	0.304	0.000	0.000	0.000	0.000	0.000	0.022	0.002	0.015	0.000	1.874

TFD 0.326

Abbildung 4.1.: TFD-Beispielberechnung für eine Konformation des Liganden aus dem PDB-Komplex 1dwd [85]. Die Referenz-Struktur ist in grün, die Konformation in pink dargestellt. Die Einfärbung der rechten Struktur entspricht der Gauß-Gewichtung. Die Torsionswinkel in jedem Fingerprint sind in Grad angegeben. Werte mit einem blauen Rahmen heben Fälle hervor, in denen Symmetrie eine Rolle spielt.

1. Für jede Bindung und jedes Ringsystem wird die Differenz der (Pseudo-) Torsionswinkel berechnet.

2. Jede dieser Differenzen wird auf eine Zahl zwischen 0 (keine Abweichung) und 1 (maximale Abweichung) normalisiert.

3. Um sicher zu stellen, dass Differenzen an topologisch zentralen Bindungen und Ringen einen höheren Einfluss auf den TFD haben als Differenzen an terminalen Bindungen und Ringen, werden die normalisierten Differenzen mit Hilfe einer Gauß-Funktion gewichtet.

4. Um den endgültigen TFD zu berechnen wird die Summe der gewichteten Differenzen durch die Summe der Gauß-Gewichte geteilt.

Abbildung 4.1 zeigt eine TFD-Beispielberechnung.

4.1.1. TF-Berechnung

Die Berechnung des TF für ein Molekül erfolgt in zwei Schritten. Im ersten Schritt werden die Torsionswinkel jeder Bindung berechnet. Davon ausgenommen sind Ringbindungen und Bindungen zu terminalen Atomen. In diesem Schritt ist es wichtig, dass die vier Atome, zwischen denen der Torsionswinkel berechnet wird, eindeutig sind, damit bei der späteren Berechnung der Differenz zu einem anderen TF eine eindeutige Zuordnung der Torsionswinkel erfolgen kann. Des Weiteren müssen Symmetrien im Molekül (mit) berücksichtigt werden. Hierzu werden für jede Bindung die Nachbaratome der beiden durch die gerade betrachtete Bindung verbundenen Atome gezählt. Der einfachste Fall ergibt sich, wenn sich auf jeder Seite nur ein weiteres Atom befindet (siehe Fall 1 in Abbildung 4.2 und Tabelle 4.1). In diesem Fall können die beiden Atome als eindeutige Referenz für die Berechnung des Torsionswinkels benutzt werden. So lange keine symmetrischen Substituenten unter den Nachbaratomen vorkommen, können die Torsionswinkel berechnet werden, in dem zwei beliebige benachbarte Atome als Referenzatome ausgewählt und gespeichert werden. Unter der Annahme, dass die Bindungswinkel konstant sind, ist die Torsionswinkel-Differenz unabhängig von den ausgewählten Atomen. Ohne Betrachtung der Drehrichtung beträgt die maximale Differenz zwischen zwei Torsionswinkeln 180 Grad. Wenn im Molekül Symmetrien vorkommen, also mindestens zwei der Nachbaratome topologisch äquivalent sind, ändert sich die maximale Differenz. Hierbei müssen mehrere Fälle unterschieden werden. Wenn zwei topologisch äquivalente Atome mit einem Atom der gerade betrachteten Bindung in planar-trigonaler Geometrie verbunden sind (siehe Fall 3 in Abbildung 4.2 und Tabelle 4.1), wird dasjenige Atom als Referenz ausgewählt, welches bei der Berechnung des Torsionswinkels den kleinsten Wert ergibt. Die maximale Differenz beträgt in diesem Fall 90 Grad. Alle weiteren Fälle sind in Abbildung 4.2 und Tabelle 4.1 zusammengefasst. In einigen Fällen wird eine große Zahl an zusätzlichen Torsionswinkelberechnungen durchgeführt. In Fall 21 in Abbildung 4.2 und Tabelle 4.1 werden zum Beispiel neun verschiedene Torsionswinkel berechnet. Wenn bei diesem Fall die Atome b und c eine perfekte tetraedrische Geometrie haben, können die neun Berechnungen auf drei reduziert werden. Da nicht davon ausgegangen werden kann, dass in jedem Fall eine perfekte Geometrie vorliegt, wurden die zusätzlichen Torsionswinkelberechnungen eingeführt, um immer eine korrekte Berechnung der Torsionswinkel sicher zu stellen.

Im zweiten Schritt der TF-Berechnung werden die Ringe des Moleküls betrachtet. Die Ringe sind hier definiert als das „set of relevant cycles“, welches mit der Methode von Vismara [86] berechnet wird. Um Ringkonformationen

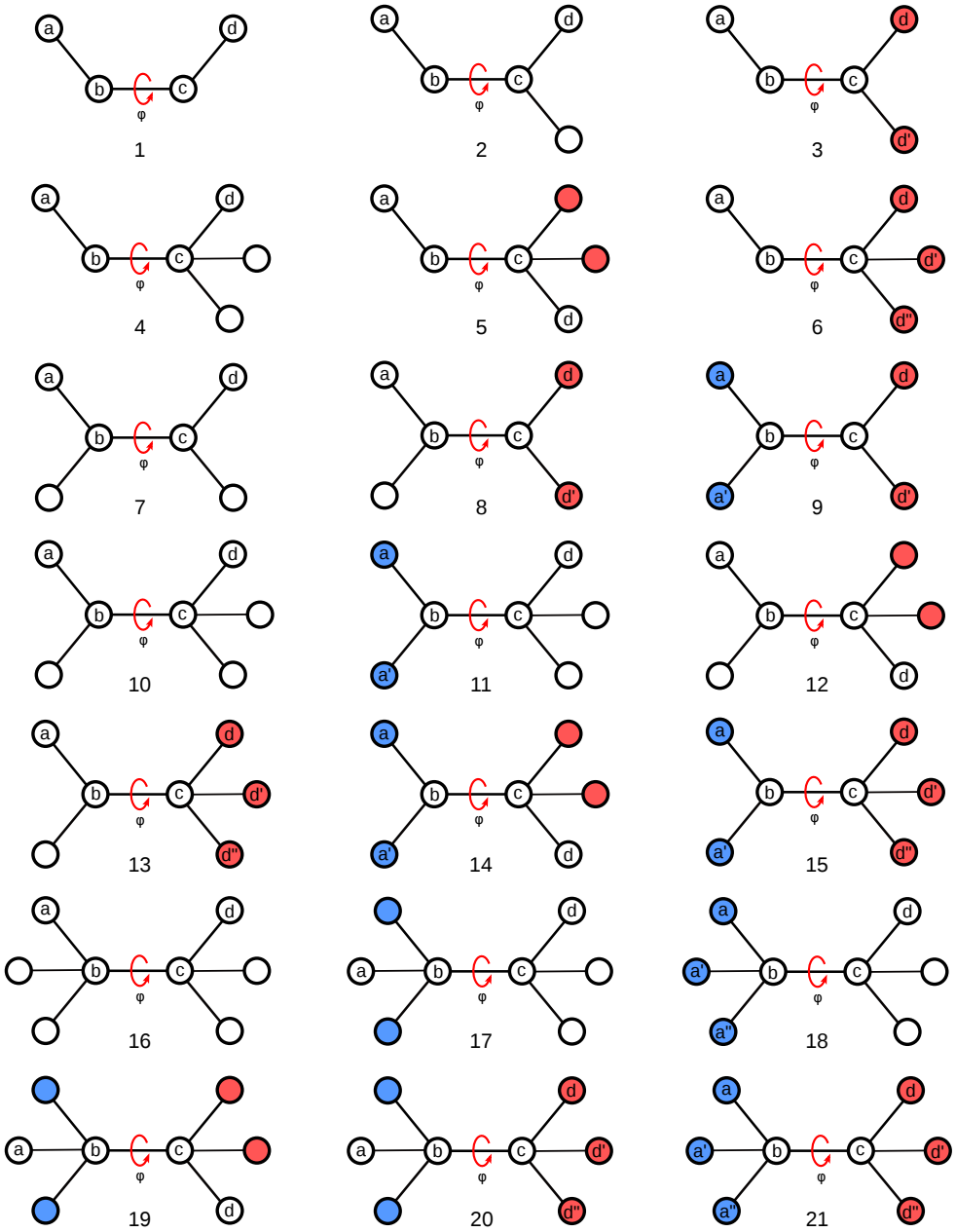


Abbildung 4.2.: Fallunterscheidung bei der TF-Berechnung: φ: Torsionswinkel, weiße Kreise: Atome, rote und blaue Kreise: topologisch äquivalente Atome. Siehe Tabelle 4.1 für die Berechnung der Torsionswinkel in den unterschiedlichen Fällen.

Tabelle 4.1.: Fallunterscheidung bei der TF-Berechnung. Die Nummern für jeden Fall (Spalte 1) beziehen sich auf die Darstellungen in Abbildung 4.2. $T(a,b,c,d)$ bezeichnet den Torsionswinkel zwischen den Atomen a, b, c, und d wobei b,c benachbart zur gerade betrachteten Bindung und a und d die Referenzatome sind. LP („lone pair") bezeichnet ein freies Elektronenpaar. Die Werte in der Spalte „Max Diff" stehen für die maximal mögliche Differenz dieses Torsionswinkels in zwei unterschiedlichen Konformationen unter der Annahme einer idealen Bindungswinkel-Geometrie.

Fall	Berechnung des Torsionswinkels	max Diff
1	$T(a,b,c,d)$	180°
2	$T(a,b,c,d)$. Speichere d	180°
3	$\min\{T(a,b,c,d), T(a,b,c,d')\}$	90°
	wenn Atom c ein trigonal pyramidales N ist: $T(a,b,c,LP)$	180°
4	$T(a,b,c,d)$. Speichere d	180°
5	$T(a,b,c,d)$	180°
6	$\min\{T(a,b,c,d), T(a,b,c,d'), T(a,b,c,d'')\}$	60°
7	$T(a,b,c,d)$. Speichere a,d	180°
8	$\min\{T(a,b,c,d), T(a,b,c,d')\}$. Speichere a	90°
	wenn Atom c ein trigonal pyramidales N ist: $T(a,b,c,LP)$. Speichere a	180°
9	$\min\{T(a,b,c,d), T(a,b,c,d'), T(a',b,c,d), T(a',b,c,d')\}$	90°
	wenn Atom c ein trigonal pyramidales N ist: $\min\{T(a,b,c,LP), T(a',b,c,LP)\}$	90°
	wenn die Atome b und c trigonal pyramidale N sind: $T(LP,b,c,LP)$	180°
10	$T(a,b,c,d)$. Speichere a,d	180°
11	$\min\{T(a,b,c,d), T(a',b,c,d)\}$. Speichere d	90°
	wenn Atom b ein trigonal pyramidales N ist: $T(LP,b,c,d)$. Speichere d	180°
12	$T(a,b,c,d)$. Speichere a	180°
13	$\min\{T(a,b,c,d), T(a,b,c,d'), T(a,b,c,d'')\}$. Speichere a	60°
14	$\min\{T(a,b,c,d), T(a',b,c,d)\}$	90°
	Atom b ein trigonal pyramidales N ist: $T(LP,b,c,d)$	180°
15	$\min\{T(a,b,c,d), T(a,b,c,d'), T(a,b,c,d''), T(a',b,c,d), T(a',b,c,d'), T(a',b,c,d'')\}$	30°
	wenn Atom b ein trigonal pyramidales N ist: $\min\{T(LP,b,c,d), T(LP,b,c,d'), T(LP,b,c,d'')\}$	60°
16	$T(a,b,c,d)$. Speichere a, d	180°
17	$T(a,b,c,d)$. Speichere d	180°
18	$\min\{T(a,b,c,d), T(a',b,c,d), T(a'',b,c,d)\}$. Speichere d	60°
19	$T(a,b,c,d)$	180°
20	$\min\{T(a,b,c,d), T(a,b,c,d'), T(a,b,c,d'')\}$	60°
21	$\min\{T(a,b,c,d), T(a,b,c,d'), T(a,b,c,d''), T(a',b,c,d), T(a',b,c,d'), T(a',b,c,d''), T(a'',b,c,d), T(a'',b,c,d'), T(a'',b,c,d'')\}$	60°

bei der TFD-Berechnung berücksichtigen zu können, wird ein kompatibles Torsionswinkel-Maß (*Ringtorsion*) für Ringe benötigt. Die Ringtorsion ist äquivalent zu dem Torsionswinkel einer einzelnen Bindung und erlaubt es die Ringkonformation mit einem einzelnen Wert zu beschreiben. Um die Ringtorsion für einen Ring zu berechnen, werden zuerst alle Torsionswinkel der Ringbindungen berechnet, wobei nur Ringatome als Referenzatome benutzt werden. Anschließend wird die Summe der absoluten Torsionswinkel durch die Größe des Rings (Anzahl der Ringbindungen, *rSize*) geteilt, um den finalen Wert zu erhalten. Der letzte Schritt in dieser Berechnung ist die Bestimmung der maximal möglichen Differenz für eine Ringtorsion. Hierzu wird die folgende Gauß-Funktion für Ringe mit 3 bis 14 Bindungen benutzt:

$$maxDev(rSize) = 180e^{-0.025(rSize-14)^2} \tag{4.1}$$

Für Ringe mit mehr als 14 Bindungen beträgt die maximale Differenz 180°.

4.1.2. Gewichtung

Die Rotation um eine topologisch zentrale Bindung, bzw. die Konformationsänderung eines topologisch zentralen Ringes hat einen viel stärkeren Einfluss auf die Konformation eines Moleküls als die Rotation um eine terminale Bindung, bzw. die Konformationsänderung eines terminalen Ringes. Um diesen Einfluss im TFD abzubilden, werden die Bindungen und Ringe je nach Position im Molekül unterschiedlich mit Hilfe einer Gauß-Funktion gewichtet. Der erste Schritt ist hierbei die Bestimmung der topologisch zentralen Bindung eines Moleküls. Hierzu wird der kürzeste Weg $\delta(b_1, b_2)$ zwischen allen Paaren von Bindungen (b_1, b_2) (ausgenommen Wasserstoffatome) mit Hilfe des Floyd-Warshall-Algorithmus [87] berechnet. Die zentrale Bindung c wird dann als die Bindung definiert, welche die kleinste Standardabweichung über alle kürzesten Wege hat. Die folgende Gauß-Funktion wird benutzt um die berechneten Distanzwerte in normalisierte Gewichte zu überführen:

$$w(b) = e^{-\beta(\delta(b,c))^2} \tag{4.2}$$

Nach Betrachtung einiger initialer Testfälle wurde entschieden, dass eine Bindung, welche auf halbem Wege von der zentralen Bindung zum am weitesten entfernten Atom liegt, 10% des maximalen Gewichtes erhält. Dazu wurde β so gewählt, dass $w(\frac{\delta_{max}}{2}) = 0.1$. δ_{max} bezeichnet hierbei die Länge des längsten kürzesten Weges. Unter der Bedingung, dass eine Ringtorsion so viel zählt wie $\frac{rSize}{2}$ azyklische Bindungen, wird das Gewicht

für einen Ring berechnet, indem die Summe der normalisierten Gewichte der Ringbindungen durch 2 geteilt wird.

4.1.3. TFD-Berechnung

Bevor der TFD zwischen zwei Konformationen berechnet werden kann, muss sichergestellt werden, dass es sich um Konformationen des selben Moleküls handelt. Hierzu werden unique SMILES (USMILES) [88] für beide Konformationen berechnet und miteinander verglichen. Nur wenn beide USMILES identisch sind, handelt es sich um Konformationen des selben Moleküls. Die USMILES werden ebenfalls benutzt, um die Atome beider Konformationen einander zuzuordnen und so sicher zu stellen, dass die richtigen Torsionswinkel und Ringtorsionen miteinander vergleichen werden.

Um den TFD zwischen zwei TFs zu berechnen, wird zuerst die absolute Differenz zwischen jedem Torsionswinkelpaar berechnet. Die Differenzen werden anschließend auf eine Zahl zwischen 0 (keine Abweichung) und 1 (maximale Abweichung) normalisiert, indem sie durch 180 Grad (die maximal mögliche Differenz) geteilt werden. In Fällen mit topologisch äquivalenten Atomen wird die maximale Differenz von 1 nicht erreicht. Wenn zum Beispiel die maximal mögliche Differenz 90 Grad ist, beträgt die maximal mögliche normalisierte Differenz 0.5. Zur Normalisierung von Ringtorsionen wird die absolute Differenz durch die maximal mögliche Differenz $maxDev$ (siehe Gleichung 4.1) geteilt. Die normalisierten Differenzen werden dann mit dem zugehörigen Gewicht w (siehe Gleichung 4.2) multipliziert. Um den endgültigen TFD zu erhalten, werden die gewichteten Differenzen aufsummiert und durch die Summe aller Gewichte geteilt.

TFD-Werte liegen im Bereich zwischen 0.0 und 1.0. Wenn zwei Konformationen miteinander verglichen werden und keiner ihrer Torsionswinkel und Ringtorsionen voneinander abweicht, dann beträgt der TFD 0.0. Wenn, im Gegensatz dazu, jeder Torsionswinkel und jede Ringtorsion maximale Abweichung aufweisen, beträgt der TFD 1.0. Ein TFD von 0.5 kann verschiedene Ursachen haben:

1. ein Torsionswinkel an einer zentralen Bindung weicht maximal um 180 Grad ab, oder

2. die Torsionswinkel zweier relativ zentraler Bindungen weichen jeweils um 90 Grad voneinander ab.

Ein TFD von 0.3 könnte darauf hinweisen, dass der Torsionswinkel einer terminalen Bindung maximal abweicht, oder dass mehrere Torsionswinkel und Ringtorsionen leicht voneinander abweichen.

4.2. Torsionsbibliothek

Die *Torsionsbibliothek* ist eine Sammlung von *Torsionssignaturen*, welche sich aus einem *Torsionsmuster*, einer Liste von häufig vorkommenden Torsionswinkeln und einem oder zwei *Torsionshistogrammen* zusammen setzen [31]. Im folgenden Abschnitt werden die einzelnen Teile der Torsionssignatur, die Hierarchie der Signaturen in der Torsionsbibliothek sowie die Generierung und Analyse von Torsionshistogrammen näher beschrieben.

4.2.1. Torsionssignatur

Eine Torsionssignatur besteht, wie bereits erwähnt, aus einem Torsionsmuster, einer Liste mit Torsionswinkeln und einem oder zwei Torsionshistogrammen. Das Torsionsmuster beschreibt die an einem Torsionswinkel beteiligten Atome und, je nach Spezifität, mehr oder weniger die weitere chemische Umgebung. Torsionsmuster werden in der SMARTS-Notation [89] angegeben, welche in der Chemieinformatik unter anderem in der Substruktursuche eingesetzt wird. Dabei wird nach einem bestimmten Muster (Subgraph, Substruktur), in einem Molekül (-graph) gesucht. Die SMARTS-Notation eignet sich sehr gut zur Beschreibung von Torsionsmustern, da sie einerseits in der Lage ist, bestimmte Substrukturen eines Moleküls exakt zu beschreiben und andererseits flexibel genug ist, um mehrere unterschiedliche Substrukturen zusammenzufassen. Ein Torsionsmuster muss mindestens die vier Atome, die den Torsionswinkel definieren (siehe Abbildung 2.1), enthalten. Die Atome werden dabei mit den Zahlen 1–4 markiert. Die zentrale rotierbare Bindung zwischen den Torsionsatomen 2 und 3 wird als eine beliebige azyklische Bindung (SMARTS `!@`) definiert. Die chemische Umgebung des Torsionswinkels kann näher spezifiziert werden, indem weitere mit den Torsionsatomen verbundene Substrukturen beschrieben werden. Abbildung 4.3 zeigt eine Torsionssignatur für ein Beispielmolekül.

Um noch flexibler bei der Beschreibung von Torsionswinkeln sein zu können, wird eine Erweiterung der SMARTS-Notation zur Beschreibung der Hybridorbitale von Atomen benutzt. Diese Erweiterung wird bereits von OpenEye [90] und Open Babel [91] benutzt. Die Hybridorbitale sp3, sp2 und

O=[S:1](=O)[NH0:2]!@[c:3]([cH1])[cH0:4]

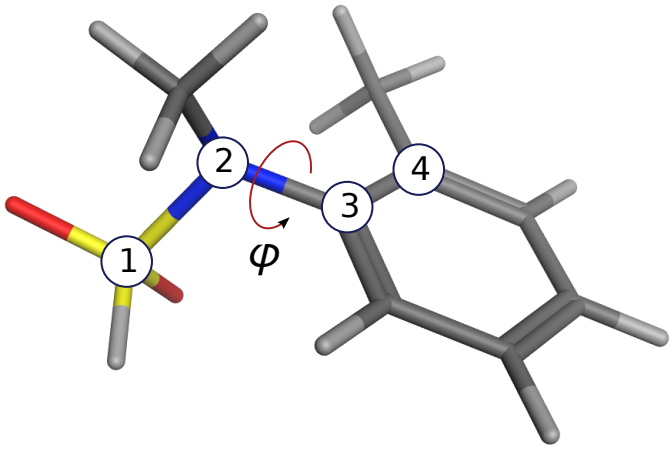


Abbildung 4.3.: Torsionssignatur in der SMARTS-Notation für ein Beispielmolekül. Die vier Torsionsatome sind in der Torsionssignatur mit den Zahlen 1–4 markiert.

sp1 werden dabei durch die SMARTS-Primitive `^3`, `^2` und `^1` definiert. Die SMARTS-Notation wurde zusätzlich noch um die Primitive `N_lp` zur Beschreibung von Torsionswinkeln, die das freie Elektronenpaar von Stickstoffatomen beinhalten, erweitert. Diese Erweiterung erlaubt die eindeutige Beschreibung von Torsionswinkeln mit sp3-Stickstoffatomen, wie sie zum Beispiel in Sulfonamiden vorkommen. Die Beschreibung dieser Torsionswinkel bildet eine Ausnahme von der Regel, dass ein Torsionsmuster mindestens die vier Torsionsatome beinhalten muss, da hier kein viertes Atom benötigt wird.

Das Torsionshistogramm zeigt die Verteilung der vorkommenden Werte des beschriebenen Torsionswinkels in einem gegebenen Datensatz. Eine Torsionssignatur kann zwei Torsionshistogramme beinhalten, um die Verteilung der Werte des Torsionswinkels in unterschiedlichen Datensätzen anzugeben. Zur Erstellung eines Torsionshistogramms werden die berechneten Torsionswinkelwerte zuerst in *Klassen (bins)* aufgeteilt. Dann wird für jede Klasse die absolute Anzahl der ihr zugeordneten Werte berechnet. Die automatische Generierung von Torsionshistogrammen wird in Abschnitt 4.2.5 näher beschrieben. Die Liste mit Torsionswinkeln spiegelt die im ersten oder zweiten Torsionshistogramm am häufigsten vorkommenden Werte (*Peaks*) wieder. Um auch die Breite der Peaks im Histogramm zu beschreiben, sind jedem Torsionswinkel in der Liste zwei Toleranzen zugeordnet, wobei die erste Toleranz immer enger als die zweite Toleranz ist. Die auto-

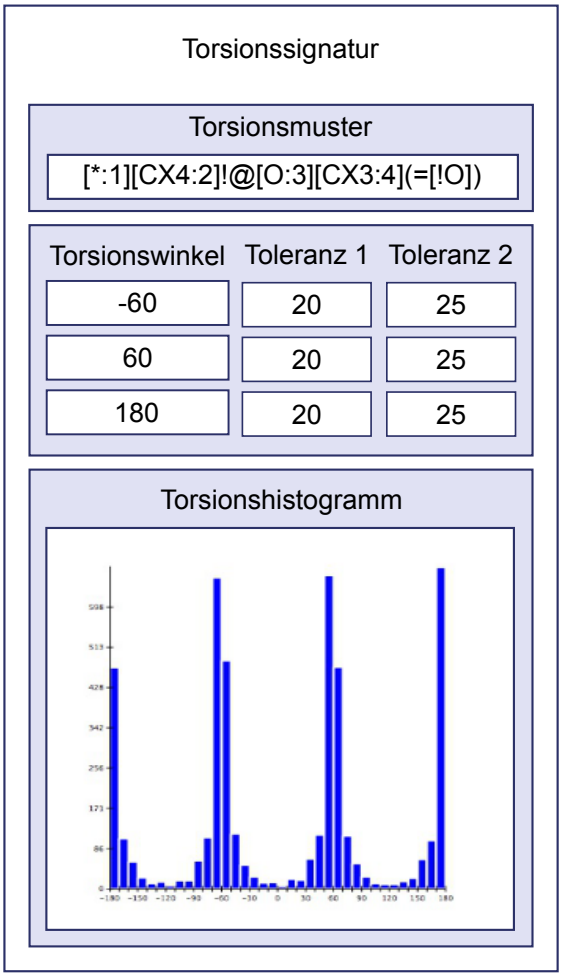


Abbildung 4.4.: Beispiel einer Torsionssignatur bestehend aus einem Torsionsmuster, einer Liste mit häufig vorkommenden Torsionswinkeln und einem Torsionshistogramm.

matische Bestimmung von Peaks und Toleranzen wird ebenfalls in Abschnitt 4.2.5 näher erläutert. Jedem Torsionswinkel kann zusätzlich noch ein *Score* zugeordnet werden, welcher später bei der Konformationsgenerierung benutzt werden kann. Die Berechnung und Benutzung des Scores wird in Abschnitt 4.3.4 näher beschrieben. Abbildung 4.4 zeigt ein Beispiel für eine Torsionssignatur.

4.2.2. Systematik von Torsionssignaturen

Alle Torsionssignaturen werden in der Torsionsbibliothek zusammengefasst und hierarchisch angeordnet. Auf der obersten Ebene werden die Torsionssignaturen anhand der Elemente der Torsionsatome 2 und 3 klassifiziert. Hierbei werden nur die Elemente C, N, O und S betrachtet. Jede der möglichen Kombinationen formt eine eigene Hierarchieklasse. Eine zusätzliche Klasse deckt alle generischen Versionen der vorherigen Klassen ab, so wie generische Beschreibungen anderer Heteroatom-Kombinationen. Diese generische Klasse wird mit GG bezeichnet, wobei jedes G eine generische Beschreibung der beiden zentralen Torsionsatome 2 und 3 repräsentiert. Die Torsionsatome 1 und 4 werden mit der SMARTS-Primitive $*$ beschrieben.

Jede der Hauptklassen kann weitere Subklassen, die typische funktionelle Gruppen oder häufig auftretende Substrukturen beschreiben, enthalten. Jede Subklasse kann wiederum weitere Subklassen enthalten. Innerhalb jeder Klasse oder Subklasse sind die Torsionsmuster nach abnehmender Spezifität angeordnet, wobei das spezifischste Muster ganz oben steht. Das hierarchische System wurde so strukturiert und bezeichnet, dass es einfach zu verstehen und zu modifizieren ist. Die Einteilung in Hauptklassen und Subklassen ermöglicht es außerdem, eine bestimmte Torsionssignatur schneller zu finden, da nur die entsprechende Hauptklasse, bzw. Subklasse durchsucht werden muss. In Abschnitt 4.2.4 wird die Suche nach einer Torsionssignatur genauer beschrieben. Abbildung 4.5 zeigt ein Beispiel zur hierarchischen Anordnung von Torsionssignaturen in der Torsionsbibliothek.

4.2.3. Definition der Torsionsbibliothek

Die Torsionsbibliothek kann beliebig viele Hauptklassen, Subklassen und Torsionssignaturen enthalten. Das Format, bzw. die Sprache in der die Torsionsbibliothek definiert wird, sollte daher möglichst flexibel sein und es erlauben, möglichst einfach neue Hauptklassen, Subklassen und Torsionssignaturen hinzuzufügen und bestehende Daten zu editieren oder zu löschen. Die *Extensible Markup Language* (XML) [92] ist ein einfaches und gleichzeitig sehr flexibles Textformat, welches sich sehr gut für die Beschreibung der Torsionsbibliothek eignet. Mit Hilfe eines XML-Schemas [93] lassen sich Struktur, Inhalt und Semantik von XML-Dateien definieren. Dies ermöglicht es, die hierarchische Struktur der Torsionsbibliothek exakt abzubilden. Abbildung 4.6 zeigt einen Auszug der Torsionsbibliothek. Das dazugehörige XML-Schema befindet sich in Anhang C.

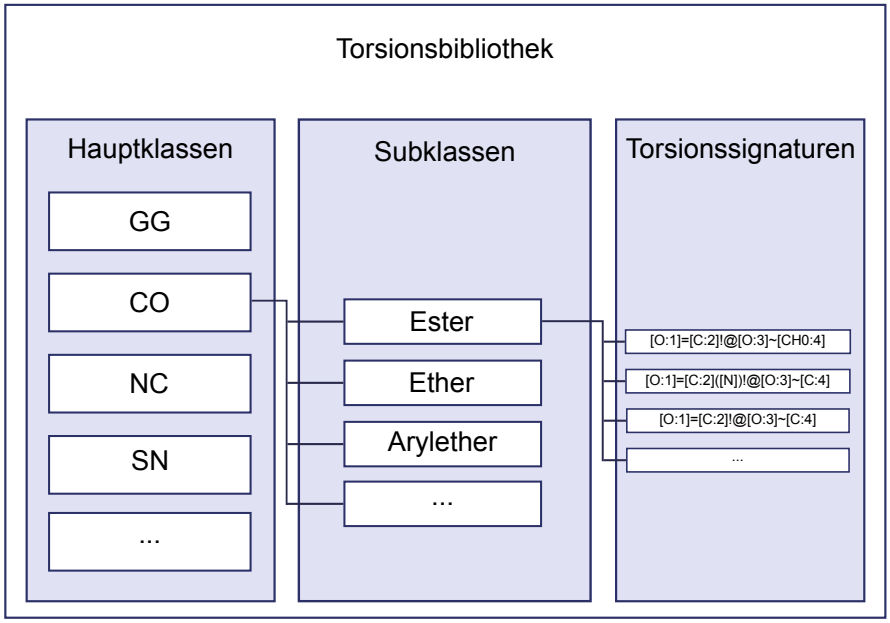


Abbildung 4.5.: Beispiel zur hierarchischen Anordnung von Torsionssignaturen in der Torsionsbibliothek.

4.2.4. Zuordnung einer Torsionssignatur

Die Zuordnung einer Torsionssignatur bzw. des Torsionsmusters zu einer rotierbaren Bindung wird an drei verschiedenen Stellen gebraucht: bei der Analyse von Molekülkonformationen, bei der Generierung von Molekülkonformationen (siehe Abschnitt 4.3) und bei der Generierung von Torsionshistogrammen (siehe Abschnitt 4.2.5). In allen drei Fällen erfolgt die Zuordnung mit Hilfe eines SMARTS-Matching-Algorithmus. Von den bereits publizierten Methoden zum SMARTS-Matching [94–98] wird hier eine modifizierte Version des VF2-Algorithmus [98] benutzt, da dieser als einer der schnellsten für die Anwendung auf Molekülen gilt [99].

Bei der Analyse von Konformationen wird jeder rotierbaren Bindung der Konformation eine passende Torsionssignatur zugeordnet. Anhand der einzelnen Daten der Torsionssignaturen lassen sich dann Rückschlüsse auf die Qualität der Konformation ziehen. Dazu wird der aktuelle Torsionswinkel der rotierbaren Bindung berechnet und mit Hilfe der Torsionswinkelliste als häufig, grenzwertig oder selten klassifiziert (*Ampel-Regel*):

- **häufig:** Der aktuell gemessene Torsionswinkel liegt innerhalb des ersten Toleranzbereiches eines der Winkel aus der Torsionswinkelliste. (Winkel aus der Liste +/- erste Toleranz)

```
<?xml version="1.0"?>
<library>
  ...
  <hierarchyClass name="CO" id1="C" id2="O">
    <hierarchySubClass name="Ester␣bond␣I" smarts="O=[C:2][O:3]">
      <torsionRule smarts="[O:1]=[C:2]!@[O:3]~[CHO:4]">
        <angleList>
          <angle value="0.00" tolerance1="20.00" tolerance2="20.00" score="0.00"/>
        </angleList>
        <histogram>
          <bin count="1"/>
         ...
          <bin count="1"/>
        </histogram>
        <histogram_shifted>
          <bin count="111"/>
          ...
          <bin count="2"/>
        </histogram_shifted>
        <histogram2>
          <bin count="27"/>
          ...
          <bin count="0"/>
        </histogram2>
        <histogram2_shifted>
          <bin count="14"/>
          ...
          <bin count="0"/>
        </histogram2_shifted>
      </torsionRule>
      ...
    </hierarchySubClass>
    ...
  </hierarchyClass>
  ...
</library>
```

Abbildung 4.6.: Auszug aus der in XML definierten Torsionsbibliothek. Die hierarchische Struktur der XML-Datei spiegelt die hierarchische Anordnung der Torsionssignaturen in der Torsionsbibliothek wieder.

- **grenzwertig:** Der aktuell gemessene Torsionswinkel liegt innerhalb des zweiten Toleranzbereiches eines der Winkel aus der Torsionswinkelliste. (Winkel aus der Liste +/- zweite Toleranz)

- **selten:** Der aktuell gemessene Torsionswinkel liegt komplett außerhalb der Toleranzbereiche der Winkel aus der Torsionswinkelliste.

Die Suche nach einer passenden Torsionssignatur innerhalb der Torsionsbibliothek wird mit Hilfe des folgenden Algorithmus durchgeführt:

1. Die Elemente des 2. und 3. Atoms der gerade betrachteten rotierbaren Bindung werden benutzt, um die richtige Hauptklasse zu identifizieren. Wird keine Hauptklasse gefunden, wird in der generischen GG-Klasse gesucht (Schritt 4).

2. Torsionssignaturen innerhalb einer Haupt- oder Subklasse werden von oben nach unten durchsucht, bis die erste Signatur mit einem passenden Muster gefunden wurde. Wenn keine passende Signatur gefunden wurde, wird nach einer passenden Subklasse gesucht (Schritt 3). Wenn auch keine passende Subklasse gefunden wurde, wird in der GG-Klasse weiter gesucht (Schritt 4).

3. Wenn eine passende Subklasse gefunden wurde, wird der iterative Suchprozess wie in dem vorherigen Schritt ausgeführt, bis eine Signatur mit einem passenden Muster gefunden wurde. Wurde keine passende Subklasse gefunden, wird in der GG-Klasse gesucht (Schritt 4).

4. In der GG-Klasse wird zuerst nach einer passenden Subklasse gesucht. Wird keine passende Subklasse gefunden, werden die Torsionssignaturen von oben nach unten durchsucht bis eine Signatur mit passendem Muster gefunden wurde.

Wenn die Zuordnung der vier Torsionsatome des Torsionsmusters zu den vier Atomen der betrachteten Bindung nicht eindeutig ist, zum Beispiel wenn das 1. oder 4. Atom als beliebig (*) gekennzeichnet sind, werden alle möglichen Zuordnungen in Betracht gezogen und die sich ergebenden Torsionswinkel berechnet. Als eindeutige Zuordnung wird dann diejenige gewählt, die einen Torsionswinkel hat, der einem Winkel aus der Torsionswinkelliste am nächsten liegt.

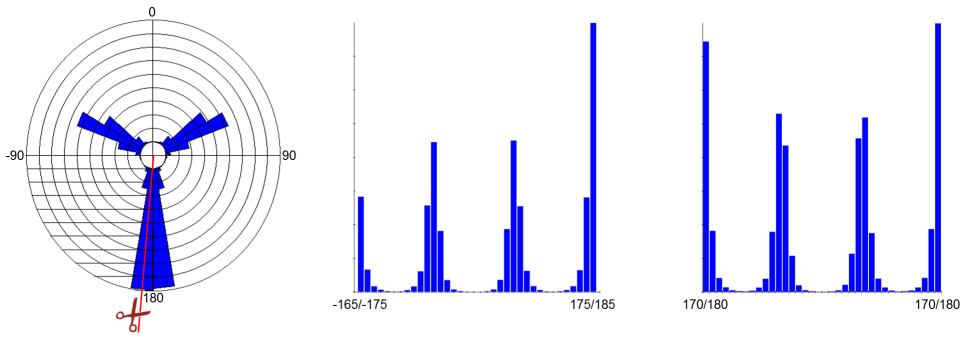


Abbildung 4.7.: Links: Zirkuläre Histogramme für Torsionswinkel können als Newman-Projektion interpretiert werden. Die rote Linie zeigt den Schnitt für die lineare Repräsentation an. Mitte: Histogramm welches für die automatische Winkelextrahierung benutzt wird. Das Histogramm erscheint asymmetrisch, da die Klassen um 5° versetzt sind. Rechts: Histogramm wie es im TorsionAnalyzer angezeigt wird.

4.2.5. Generierung und Analyse von Torsionshistogrammen

Die automatische Generierung eines Torsionshistogramms erfolgt, wie bereits erwähnt, mit Hilfe eines SMARTS-Matching-Algorithmus. Dabei wird für einen gegebenen Datensatz in allen Molekülen nach allen Vorkommen des gerade betrachteten Torsionsmusters gesucht und die gefundenen Torsionswinkel berechnet. Die Werte liegen dabei im Bereich $[-180°, +180°]$. Die berechneten Werte werden anschließend in Klassen (bins) mit einer Breite von 10° eingeteilt. Der überwiegende Teil der Torsionsmuster hat Peaks, welche mittig auf 0° oder vielfachen von 10° liegen. Um ausgeprägtere Peaks zu erhalten, welche anschließend leichter algorithmisch erkannt werden können, wurden die Klassen jeweils um 5° versetzt, so dass die Peaks mittig auf den Klassen liegen. Die zentrale Klasse liegt im Bereich $[-5°$, $+5°]$ und die beiden äußeren liegen im Bereich $[175°, 185°]$ bzw. $[-175°, -185°]$. Da Torsionswinkel mit einem absoluten Wert von größer als 180° nicht auftauchen, ist die Breite der beiden äußeren Klassen halbiert. Aus diesem Grund werden Torsionswinkel mit einem Wert zwischen $-180°$ und $-175°$ in die $[175°$, $185°]$-Klasse einsortiert. Dabei wird die zyklische Natur von Winkel-Histogrammen ausgenutzt. Die Umsortierung der Winkel ist gleichbedeutend mit dem Aufschneiden eines zirkulären Histogramms zwischen den Klassen $[-165°, -175°]$ und $[-175°, +175°]$ (siehe Abbildung 4.7). Torsionshistogramme mit Peaks bei 180° erscheinen durch dieses Verfahren asymmetrisch.

Peaks im Torsionshistogramm, welche die am häufigsten beobachteten Torsionswinkel darstellen, werden automatisch aus Histogrammen mit mehr als 100 gefundenen Torsionswinkeln (Datenpunkte) extrahiert. Um die extrahierten Winkel werden zwei Toleranzintervalle definiert, welche die Breite des Peaks repräsentieren. Die Bestimmung der Toleranzintervalle erfolgt ebenfalls automatisch für Histogramme mit mehr als 100 Datenpunkten. Zur Extraktion der Winkel und der Bestimmung der Toleranzintervalle wird der folgende Algorithmus benutzt:

1. Die absoluten Häufigkeiten werden in relative Häufigkeiten (Prozentwerte) umgewandelt.

2. Von jeder Klasse, die mehr als 4% der Datenpunkte des Histogramms enthält (Peak-Klasse), wird der zentrale Winkel extrahiert und in die Torsionswinkelliste der Torsionssignatur aufgenommen. Der zentrale Winkel einer Klasse ist dabei der Winkel, der genau in der Mitte des Intervalls liegt, welches die Klasse definiert. Zum Beispiel ist der zentrale Winkel der $[-5°, +5°]$-Klasse $0°$.

3. Zur Bestimmung des inneren Toleranzintervalls werden die ersten benachbarten Klassen zur linken und zur rechten Seite der im vorherigen Schritt bestimmten Peak-Klasse gesucht, welche weniger als 2,5% der Datenpunkte enthalten. Dann wird gezählt, wie viele Klassen zwischen diesen und der Peak-Klasse liegen. Unterscheidet sich die Anzahl der Klassen zur linken und zur rechten Seite, wird die kleinere Zahl zur Definition des Toleranzintervalls benutzt. Liegen zum Beispiel die ersten Klassen mit weniger als 2,5% der Datenpunkte drei Klassen links und vier Klassen rechts von der Peak-Klasse, wird das erste Toleranzintervall mit $30°$ definiert, da jede Klasse eine Breite von $10°$ hat.

4. Das zweite Toleranzintervall wird auf die gleiche Art berechnet, allerdings mit einem kleineren Cutoff von 1,5%.

Wenn die Daten im Histogramm gleichmäßig verteilt über alle Klassen sind (Varianz < 0.1), wird ein $30°$-Raster von $-180°$ bis $+180°$ benutzt, um einen Standardsatz von 12 Torsionswinkeln und Toleranzintervallen von $10°$ und $15°$ zu definieren. Diese Methode zur Ableitung von häufig auftretenden Torsionswinkeln aus Histogrammen ist ähnlich zu der Methode, die Sadowski und Boström benutzt haben, um automatisch Torsionsregeln aus Kristallstrukturen abzuleiten [100]. Die Autoren benutzen dabei Histogramme mit $30°$-Klassen, ein Minimum von 20 Datenpunkten pro Histogramm und eine Mindesthäufigkeit von 45% pro Klasse.

Wenn für ein Histogramm keine Daten gefunden werden, wird die entsprechende Torsionssignatur zwar in der Bibliothek behalten, aber deaktiviert. Für Histogramme mit 1 bis 100 Datenpunkten werden die Torsionswinkel und Toleranzintervalle manuell definiert. Lassen sich dabei eindeutige Peaks erkennen, werden diese als Torsionswinkel mit breiteren Toleranzintervallen zwischen 20° und 30° definiert. Wenn die Datenpunkte zu sehr verstreut sind, um eindeutige Peaks zu erkennen, wird entweder ein 30°-Raster benutzt (mit 10° und 15° für die Toleranzintervalle), oder die Torsionssignatur wird deaktiviert.

4.2.6. Abhängige Torsionswinkel

Eine Torsionssignatur kann als *abhängig* von einer anderen Torsionssignatur definiert werden. Die Abhängigkeit gilt allerdings nur, wenn beide Signaturen zwei aufeinander folgenden rotierbaren Bindungen in einem Molekül zugeordnet werden. Die abhängigen Torsionssignaturen werden jeweils um eine Liste mit *Abhängigkeitsregeln* erweitert. Jede Regel besteht aus einem Torsionswinkel für die erste Bindung (*TW*) und einer Liste von erlaubten Torsionswinkeln für die abhängige zweite Bindung.

4.3. Generierung von Konformationen

CONFECT (CONformations From an Expert Collection of Torsion patterns) [32] ist eine wissensbasierte Methode zur Generierung von Konformationen. Für ein gegebenes Molekül lässt sich die Konformationsgenerierung wie folgt beschreiben:

1. Das Molekül wird in einzelne Komponenten zerlegt und jeder rotierbaren Bindung wird eine Torsionssignatur aus der Torsionsbibliothek zugeordnet (siehe Abschnitt 4.2.4).

2. Konformationen werden erzeugt, indem ausgehend von einer Startkomponente und unter Berücksichtigung der Listen mit häufig vorkommenden Torsionswinkeln sukzessive alle Komponenten wieder hinzugefügt werden (*incremental construction* [101]). Ringkonformationen werden mit einem kombinierten Ansatz aus Ringtemplaten und kraftfeldbasierter Optimierung generiert.

3. Generierte Konformationen werden mit Hilfe eines Überlapptests auf überlappenden Atome untersucht und gegebenenfalls mit Hilfe eines chemischen Kraftfeldes optimiert.

4. Die Menge generierter Konformationen wird am Ende mit Hilfe eines Clustering-Verfahrens reduziert.

Verschiedene Qualitätsstufen werden eingesetzt, um die Abdeckung des Konformationsraumes eines Moleküls zu steuern. Im folgenden Abschnitt werden die einzelnen Schritte der Generierung von Konformationen näher beschrieben. Die Konformationsgenerierung lässt sich dabei in drei Phasen einteilen: *Vorverarbeitung*, *Aufbau* und *Nachverarbeitung*. Abbildung 4.8 zeigt den Prozess des inkrementellen Aufbaus exemplarisch für ein Beispielmolekül.

4.3.1. Komponentenbaum

Um sicher zu stellen, dass unabhängig von der Eingabestruktur das gleiche Molekül immer auf die gleiche Weise aufgeteilt und wieder zusammengesetzt wird, werden die Atome und Bindungen mit Hilfe von unique SMILES in eine eindeutige Reihenfolge gebracht. Anschließend wird das Molekül in kleinere Komponenten aufgeteilt. Die Teilung erfolgt dabei an jeder azyklischen, nicht terminalen Einfachbindung, welche nicht mit einer Methyl-, Trifluoromethyl oder Nitril-Gruppe verbunden ist. Dadurch entsteht eine Baumstruktur (*Komponentenbaum*), bei der die Knoten Teilmoleküle bzw. Komponenten und die Kanten die Bindungen zwischen den Komponenten repräsentieren. Die zentrale Komponente des Moleküls wird als *Wurzelknoten* definiert. Die Bestimmung der zentralen Komponente erfolgt ähnlich der Bestimmung der zentralen Bindung in der TFD-Berechnung (siehe Abschnitt 4.1) und startet mit der Berechnung der kürzesten Wege $\delta(k_1, k_2)$ entlang kovalenter Bindungen zwischen allen Paaren von Komponenten (k_1, k_2) mit Hilfe des Floyd-Warshall-Algorithmus [87]. Die zentrale Komponente k_{Wurzel} wird dann als die Komponente definiert, welche die kleinste Standardabweichung über alle kürzesten Wege hat. Um einen eindeutigen Komponentenbaum zu erhalten, werden Kindknoten anhand ihrer Bindung zum Elternknoten nach aufsteigender Reihenfolge der Bindungen (gemäß der unique SMILES, s.o.) sortiert. Die Reihenfolge, in der die Komponenten später angebaut werden, wird durch eine Breitensuche auf dem Komponentenbaum bestimmt. In der abstrakten Datenstruktur *Komponentenbaum* hat jede Komponente danach einen Zeiger auf ihre nächste Komponente, welche mit der Operation *GetNext()* abgerufen werden kann.

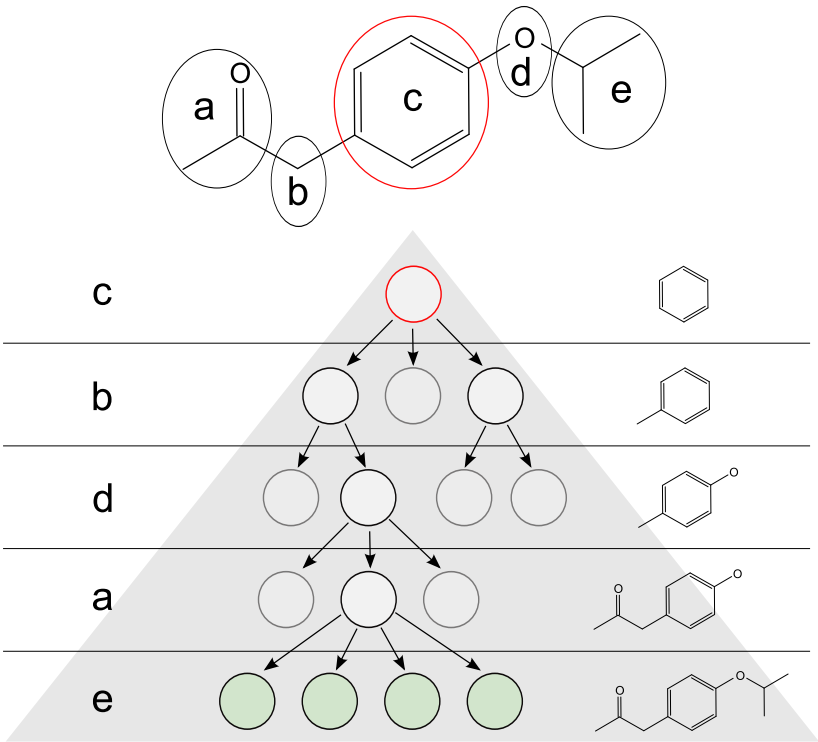


Abbildung 4.8.: Generierung von Konformationen für ein Beispielmolekül. Das Molekül wird in fünf Komponenten (a-e) aufgeteilt. Beginnend mit der zentralen Komponente c (roter Kreis), wird unter Berücksichtigung von vorher zugewiesenen Torsionswinkeln eine Komponente nach der anderen hinzugefügt. Die blass grau dargestellten Kreise repräsentieren Teilkonformationen, welche nicht mehr erweitert werden. Die grünen Kreise repräsentieren vollständig zusammengebaute Konformationen des Moleküls.

Die Kanten des Komponentenbaumes werden in der Datenstruktur abgebildet, indem jede Komponente einen Zeiger auf die Bindung zu ihrem Elternknoten erhält. Anschließend wird jeder dieser Bindungen eine Torsionssignatur aus der Torsionsbibliothek zugeordnet (siehe Abschnitt 4.2.4). Aufgrund der Hierarchie der Torsionsbibliothek ist die Zuordnung der Signaturen eindeutig. Aus der Torsionswinkelliste der zugeordneten Signatur werden die Torsionswinkel ohne Toleranzen extrahiert und als Liste in der jeweiligen Komponente gespeichert. Die Liste kann mit der Operation *GetAngleList()* abgerufen werden.

4.3.2. Ringkonformationen

Enthält das Eingabemolekül Ringe, werden für jedes Ringsystem, welches flexible Ringe mit bis zu neun Bindungen enthält, Ringkonformationen generiert. Konformationen für flexible Makrozyklen mit mehr als neun Bindungen werden nicht berechnet, da sie nicht relevant für den Wirkstoffentwurf sind. Hier wird die Ringkonformation des Eingabemoleküls behalten. Die Bestimmung der Ringe erfolgt nach dem Konzept der „unique ring families“ [102]. Die Methode zur Generierung von Ringkonformationen wurde von Benjamin Schulz im Rahmen eines Projektes am Zentrum für Bioinformatik Hamburg entwickelt und implementiert und wird hier nur kurz erläutert. Im ersten Schritt wird ausgehend vom Ring mit den meisten Verbindungen zu allen anderen Ringen eine Menge an initialen Grundgerüsten erstellt. Dazu werden solange Ring-Template aus einer Bibliothek zusammengebaut, bis eine vom Benutzer vorgegebene Anzahl an Ringkonformationen erreicht ist. Im zweiten Schritt werden die bisher generierten Grundgerüste anhand ihrer internen Energie sortiert, die mit Hilfe von vereinfachten Kraftfeldtermen berechnet wurde. Die Grundgerüste mit der niedrigsten Energie werden unter Berücksichtigung der direkten Substituenten des Rings weiter optimiert. Die Optimierung ist deterministisch und liefert so bei gleicher Eingabestruktur identische Konformationen.

Die Generierung der Ringkonformationen erfolgt noch vor der Aufbau-Phase. Für jede Komponente des Komponentenbaums wird ermittelt, ob die Komponente Ringsysteme mit flexiblen Ringen enthält. Ist dies der Fall, werden die Ringkonformationen generiert und als Liste in der Datenstruktur abgelegt. Die Liste kann dann mit der Operation *GetRingConfs()* abgerufen werden. Da die Generierung vor der Aufbau-Phase stattfindet, werden die Ringkonformationen nur ein Mal für jedes Ringsystem generiert und nicht jedes Mal, wenn die Komponente angebaut wird.

4.3.3. Qualitätsstufen

Um die Abdeckung des Konformationsraumes eines Moleküls zu steuern, gibt es acht verschiedene, feste Qualitätsstufen der Konformationsgenerierung. Die Stufen legen zum einen fest, wie viele Konformationen maximal für ein Molekül generiert werden sollen und wie viele Torsionswinkel pro rotierbarer Bindung eingestellt werden. In der ersten Qualitätsstufe werden pro rotierbarer Bindung nur die Peak-Winkel eingestellt, also die Torsionswinkel aus der Torsionssignatur, die aus den Torsionshistogrammen abgeleitet wurden. Die maximale Anzahl an Konformationen pro Molekül ist in der

Tabelle 4.2.: Überblick über die verschiedenen Qualitätsstufen in CONFECT. Die Spalte „Max Konf“ gibt die maximale Anzahl an zu erzeugenden Konformationen an. Die Spalte „Winkel“ gibt die Anzahl der benutzen Torsionswinkel pro Peak-Torsionswinkel, sowie die Einbeziehung der beiden Toleranzen (Tol1 und Tol2) an.

Stufe	Max Konf	Winkel					
1	250	1:	Peak				
2	500	3:	Peak	±Tol1			
21	500	3:	Peak		±Tol2		
3	1000	5:	Peak	±Tol1	±Tol2		
4	1000	5:	Peak	±Tol1		$\pm\frac{\text{Tol1}}{2}$	
41	1000	5:	Peak		±Tol2		$\pm(\text{Tol1} + \frac{\text{Tol2 - Tol1}}{2})$
5	2000	9:	Peak	±Tol1	±Tol2	$\pm\frac{\text{Tol1}}{2}$	$\pm(\text{Tol1} + \frac{\text{Tol2 - Tol1}}{2})$
6	1	1:	Peak				

ersten Qualitätsstufe auf 250 beschränkt. Die zweite Qualitätsstufe bezieht die erste Toleranz mit ein. Hier werden pro rotierbarer Bindung wieder jeweils die Peak-Winkel eingestellt und zusätzlich noch zwei weitere Winkel für jeden Peak-Winkel: Peak-Winkel + erste Toleranz und Peak-Winkel − erste Toleranz. Es werden also pro Winkel aus der Liste mit häufigen Torsionswinkeln einer Torsionssignatur drei Torsionswinkel eingestellt. Bei der sechsten Qualitätsstufe werden wieder nur die Peak-Winkel benutzt, allerdings wird nur eine einzige Konformation generiert. Die restlichen Qualitätsstufen nutzen entweder eine oder beide Toleranzen und jeweils einen zusätzlichen Winkel zwischen Peak und erster Toleranz und zwischen erster und zweiter Toleranz. Diese zusätzlichen Winkel werden nur benutzt, wenn der Abstand zwischen Peak und erster Toleranz, bzw. zwischen erster und zweiter Toleranz mehr als 20° beträgt. Eine genau Auflistung der Qualitätsstufen inklusive benutzter Torsionswinkel und maximaler Anzahl an Konformationen ist in Tabelle 4.2 angegeben.

Um die Flexibilität eines Moleküls beurteilen zu können, wird zuerst angenommen, dass für jede rotierbare Bindung nur die Peak-Winkel eingestellt werden. Die Anzahl der maximal möglichen Konformationen max_{Konf} wird berechnet, in dem für jede rotierbare Bindung die Anzahl der Torsionswinkel aus der zugeordneten Torsionssignatur und für jedes Ringsystem die Anzahl der generierten Ringkonformationen miteinander multipliziert werden. Ist $max_{Konf} > 100.000$, wird die Qualitätsstufe automatisch auf 1 gesetzt und für jede rotierbare Bindung, der eine Torsionswinkelliste mit

einem 30°-Raster zugeordnet wurde, wird die Liste auf ein 60°-Raster reduziert. Wenn $max_{Konf} > 1.000.000$ ist, werden die Listen mit 30°-Raster auf 120°-Raster reduziert. So wird sichergestellt, dass auch für sehr flexible Moleküle in angemessener Zeit sinnvolle Konformationen generiert werden können.

Wird eine andere Qualitätsstufe als 1 oder 6 ausgewählt, werden die Torsionswinkellisten der Komponenten um die entsprechenden Winkel erweitert.

4.3.4. Bewertungsfunktion

Die Steuerung der Aufbau-Phase und die spätere Sortierung der generierten Konformationen erfolgt mit Hilfe einer Bewertungsfunktion für Konformationen und Teilkonformationen. Für die Bewertungsfunktion gelten die folgenden Annahmen:

1. Bindungslängen und Bindungswinkel sind so eingestellt, dass sie in ihrem globalen Minimum liegen. Sie bleiben konstant und werden im Laufe des Algorithmus nicht weiter betrachtet.

2. Überlapptest und Optimierung sorgen dafür, dass die Konformationen keine überlappenden Atome haben. Ein van-der-Waals-Term ist daher nicht notwendig.

3. Elektrostatische Terme können auf Grund der unbekannten Interaktion mit dem Protein und der dadurch ebenfalls unbekannten Atomumgebung nicht berücksichtigt werden.

Zusammenfassend enthält die Bewertungsfunktion daher nur zwei Terme, von denen der Erste die Häufigkeit der Torsionswinkel und der zweite die Qualität der Ringkonformation abschätzt. Die Bewertungsfunktion hat nur positive Werte und muss maximiert werden.

Die Bewertung von Torsionswinkeln wird aus dem zugehörigen Torsionshistogramm abgeleitet und basiert auf den Annahmen, dass

1. Konformationen mit häufig vorkommenden Torsionswinkeln, also solchen, die auf oder nahe einem Peak im Torsionshistogramm liegen, eine niedrige Energie aufweisen, und dass

2. diese Energie steigt, je weiter der Winkel vom Peak entfernt ist.

Die absoluten Häufigkeiten des Torsionshistogramms werden in relative Häufigkeiten (Prozentwerte) umgerechnet. Jedem Peak-Torsionswinkel einer Signatur wird dann die relative Häufigkeit der passenden Histogrammklasse als *Score* zugewiesen. Die Bewertung der, durch die Qualitätsstufe festgelegten, zusätzlichen Torsionswinkel berechnet sich prozentual am Score des Peak-Torsionswinkels. So werden die folgenden Werte verwendet:

- Toleranz 1: 30% des Peak-Torsionswinkels
- Toleranz 2: 15% des Peak-Torsionswinkels
- Winkel zwischen Peak und Toleranz 1: 50% des Peak-Torsionswinkels
- Winkel zwischen Toleranz 1 und Toleranz 2: 20% des Peak-Torsionswinkels

Die Scores der Peak-Torsionswinkel werden in der Torsionsbibliothek gespeichert. Die gespeicherten Scores werden benutzt, falls das dazugehörige Torsionshistogramm nach Ableitung der Torsionswinkel und Scores aus der Torsionsbibliothek gelöscht wird.

Da für Ringkonformationen im Rahmen dieser Arbeit keine äquivalenten Häufigkeitsverteilungen erstellt wurden, werden diese wie folgt bewertet: der ersten Ringkonformation wird ein Score von 12 zugewiesen, was in etwa dem durchschnittlichen Score der am häufigsten vorkommenden Torsionswinkel entspricht. Jeder weiteren Ringkonformation wird die Hälfte vom Score der vorangegangenen Konformation zugewiesen.

Die Bewertung einer vollständig zusammengebauten Konformation erfolgt nun indem die Scores der eingestellten Torsionswinkel und Ringkonformationen aufsummiert werden. Eine Teilkonformation n wird mit folgender Funktion bewertet:

$$f(n) = g(n) + h(n) \tag{4.3}$$

wobei $g(n)$ ein berechneter Wert und $h(n)$ ein geschätzter Wert ist. $g(n)$ wird berechnet, in dem die Werte der eingestellten Torsionswinkel und Ringkonformationen der Teilkonformation aufsummiert werden. Der geschätzte Wert $h(n)$ wird berechnet, in dem für alle noch nicht eingestellten Winkel und Ringkonformationen angenommen wird, dass jeweils der Torsionswinkel bzw. die Ringkonformation mit dem höchsten Score eingestellt wird und diese dann aufsummiert werden. Die hier beschriebene Bewertungsfunktion $f(n)$ ist ähnlich denen in [10, 27, 103] beschriebenen Bewertungsfunktionen.

4.3.5. Aufbau der Konformationen

Konformationen für ein Molekül werden erzeugt, indem ausgehend von der zentralen Komponente und unter Berücksichtigung der Torsionswinkellisten eine Komponente nach der anderen hinzugefügt wird. Dabei entsteht ein *Konformationsbaum* (KB), bei dem innere Knoten teilweise zusammengebaute und Blattknoten vollständig zusammengebaute Konformationen repräsentieren. Jede Kante des KB steht für einen an einer rotierbaren Bindung eingestellten Torsionswinkel (siehe Abbildung 4.8). Wenn die Komponente K_i durch eine rotierbare Bindung mit der Komponente K_j verbunden ist, und K_j eine Torsionswinkelliste mit n Torsionswinkeln enthält, dann entstehen im KB n Kanten vom *Konformationsbaumknoten* (KBK) KBK_i zu n neuen KBK_{jk}, mit $1 \leq k \leq n$. Jeder KBK_{jk} repräsentiert dabei eine andere Teilkonformation des Moleküls. Enthält K_j zusätzlich noch ein Ringsystem mit m vorher generierten Ringkonformationen, dann entstehen im KB mn Kanten von KBK_i zu mn neuen KBK_{jk}, mit $1 \leq k \leq nm$. Ein neuer KBK wird mit der Operation *Init()* erzeugt. Wird der Init-Operation zusätzlich noch ein KBK übergeben, wird eine Kopie des übergebenen KBK erzeugt. Die Komponenten, aus denen eine Teilkonformation besteht, werden im entsprechenden KBK als Stack gespeichert, wobei die zuletzt angebaute Komponente oben auf dem Stack liegt. Die Komponenten eines KBK können mit der Operation *GetComponents()* abgerufen werden. Die Bewertung der Teilkonformation wird ebenfalls im KBK gespeichert (KBK.*Score*). Der Schritt, in dem eine neue Komponente an einen bereits vorhandenen KBK_i angebaut wird, wird auch *Erweiterung* von KBK_i genannt. Alle KBK, die noch erweitert werden können, werden in einer Prioritätswarteschlange (EQ) gespeichert, wobei die KBK darin nach absteigender Bewertung der Teilkonformationen geordnet sind. Die am besten bewertete Teilkonformation wird aus der EQ extrahiert und um die nächste Komponente erweitert. Die bei der Erweiterung neu entstandenen KBK werden mit Hilfe des Überlapptests aus Algorithmus 1 [104] auf überlappende Atome untersucht. Wenn es keine überlappenden Atome gibt oder die Überlappungen durch kleine Änderungen der eingestellten Torsionswinkel behoben werden könnten (Überlapp < 70), werden die neuen KBK in die EQ einsortiert, andernfalls ist die neu entstandene Teilkonformation nicht valide und wird verworfen. So wird sichergestellt, dass Konformationen mit stark überlappenden Atome bereits möglichst früh aussortiert werden, und nicht valide Teilkonformationen nicht unnötig vollständig zusammengebaut werden. Der Algorithmus zur Erweiterung eines KBK ist in Algorithmus 2 beschrieben. Wenn es für einen KBK keine weitere Komponente zum Anbauen gibt, dann ist die Konformation bereits vollständig zusammengebaut und der KBK wird in die ebenfalls

nach absteigender Bewertung sortierten Prioritätswarteschlange (KQ) für fertige Konformationen einsortiert. Es werden solange KBK erweitert, bis die maximale Anzahl an Konformationen erreicht ist und keine Konformation mehr generiert werden kann, die eine bessere Bewertung als die letzte Konformation in KQ hat, oder die EQ leer ist. Letzteres bedeutet, dass alle möglichen validen Konformationen bereits generiert wurden. Da jeweils immer die am besten bewertete Teilkonformation erweitert wird, werden zuerst die Konformationen erzeugt, bei denen die am häufigsten vorkommenden Torsionswinkel eingestellt wurden. So werden auch für sehr flexible Moleküle, für die nicht alle möglichen validen Konformationen erzeugt werden können, nur die Konformationen mit den statistisch am häufigsten vorkommenden Torsionswinkeln generiert. Es kann vorkommen, dass für ein Molekül nur sehr wenige valide Konformationen erzeugt werden können. Dies hat zur Folge, dass das Abbruchkriterium der maximal zu generierenden Konformationen nicht erreicht wird, und alle Möglichkeiten ausprobiert werden, bis die EQ leer ist. Da dies bei sehr flexiblen Molekülen zu einer sehr langen Laufzeit führen kann, wurde noch ein weiteres Abbruchkriterium eingeführt. Der Algorithmus bricht ebenfalls ab, wenn bereits 100.000 KBK erzeugt wurden. Algorithmus 3 beschreibt den Algorithmus zum Aufbau von Konformationen. Eine Übersicht der einzelnen Phasen der Konformationsgenerierung ist in Abbildung 4.9 dargestellt.

Wenn zwei aufeinander folgenden rotierbaren Bindungen ein abhängiges Torsionsmuster zugeordnet wurde, werden die Torsionswinkel der beiden Bindungen nicht anhand der Liste mit häufig vorkommenden Torsionswinkeln eingestellt, sondern anhand der Abhängigkeitsregeln (siehe 4.2.6). Für die erste Bindung können alle TW der Abhängigkeitsregeln eingestellt werden. Für die zweite Bindung wird, abhängig vom eingestellten TW der ersten Bindung, nach der passenden Abhängigkeitsregel gesucht und die Torsionswinkel werden anhand der Liste der erlaubten Torsionswinkel eingestellt.

4.3.6. TFD- und RMSD-Clustering

Zur Reduzierung der Menge der generierten Konformationen wird ein einfacher Cluster-Algorithmus (Algorithmus 4) angewendet, bei dem entweder TFD- (*TFD-Clustering*) oder RMSD-Werte (*RMSD-Clustering*) als Ähnlichkeitsmaß benutzt werden. Das Ergebnis des Clusterings ist dabei nicht eine Menge von Clustern, sondern jeweils eine repräsentative Konformation pro Cluster. Der Cluster-Algorithmus nutzt dabei die nach Bewertung sortierte KQ aus und startet mit der am besten bewerteten Konformation.

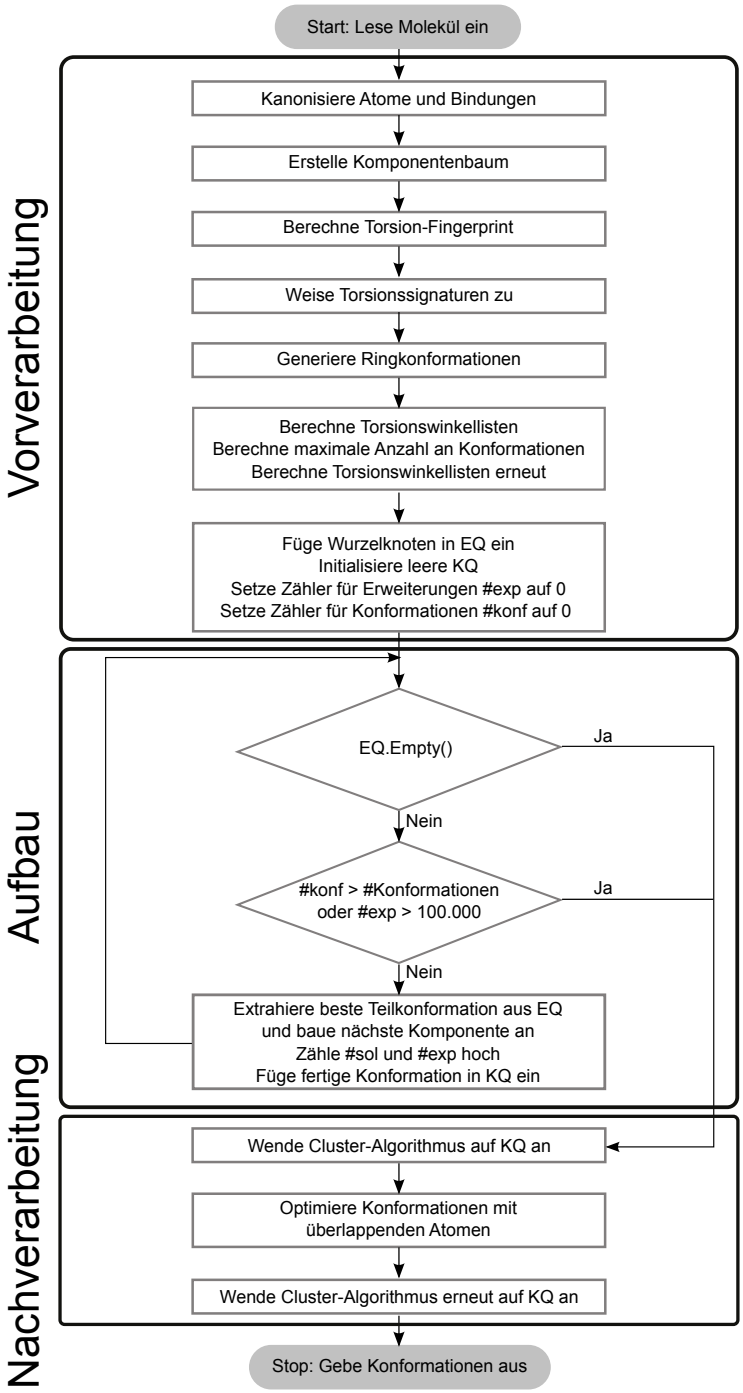


Abbildung 4.9.: Übersicht der einzelnen Phasen der Konformationsgenerierung. EQ: Prioritätswarteschlange mit erweiterbaren Konformationsbaumknoten. KQ: Prioritätswarteschlange mit vollständigen Konformationen

Algorithmus 1 Überlapptest

Input: Molekül m
Output: Überlapp

```
1:
2: Überlapp ← 0
3: for all Atompaare i, j von m do
4:     if i ist Wasserstoffatom or j ist Wasserstoffatom then
5:         continue
6:     else if i und j gehören zum selben Ringsystem then
7:         continue
8:     else if zwischen i und j liegen weniger als 3 Bindungen then
9:         continue
10:    else
11:        k_i ← Koordinaten von i
12:        k_j ← Koordinaten von j
13:        r_i ← VdW-Radius von i
14:        r_j ← VdW-Radius von j
15:        d1 = (Distanz(k_i, k_j))^2
16:        d2 = 0.5(r_i + r_j)^2
17:        if d1 > d2 then
18:            Überlapp = Überlapp + 10(d2/d1)
19:        end if
20:    end if
21: end for
22: return Überlapp
```

Von dieser Konformation K_i wird nacheinander der TFD- bzw. RMSD-Wert zu allen anderen Konformationen K_j ($i + 1 \leq j \leq n$, wobei n die Anzahl der Konformationen angibt) berechnet. Wenn der TFD- bzw. RMSD-Wert unter einem vorgegebenen Schwellenwert liegt, wird die Konformation K_j gelöscht. Die Konformation K_i wird als *Repräsentant* des Clusters behalten. Dann wird mit der am zweitbesten bewerteten, bisher nicht gelöschten Konformation weitergemacht und TFD- bzw. RMSD-Werte nacheinander zu allen noch verbliebenen Konformationen berechnet. Auch hier werden wieder die Konformationen gelöscht, bei denen der TFD- bzw. RMSD-Wert den vorgegebenen Schwellenwert unterschreitet. Durch die Nutzung der nach Bewertung sortierten KQ wird sichergestellt, dass immer jeweils die besser bewertete Konformation behalten und die schlechter bewertete Konformation gelöscht wird. Die Laufzeitkomplexität des TFD- bzw. RMSD-Clustering liegt in $O(n^2)$. Haben alle Konformationen einen unter dem Schwellenwert

Algorithmus 2 Erweiterung von Konformationsbaumknoten

Input: zu erweiternder KBK (ek), Prioritätswarteschlange mit zu erweiternden KBK (EQ), nächste anzubauende Komponente (nk)

```
 1:
 2: if not nk.GetAngleList() then
 3:     if nk.GetRingConfs() then    → starre Bindung, flexibles Ringsystem
 4:         for all nk.GetRingConfs() do
 5:             kbk.init(ek)    → neuer KBK mit Daten von ek
 6:             kbk.GetComponents().Push(nk)
 7:             Berechne neue Koordinaten und mache Überlapptest
 8:             if kein Überlapp or Überlapp optimierbar then
 9:                 kbk.Score ← Bewerte neue Teilkonformation
10:                 EQ.Enqueue(kbk)
11:             end if
12:         end for
13:     else    → starre Bindung, kein oder starres Ringsystem
14:         kbk.init(ek)
15:         kbk.GetComponents().Push(nk)
16:         Überlapptest
17:         if kein Überlapp or Überlapp optimierbar then
18:             kbk.Score ← Bewerte neue Teilkonformation
19:             EQ.Enqueue(kbk)
20:         end if
21:     end if
22: else
23:     if nk.GetRingConfs() then    → rotierbare Bindung, flexibles Ringsystem
24:         for all nk.GetRingConfs() do
25:             for all GetAngleList() do
26:                 kbk.init(ek)
27:                 kbk.GetComponents().Push(nk)
28:                 Berechne neue Koordinaten und mache Überlapptest
29:                 if kein Überlapp or Überlapp optimierbar then
30:                     kbk.Score ← Bewerte neue Teilkonformation
31:                     EQ.Enqueue(kbk)
32:                 end if
33:             end for
34:         end for
35:     else    → rotierbare Bindung, kein oder starres Ringsystem
36:         for all GetAngleList() do
37:             kbk.init(ek)
38:             kbk.GetComponents().Push(nk)
39:             Berechne neue Koordinaten und mache Überlapptest
40:             if kein Überlapp or Überlapp optimierbar then
41:                 kbk.Score ← Bewerte neue Teilkonformation
42:                 EQ.Enqueue(kbk)
43:             end if
44:         end for
45:     end if
46: end if
```

Algorithmus 3 Generierung von Konformationen

Input: Gewurzelter Komponentenbaum (*KompB*) eines Moleküls
Input: Maximale Anzahl an Konformationen (k)
Output: Eine Menge an Konformationen (KQ)

```
 1:
 2: KQ.Init()    → Prioritätswarteseschlange für fertige Konformationen
 3: EQ.Init()    → Prioritätswarteseschlange für zu erweiternde KBK
 4:
 5: kbk.Init()    → KBK-Wurzel
 6: kbk.Komponenten.Push(KompB-Wurzel)
 7: EQ.Enqueue(kbk)
 8:
 9: while not EQ.Empty() do
10:     if > 100.000 ek erzeugt then
11:         break
12:     end if
13:     if Anzahl Elemente in KQ ≥ k then
14:         for i = 1 to KQ.Size() - k do
15:             Lösche KQ.back()
16:         end for
17:         if KQ.back().Score > EQ.Front().Score then
18:             break
19:         end if
20:     end if
21:     ek ← EQ.Front()    → zu erweiternder KBK
22:     EQ.Dequeue()
23:     lk ← ek.GetComponents().Top()    → zuletzt angebaute Komponente
24:     nk ← lk.GetNext()    → nächste anzubauende Komponente
25:     if not nk then
26:         KQ.Enqueue(ek)    → fertige Konformation
27:
28:     else
29:         erweitereKnoten(ek, EQ, nk)
30:     end if
31: end while
32: return KQ
```

liegenden TFD- bzw RMSD-Wert zur ersten Konformation, benötigt der Algorithmus $n - 1$ TFD- bzw. RMSD-Berechnungen. Im schlechtesten Fall, wenn die TFD- bzw. RMSD-Werte zwischen allen Konformationen über dem Schwellenwert liegen, werden n^2 TFD- bzw. RMSD-Berechnungen benötigt. Der Algorithmus für das RMSD-Clustering funktioniert analog, wobei als Eingabe ein RMSD-Schwellenwert benutzt wird und in Zeile 19 nicht der TFD-Wert sondern der RMSD-Wert berechnet wird. Der hier beschriebene Cluster-Algorithmus ist eine modifizierte Version des von Kayser in seiner Diplomarbeit beschriebenen *Iterative in-place clustering* [103]. Hierin vergleicht Kayser das iterative Clustering mit dem hierarchisch agglomerativen Complete-Linkage-Verfahren und kommt zu dem Ergebnis, dass das iterative Clustering dem Complete-Linkage-Clustering vorzuziehen ist, das iterative Clustering aber die Nachteile hat, dass die Anzahl der Cluster nicht kontrolliert werden kann und dass die Sortierung der Konformationen während des Clusterings verloren geht. Die hier beschriebene modifizierte Version behält zwar die Reihenfolge der Konformationen während des Clusterings bei, braucht dafür aber zwei zusätzliche Iterationen über die Konformationen, eine am Anfang, in dem die Queue in ein Array umgewandelt wird, und eine am Ende, in dem das Array wieder in die Queue überführt wird.

Um das TFD-Clustering zu beschleunigen, wird der TF für jede Teilkonformation während des Aufbaus angepasst. Dazu wird am Anfang der TF für das Eingabe-Molekül berechnet und dann wird bei jedem Anbau-Schritt nur der TF-Wert für den entsprechenden neuen Torsionswinkel bzw die neue Ringkonformation neu berechnet und in dem, vom Elternknoten kopierten, TF ausgetauscht.

4.3.7. Optimierung

Aus der nach dem TFD- bzw. RMSD-Clustering übrig gebliebenen Menge an Konformationen werden die Konformationen extrahiert, bei denen nicht kovalent gebundene Atome überlappen, die Überlappungen aber durch Anpassung der eingestellten Torsionswinkel innerhalb gegebener Toleranzen behoben werden können. Diese Konformationen werden anschließend mit Hilfe eines chemischen Kraftfeldes [105] optimiert. Die Optimierung wurde dabei so angepasst, dass nur die Torsionswinkel neu eingestellt werden und bei der Einstellung die, aus der passenden Torsionssignatur abgeleitete, zweite Toleranz nicht überschritten wird. Konformationen, deren Atom-Überlappungen mit Hilfe der Optimierung nicht behoben werden konnten, werden verworfen. Die andern Konformationen werden wieder in die Menge

Algorithmus 4 TFD-Clustering

Input: Prioritätswarteschlange (Q) mit n Konformationsbaumknoten, wobei $n > 2$
Input: Schwellenwert (sw) für TFD-Werte
Output: Q nach Clustering

```
1:
2: kbk ← erstelle ein neues Array der Größe n.
3: zuLöschen ← erstelle ein neues Array der Größe n
4:
5: for i = 0 to n − 1 do
6:     kbk[i] ← Q.Front()
7:     Q.Dequeue()
8:     zuLöschen[i] ← false
9: end for
10:
11: for i = 0 to n − 2 do
12:     if zuLöschen[i] then
13:         continue
14:     end if
15:     for j = i + 1 to n − 1  do
16:         if zuLöschen[j] then
17:             continue
18:         end if
19:         tfd ← Berechne TFD zwischen kbk[i] und kbk[j]
20:         if tfd ≤ sw then
21:             zuLöschen[j] ← true
22:         end if
23:     end for
24: end for
25:
26: for i = 0 to n − 1 do
27:     if not zuLöschen[i] then
28:         Q.Enqueue(kbk[i])
29:     end if
30: end for
31: return Q
```

der generierten Konformationen einsortiert. Da sich bei der Optimierung die Koordinaten der Konformationen leicht verändert haben, erfolgt zum Abschluss ein erneutes TFD- bzw. RMSD-Clustering.

4.4. TorsionAnalyzer

Der *TorsionAnalyzer* ist ein graphisches Softwarewerkzeug zur Anzeige, Erstellung und Bearbeitung von Torsionsbibliotheken, zur automatischen Generierung und Analyse von Torsionshistogrammen und zur Analyse von Molekülkonformationen. Ziel ist die Unterstützung bei der Analyse von Molekülkonformation. Im folgenden Abschnitt wird der Aufbau und die Benutzung, sowie die einzelnen Komponenten des TorsionAnalyzers näher beschrieben. Abbildung 4.10 zeigt einen Screenshot des TorsionAnalyzers.

4.4.1. Benutzungsschnittstelle (UI)

Beim Starten des TorsionAnalyzers öffnet sich das Hauptfenster, welches in 3 Teilbereiche untergliedert ist (siehe Abbildung 4.10). Die *Bibliotheks-Ansicht* (Abb. 4.10(b)) zeigt eine Standard-Torsionsbibliothek, welche bereits automatisch beim Starten des TorsionAnalyzers geladen wird. Durch Klick auf eine der Torsionssignaturen in der Bibliotheks-Ansicht werden die vollständigen Daten der Signatur (Torsionsmuster, Liste mit Torsionswinkeln und Toleranzen, Torsionshistogramme) in der *Signatur-Ansicht* angezeigt (Abb. 4.10(c)). Über das *File*-Menü können jetzt mit *Load Molecule(s)...* ein oder mehrere Moleküle aus einer Datei im Tripos MOL2-Format [107] oder im Accelrys SDF-Format (früher Symyx oder MDL) [108] geladen werden. Die dreidimensionale Struktur des Moleküls wird dann in der *Molekül-Ansicht* (Abb. 4.10(a)) angezeigt. Werden mehrere Moleküle geladen, wird zunächst das erste Molekül aus der Datei in der Molekül-Ansicht angezeigt. Mit dem Schieberegler oberhalb der Molekül-Ansicht lässt sich dann zwischen den anderen Molekülen wechseln. Über das Menü *Torsion Library* → *Load Torsion Library...* kann eine andere Torsionsbibliothek geladen werden. Die weiteren Menüpunkte des Hauptfensters werden in den folgenden Abschnitten erläutert.

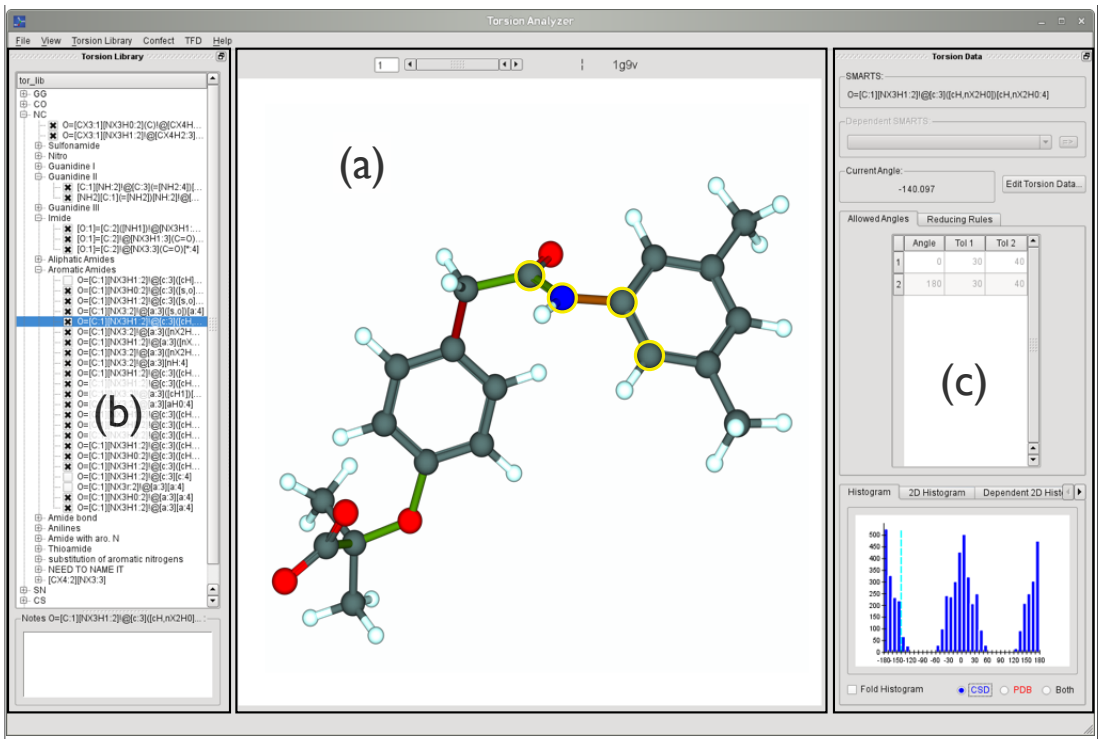


Abbildung 4.10.: Screenshot des TorsionAnalyzer-Hauptfensters. (a) Molekül-Ansicht, (b) Bibliotheks-Ansicht, (c) Signatur-Ansicht. Als Beispiel ist hier der Ligand aus dem PDB-Komplex 1g9v [106] geladen. Die Molekül-Ansicht (a) zeigt die dreidimensionale Struktur des Liganden. Die Bindungen sind nach der Ampel-Regel (häufig, grenzwertig, selten, siehe Abschnitt 4.2.4) eingefärbt. In dem Beispiel ist die orangefarbene Bindung ausgewählt und die vier an dem Torsionswinkel beteiligten Atome sind durch einen gelben Kreis hervorgehoben. In der Bibliotheksansicht (b) ist das Torsionsmuster, welches der ausgewählten Bindung zugeordnet wurde, durch einen blauen Balken hervorgehoben. Die Signatur-Ansicht (c) zeigt die detaillierten Daten der der ausgewählten Bindung zugeordneten Torsionssignatur. Der aktuell berechnete Torsionswinkel wird im Feld *Current Angle* angezeigt und als hellblau gestrichelte Linie im Torsionshistogramm dargestellt.

4.4.2. Arbeiten mit der Torsionsbibliothek

Mit dem TorsionAnalyzer lassen sich die in XML definierten Torsionsbibliotheken ansehen, erstellen, bearbeiten und abspeichern. In der Bibliotheks-Ansicht wird die Torsionsbibliothek in einer Baumstruktur dargestellt. Dies spiegelt den hierarchischen Aufbau der Bibliothek wider und erleichtert dem Benutzer das Finden und Editieren von Torsionssignaturen. Beim Start des TorsionAnalyzers werden zunächst die Hauptklassen der Torsionsbibliothek angezeigt. Diese lassen sich weiter aufklappen, und zeigen dann die darunter liegenden Subklassen und Signaturen. Von den Torsionssignaturen wird in der Bibliotheks-Ansicht nur das Torsionsmuster angezeigt. Alle weiteren Daten (Liste mit häufig vorkommenden Torsionswinkeln und Torsionshistogramme) werden in der Signatur-Ansicht dargestellt. Hauptklassen, Subklassen und Signaturen lassen sich durch einen Doppelklick editieren. Bei den Hauptklassen öffnet sich ein weiteres Dialogfenster, in dem die beiden Elemente, welche die Hauptklasse definieren, über ein Drop-Down-Menü ausgewählt werden können. Das Dialogfenster zum Editieren von Subklassen enthält ein Eingabefeld für einen optionalen Namen und ein Eingabefeld für das SMARTS-Muster der Klasse. Der SMARTS wird automatisch während der Eingabe validiert, wobei sich das Eingabefeld bei einem nicht validen SMARTS rot färbt. Der Benutzer kann also sofort erkennen, ob der SMARTS valide ist und die Eingabe notfalls korrigieren. Abbildung 4.11 zeigt das Dialogfenster zum Editieren von Torsionssignaturen. Dieses Dialogfenster enthält ebenfalls ein Eingabefeld für das Torsionsmuster (Abb. 4.11(a)) welches den SMARTS während der Eingabe automatisch validiert. Des weiteren enthält das Dialogfenster ein SMARTSViewer-Bild [109] des Torsionsmusters (Abb. 4.11(c)). Das SMARTSViewer-Bild zeigt die 2D-Visualisierung von SMARTS-Mustern und ist damit eine graphische Unterstützung bei der Erstellung von Torsionsmustern. Das Bild wird ebenfalls automatisch während der Eingabe des Torsionsmusters aktualisiert. Eine Tabelle erlaubt es, die Liste mit häufig vorkommenden Torsionswinkeln und Toleranzen zu bearbeiten (Abb. 4.11(b)). Das Dialogfenster zur Bearbeitung von Torsionssignaturen lässt sich ebenfalls über den Button *Edit Torsion Data* in der Signatur-Ansicht aufrufen. Alle drei Editier-Dialogfenster werden auch zum Hinzufügen neuer Hauptklassen, Subklassen oder Torsionssignaturen benutzt. Die Bibliotheksansicht enthält vier verschiedene Kontextmenüs, jeweils eins für die gesamte Torsionsbibliothek, Hauptklassen, Subklassen und Torsionssignaturen. Die Kontextmenüs beinhalten Menüpunkte, um die bereits erwähnten Editier-Dialogfenster aufzurufen, Menüpunkte zum Laden und Speichern von Torsionsbibliotheken, sowie Menüpunkte zum Löschen von Hauptklassen, Subklassen und Torsionssignaturen. Mit den

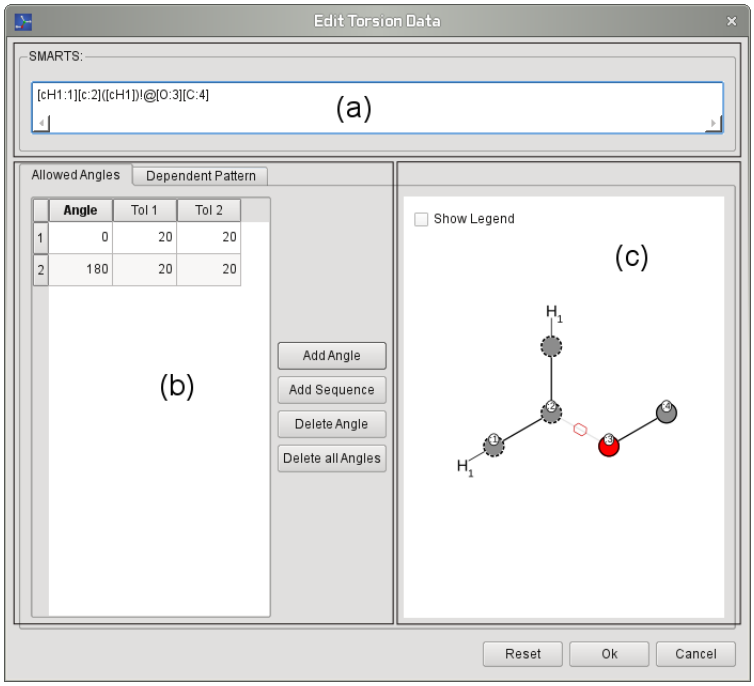


Abbildung 4.11.: Dialogfenster zum Editieren von Torsionssignaturen.(a) Eingabefeld für das Torsionsmuster, (b) Tabelle für häufige Torsionswinkel inklusive Toleranzen, (c) 2D-Visualisierung des Torsionsmusters

Tasten `u` (up) und `d` (down) lassen sich Torsionssignaturen innerhalb ihrer Klassen nach oben oder unten verschieben, um sie entsprechend ihrer Spezifität in die Hierarchie einzufügen.

Wenn eine Torsionsbibliothek verändert wird, oder wenn eine komplett neue Bibliothek angelegt wird, kann es hilfreich sein, nach fehlenden Torsionsmustern zu suchen. Ein fehlendes Torsionsmuster bedeutet in diesem Zusammenhang, dass für eine oder mehrere rotierbare Bindungen einer Molekülkonformation keine passende Torsionssignatur in der Torsionsbibliothek gefunden wird. Um nach fehlenden Torsionsmustern zu suchen, werden zuerst mehrere Moleküle in den TorsionAnalyzer geladen. Über den Menüpunkt *Torsion Library* → *Check Molecules for missing Patterns...* wird dann die Suche nach fehlenden Mustern gestartet. Dabei wird in allen Molekülen für jede rotierbare Bindung nach einer passenden Torsionssignatur gesucht (siehe Abschnitt 4.2.4). Am Ende der Suche öffnet sich ein Dialogfenster mit der Anzahl der Moleküle mit fehlenden Torsionsmustern. Mit einem Klick auf *Show Details...* werden die Namen der Moleküle sowie die genaue Anzahl der rotierbaren Bindungen, denen keine Torsionssignatur zugeordnet werden konnte, angezeigt. Bei der Anzeige eines der gefun-

denen Moleküle in der Molekül-Ansicht sind die Bindungen, denen keine Torsionssignatur zugeordnet werden konnte, gelb eingefärbt. Der Datensatz, in dem nach fehlenden Mustern gesucht wird, sollte entsprechend divers sein.

Eine weitere nützliche Information ist, wie häufig die einzelnen Torsionssignaturen in einem gegebenen Moleküldatensatz vorkommen, also welche Torsionssignatur wie oft einer rotierbaren Bindung zugeordnet wird. So können zum Beispiel Signaturen identifiziert werden, die entweder nie, oder besonders häufig einer rotierbaren Bindung zugeordnet werden. Bei besonders häufig genutzten Torsionsmustern stellt sich die Frage, ob diese gegebenenfalls in spezifischere Muster unterteilt werden könnten. Durch Auswahl des Menüpunktes *Torsion Library* → *Calculate Pattern Usage...* öffnet sich ein Dialogfenster zur Auswahl des Datensatzes. Die Moleküle können dabei im MOL2- oder SDF-Format vorliegen. Nach Auswahl des Datensatzes startet die Häufigkeitsberechnung automatisch. Hierzu wird wieder jeder rotierbaren Bindung jedes Moleküls eine Torsionssignatur zugeordnet und für jede Signatur wird gezählt, wie oft sie zugeordnet wurde. Am Ende der Berechnung öffnet sich ein weiteres Dialogfenster mit einer Ergebnis-Tabelle, in der alle Torsionsmuster mit ihrer entsprechenden Häufigkeit eingetragen sind. Die Tabelle lässt sich entweder alphabetisch nach Torsionsmuster, oder nach der Häufigkeit sortieren. Die Ergebnisse können im Comma-Separated-Values-Format (CSV) abgespeichert werden. Das CSV-Format ist ein weit verbreitetes Dateiformat, welches häufig benutzt wird, um Tabellendaten zwischen verschiedenen Programmen auszutauschen. So können die Ergebnisse zum Beispiel in einem Tabellenkalkulationsprogramm wie Microsoft Excel geöffnet und weiter bearbeitet werden.

4.4.3. Arbeiten mit Molekülkonformationen

Die Hauptfunktion des TorsionAnalyzers ist, wie der Name schon vermuten lässt, die Analyse von Molekülkonformationen. Hierzu werden eine Torsionsbibliothek und eine oder mehrere Konformationen in den TorsionAnalyzer geladen. Beim Laden der Konformationen wird sofort für alle rotierbaren Bindungen der ersten Konformation nach einer passenden Torsionssignatur in der aktuell geladenen Torsionsbibliothek gesucht. Die rotierbaren Bindungen werden anschließend nach der Ampel-Regel (häufig, grenzwertig, selten; siehe Abschnitt 4.2.4) eingefärbt. Der Benutzer kann so auf den ersten Blick erkennen, ob eine der Konformationen seltene Torsionswinkel enthält. Die einer rotierbaren Bindung zugeordnete Torsionssignatur wird durch Anklicken der rotierbaren Bindung in der Bibliotheks-Ansicht hervorgehoben

und die vollständigen Daten der Signatur werden in der Signatur-Ansicht angezeigt. Die vier an dem Torsionswinkel beteiligten Atome werden durch einen gelben Kreis hervorgehoben und der aktuelle Torsionswinkel wird berechnet und im Feld *Current Angle* angezeigt. Der berechnete Torsionswinkel wird zusätzlich als hellblau gestrichelte Linie im Torsionshistogramm dargestellt (siehe auch Abbildung 4.10). Diese Linie unterstützt eine schnelle Abschätzung, wie weit der aktuell berechnete Torsionswinkel von einem Peak im Histogramm entfernt ist. Um zu untersuchen, ob einer rotierbaren Bindung auch eine Torsionssignatur mit weniger spezifisch definiertem Torsionsmuster zugeordnet werden kann, können die Torsionssignaturen über eine Checkbox neben dem Torsionsmuster in der Bibliotheks-Ansicht deaktiviert werden. Sobald eine Signatur deaktiviert wurde, wird allen rotierbaren Bindungen erneut eine (aktive) Torsionssignatur zugeordnet. Wird für eine Bindung keine Torsionssignatur mehr gefunden, wird die Bindung gelb eingefärbt. Wenn mehrere Konformationen in den TorsionAnalyzer geladen werden, werden die passenden Torsionssignaturen gesucht und die rotierbaren Bindungen eingefärbt, sobald mit Hilfe des Schiebereglers zur nächsten Konformation gewechselt wird.

Ein weiterer Punkt bei der Analyse von Molekülkonformationen ist der Vergleich von mehreren Konformationen eines Moleküls. Der TorsionAnalyzer bietet Funktionen zur Berechnung von TFD (siehe Abschnitt 4.1) und RMSD (siehe Abschnitt 3.1.1). Über den Menüpunkt *TFD* → *Show Fingerprint* wird der TF des aktuell angezeigten Moleküls berechnet und angezeigt. Die Berechnung des TFD erfolgt über den Menüpunkt *TFD* → *Calculate TFD...*. Nach Auswahl des Menüpunktes öffnet sich ein Dialogfenster, welches jeweils zwei Möglichkeiten zur Auswahl der Referenzkonformation und der Anfragekonformation bietet. Als Referenzkonformation(en) lassen sich entweder die aktuell angezeigte Konformation, oder Konformationen aus einer externen Datei wählen. Für die Anfragekonformation(en) lassen sich entweder die aktuell geladenen Konformationen oder ebenfalls Konformationen aus einer externen Datei auswählen. Die externen Konformationen können im MOL2- oder SDF-Format vorliegen. Nach der Auswahl von Referenz- und Anfragekonformationen wird die TFD-Berechnung über den Button *Calculate TFD* gestartet. Enthalten die Dateien der Referenz- und Anfrage-Konformationen mehrere unterschiedliche Moleküle, erfolgt die Zuordnung von Anfrage- zu Referenzkonformation mit Hilfe ihrer USMILES. Am Ende der TFD-Berechnung öffnet sich ein Dialogfenster mit den Ergebnissen. In der unteren Tabelle des Fensters werden ID, Name und TF der Referenzkonformation(en) angezeigt. Die obere Tabelle listet ID, Name, TF und berechneten TFD der Anfragekonformation(en) auf. Die Tabelle lässt sich nach allen vier Feldern sortieren. Auch hier können die Ergebnisse als

TFD

	Molecule ID	Molecule Name	Torsion Fingerprint	TFD
1	1	ABADOX	(285.73 94.56 1.38 0.42)	0
2	2	ABADOX_1	(90.00 288.33 1.37 0.42)	0.227814
3	3	ABADOX_2	(90.00 168.33 1.38 0.42)	0.481464
4	4	ABADOX_3	(90.00 48.33 1.38 0.42)	0.401682
5	5	ABAKUK	(63.63 12.60 57.64 62.02 62.06 354.96 118.89 58.74 70.76 60.98 22.35)	0
6	6	ABAKUK_1	(360.00 56.42 360.00 119.54 60.00 60.00 60.00 180.00 60.00 60.00 22.34)	0.230879
7	7	ABAKUK_2	(360.00 56.42 360.00 123.80 60.00 60.00 60.00 180.00 60.00 59.99 25.37)	0.25366
8	8	ABAKUK_3	(360.00 56.42 360.00 119.14 60.00 60.00 60.00 180.00 60.01 60.00 17.25)	0.269309
9	9	ABAKUK_4	(360.00 56.42 360.00 116.00 60.00 60.00 60.00 180.00 60.00 60.00 5.76)	0.355925
10	10	ABAWEG01	(269.06 177.64 56.15 180.00 90.94 359.13 4.14)	0
11	11	ABAWEG01_1	(0.00 180.00 360.00 180.00 60.00 119.72 180.00)	0.145309
12	12	ABAWEG01_2	(0.00 180.00 360.00 280.00 60.00 119.72 180.00)	0.462351
13	13	ABAWEG01_3	(0.00 280.00 360.00 180.00 60.00 119.72 180.00)	0.150588
14	14	ABAWEG01_4	(0.00 180.00 0.00 80.00 60.00 119.72 180.00)	0.462419

	Molecule ID	Molecule Name	Torsion Fingerprint
1	1	ABADOX	(285.73 94.56 1.38 0.42)
2	2	ABAKUK	(63.63 12.60 57.64 62.02 62.06 354.96 118.89 58.74 70.76 60.98 22.35)
3	3	ABAWEG01	(269.06 177.64 56.15 180.00 90.94 359.13 4.14)
4	4	ABAXES	(86.47 38.14 63.26 352.12 101.39 28.43 51.74 52.13 53.41)
5	5	ABEBAK	(359.90 2.80 336.58 119.25 183.10 60.00 332.63 352.12 60.00 1.47 13.71 1.42 3.24 1.26)
6	6	ABEBEO	(206.46 183.65 63.66 307.43 57.03 66.54 0.97 26.78 4.63 0.66)
7	7	ABEFAP	(171.91 184.35 172.19 354.10 59.42 55.90 58.76 183.17 148.89 1.10 0.95)

Save Results Close

Abbildung 4.12.: Dialogfenster zur Anzeige der Ergebnisse aus der TFD-Berechnung. Obere Tabelle: ID, Name, TF und TFD der Anfrage-Konformationen. Untere Tabelle: ID, Name und TF der Referenz-Konformationen.

CSV-Datei gespeichert werden. Die RMSD-Berechnung erfolgt analog zur TFD-Berechnung und lässt sich über den Menüpunkt *Confect* → *Calculate RMSD...* aufrufen. Abbildung 4.12 zeigt beispielhaft das Dialogfenster für die Ergebnisse aus der TFD-Berechnung.

Der TorsionAnalyzer bietet nicht nur die Möglichkeit zur Analyse von Molekülkonformationen, sondern auch die Möglichkeit, diese zu generieren (siehe Abschnitt 4.3). Durch Auswahl des Menüpunktes *Confect* → *Generate*

Conformations... öffnet sich ein Dialogfenster, in dem die folgenden Einstellungen gemacht werden können:

- Auswahl der Qualitätsstufe (siehe Abschnitt 4.3.3)
- anschalten der Optimierung (siehe Abschnitt 4.3.7)
- anschalten der Generierung von Ringkonformationen (siehe Abschnitt 4.3.2)
- anschließende Anzeige der generierten Konformationen im Torsion-Analyzer
- optionale Angabe der Anzahl der maximal zu generierenden Konformationen
- Auswahl des Clustering-Verfahrens (TFD oder RMSD, siehe Abschnitt 4.3.6)
- Angabe des Schwellenwertes für das Clustering (siehe Abschnitt 4.3.6)

Nach erfolgreicher Generierung der Konformationen werden diese zur Ansicht und eventuell weiteren Analyse automatisch in den TorsionAnalyzer geladen, sofern dies bei den Einstellungen ausgewählt wurde.

Wird bei der Generierung von Molekülkonformationen die Optimierung ausgeschaltet, kann es vorkommen, dass nicht kovalent gebundene Atome überlappen. Dies wird in der Molekül-Ansicht durch eine rot gestrichelte Linie zwischen den beiden betroffenen Atomen dargestellt.

4.4.4. Arbeiten mit Torsionshistogrammen

Die Torsionshistogramme der aktuell ausgewählten Torsionssignatur werden im unteren Teil der Signatur-Ansicht dargestellt. Die Histogramm-Anzeige bietet die Möglichkeit, zwischen verschiedenen Ansichten zu wechseln. Die vorausgewählte Ansicht zeigt das erste Torsionshistogramm der Signatur (im TorsionAnalyzer mit *CSD* bezeichnet, da die Daten aus einem Datensatz der CSD [41] erzeugt wurden, Abschnitt 5.2.2). Die zweite Ansicht zeigt das zweite Torsionshistogramm der Signatur (im TorsionAnalyzer mit *PDB* bezeichnet, da die Daten aus einem Datensatz der PDB [42] erzeugt wurden, Abschnitt 5.2.2). In der dritten Ansicht (im TorsionAnalyzer mit *Both* bezeichnet) werden die Daten aus beiden Histogrammen gemeinsam angezeigt. Für alle drei Ansichten kann jeweils noch die „gefaltete“ Variante ausgewählt werden. Für die gefaltete Variante werden die absoluten Werte der gemessenen Torsionswinkel des jeweiligen Histogramms verwendet,

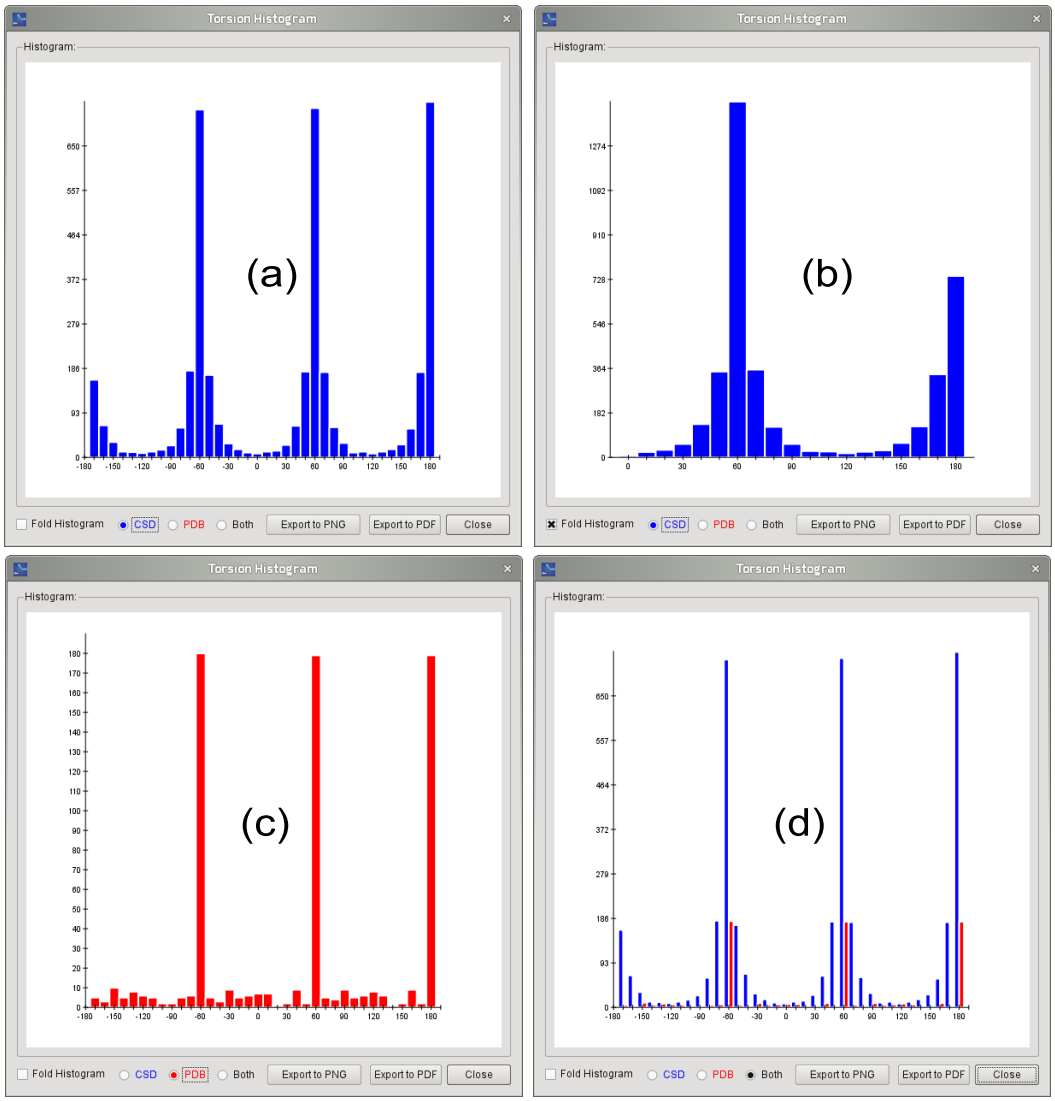


Abbildung 4.13.: Verschiedene Ansichten von Torsionshistogrammen. (a) Verteilung der Torsionswinkelwerte im CSD-Datensatz, (b) gefaltete Variante der CSD-Daten, (c) Verteilung der Torsionswinkelwerte im PDB-Datensatze, (d) gemeinsame Ansicht von CSD- und PDB-Daten.

wodurch die Werte im Bereich [0°, +180°] liegen. Abbildung 4.13 zeigt vier verschiedene Histogramm-Ansichten für eine Torsionssignatur.

Die Generierung eines Histogramms (siehe Abschnitt 4.2.5) erfolgt über ein Kontextmenü der Histogramm-Anzeige. Bei Auswahl des Menüpunktes *Generate Histogram...* bzw. *Generate Second Histogram...* öffnet sich ein Dialogfenster zur Auswahl des Datensatzes, aus dem das Histogramm erzeugt

werden soll. Die Daten können auch hier wieder im MOL2- oder SDF-Format vorliegen. Nach Auswahl des Datensatzes wird die Generierung des Histogramms automatisch gestartet. Ist bereits ein Histogramm vorhanden, wird es vorher gelöscht. Das fertige Histogramm wird automatisch in der Signatur-Ansicht angezeigt, sofern die entsprechende Torsionssignatur noch ausgewählt ist. Da häufig der gleiche Datensatz zur Generierung der Histogramme innerhalb einer Torsionsbibliothek benutzt wird, bietet der TorsionAnalyzer außerdem die Möglichkeit über den Menüpunkt *Torsion Library → Generate all Histograms...* bzw. *Torsion Library → Generate all Second Torsion Histograms...* einen Datensatz auszuwählen und dann Histogramme für alle Torsionssignaturen zu generieren. Über die Menüpunkte *Torsion Library → Generate missing Torsion Histograms...* und *Torsion Library → Generate missing Second Torsion Histograms...* lassen sich fehlende Torsionshistogramme generieren.

Alle Histogramme lassen sich über den Kontextmenüpunkt *Maximize Histogram...* in einem weiteren Dialogfenster vergrößert anzeigen und dann als PDF- oder PNG-Datei exportieren. Das Kontextmenü enthält außerdem einen Menüpunkt zum Löschen eines Histogramms.

Die in Abschnitt 4.2.5 beschriebene automatische Bestimmung von Peaks und Toleranzen eines Histogramms wird über den Kontextmenüpunkt *Analyse Histogram...* bzw. *Analyse Second Histogram...* aufgerufen. Nach Auswahl des Menüpunktes öffnet sich ein weiteres Dialogfenster (siehe Abbildung 4.14) in dem das entsprechende Histogramm mit den um 5° versetzten Klassen angezeigt wird (siehe Abschnitt 4.2.5). In dem Dialogfenster lassen sich die Cutoff-Werte für Peaks und Toleranzen einstellen, welche zusätzlich als unterschiedlich farbige Linien (Cutoff, erste Toleranz, zweite Toleranz) im Histogramm angezeigt werden. Die automatisch abgeleiteten Winkel und Toleranzen werden in einer Tabelle angezeigt. Bei Änderung der Cutoff-Werte werden die Linien im Histogramm automatisch verschoben und die Winkel und Toleranzen werden neu bestimmt. Über den Button *Accept Data* werden die extrahierten Winkel und Toleranzen in die Signatur übernommen. Im Dialogfenster wird außerdem angezeigt, wie viele Datenpunkte das Torsionshistogramm enthält. Bei weniger als 500 Datenpunkten wird ein Hinweis angezeigt, dass die Werte für eine eindeutige Bestimmung von Peaks und Toleranzen eventuell nicht ausreichen. Bei weniger als 100 Datenpunkten wird der Hinweis angezeigt, dass nicht genug Werte zur Bestimmung von Peaks und Toleranzen vorhanden sind. Die Tabelle bleibt leer und die Eingabefelder für die Cutoff-Werte und der *Accept Data*-Button sind deaktiviert. Sind die Daten im Histogramm gleichmäßig verteilt, sind die Eingabefelder für die Cutoff-Werte ebenfalls deaktiviert, da hier ein 30°-Raster zur

Definition der Winkel verwendet wird (siehe Abschnitt 4.2.5). Über den Menüpunkt *Torsion Library → Analyse all Torsion Histograms...* bzw. *Torsion Library → Analyse all Second Torsion Histograms...* werden die Histogramme aller Torsionssignaturen analysiert und die extrahierten Daten automatisch übernommen. Die Bestimmung der Peaks und Toleranzen erfolgt hierbei mit den in Abschnitt 4.2.5 angegebenen Cutoff-Werten.

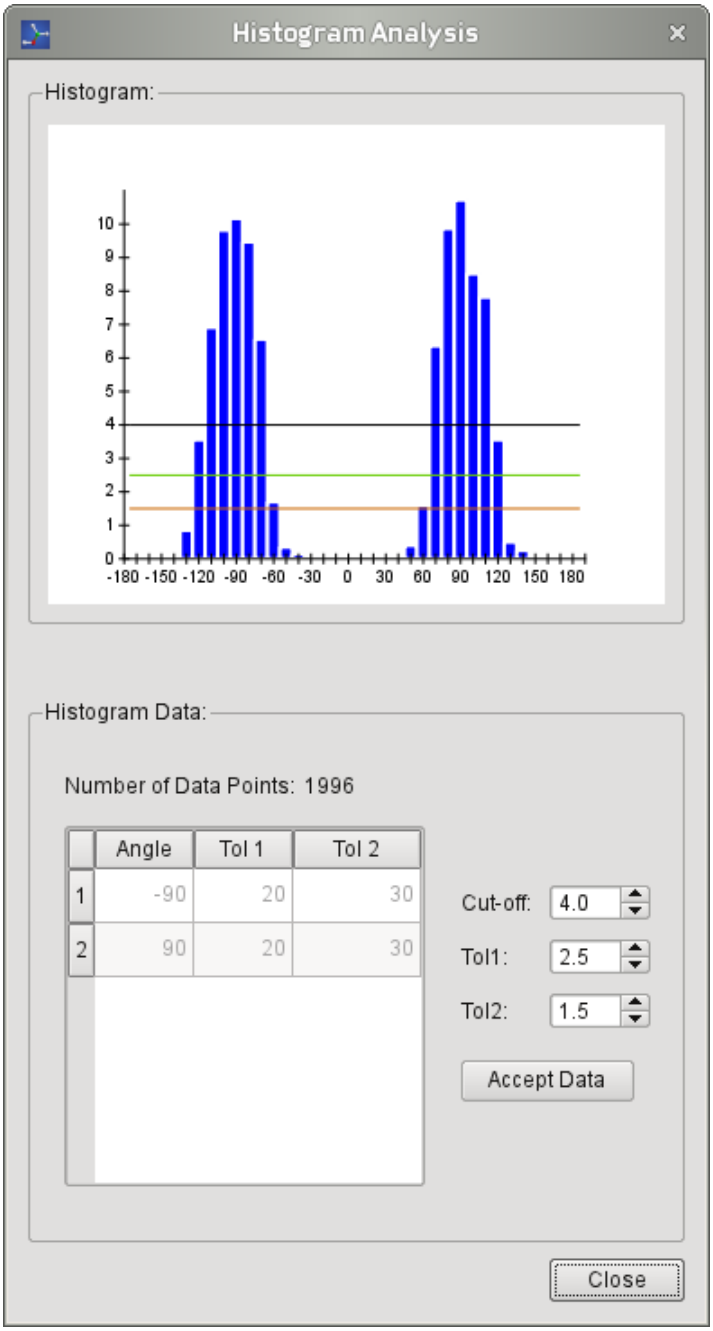


Abbildung 4.14.: Dialogfenster zur Analyse eines Torsionshistogramms. Im oberen Teil wird das Histogramm inklusive Cutoff-Linien für Peaks und Toleranzen angezeigt. Extrahierte Winkel und Toleranzen werden in der unteren Tabelle angezeigt. Die Cutoff-Werte für Peaks und Toleranzen können über die entsprechenden Eingabefelder eingestellt werden. Mit dem Button *Accept Data* werden die extrahierten Daten in die Torsionssignatur übernommen.

Kapitel 5

5 Evaluierung

In diesem Kapitel werden die Datensätze und Methoden zur Evaluierung der im vorherigen Kapitel vorgestellten Methoden zur Analyse und Generierung von Konformationen beschrieben. Die Evaluierung des TFD, der Torsionsbibliothek und der Konformationsgenerierung erfolgt jeweils getrennt voneinander. Da der TorsionAnalyzer in erster Linie der explorativen Unterstützung eines Experten dient, ist im Rahmen dieser Dissertation eine Evaluierung mit Hilfe von Referenzdaten nicht sinnvoll. Die Ergebnisse der einzelnen Evaluierungen werden in Kapitel 6 beschrieben und diskutiert.

5.1. TFD

Um zu untersuchen, ob der TFD ein geeignetes und intuitives Maß zum Vergleich von Konformationen ist und ob einige der in Abschnitt 3.1.1 beschriebenen Nachteile des relativen RMSD behoben werden konnten, wurden TFD- und RMSD-Werte zwischen bioaktiven Konformationen und den dazugehörigen generierten Konformationen berechnet. Um noch ein weiteres unabhängiges Maß zu haben, wurde der TFD zusätzlich noch mit dem *TanimotoShape* verglichen. Der auf 1 normalisierte TanimotoShape bewertet die Ähnlichkeit zweier Moleküle anhand ihrer Form und ist Teil des Programms ROCS (siehe Abschnitt 3.1.5). Je höher der TanimotoShape, desto besser ist die Überlagerung und desto ähnlicher sind sich die Moleküle.

Initiale Tests des TFD haben ergeben, dass Konformationen mit einem TFD < 0.2 der bioaktiven Konformation sehr ähnlich sind und dadurch sehr wahrscheinlich gute Ergebnisse in späteren Virtuellen-Screenings liefern werden. Bei der Evaluierung wurde daher ein TFD-Cutoff von 0.2 benutzt.

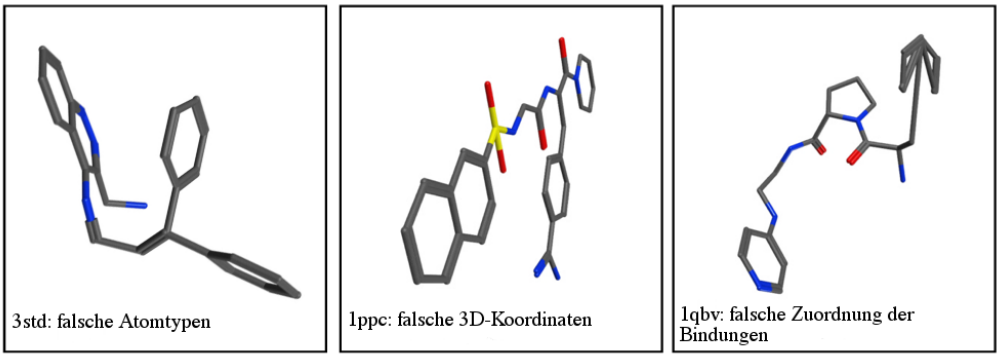


Abbildung 5.1.: Drei Beispiele für Liganden, welche aufgrund von fehlerhaften PDB-Einträgen aus dem Datensatz zur TFD-Evaluierung aussortiert wurden. Die Liganden sind aus den PDB-Komplexen 3std [111], 1ppc [112] und 1qbv [113].

Aufgrund von Erfahrungswerten aus internen virtuellen Screenings der F. Hoffmann-La Roche Inc wurde ein TanimotoShape-Cutoff von 0.75 benutzt. Für RMSD-Werte wurde ein Cutoff von 1.5 Å benutzt (siehe Abschnitt 3.1.1).

Um den TFD unabhängig von der in 4.3 vorgestellten Methode zur Konformationsgenerierung zu evaluieren, wurden die Konformationen mit OMEGA (Version 2.4.1) generiert. Die Aufbereitung der Daten, die Überlagerung der Konformationen und die RMSD-Berechnungen wurden mit MOE (Version 2010.10) durchgeführt. Die TanimotoShape-Berechnungen wurden mit ROCS (Version 3.1.0) durchgeführt.

Der für die Evaluierung des TFD verwendete Datensatz [16] enthält insgesamt 667 Strukturen aus drei verschiedenen Publikationen [16,37,110]. Die Liganden wurden aus der PDB extrahiert und in MOE manuell verifiziert. 40 Liganden wurden aufgrund von Fehlern im PDB-Eintrag aussortiert. Abbildung 5.1 zeigt drei Beispiele für Liganden aus fehlerhaften PDB-Einträgen.

Für die 627 Liganden aus dem Datensatz wurden insgesamt ca. 71.500 Konformationen mit OMEGA (Standardeinstellungen) generiert. Da OMEGA für 23 Liganden keine Konformationen generieren konnte, wurden diese ebenfalls aus dem Datensatz genommen, so dass der endgültige Datensatz 604 Liganden enthält. Mit den Standardeinstellungen von OMEGA werden für jedes Molekül maximal 200 Konformationen generiert. Diese Obergrenze wird für etwa 50% der Liganden erreicht, was zur Folge hat, dass die Konformationsmenge für diese Liganden nicht umfassend dargestellt wird. Abbildung 5.2 zeigt die Anzahl der generierten Konformationen in Abhängigkeit von der Anzahl der rotierbaren Bindungen. Bereits bei

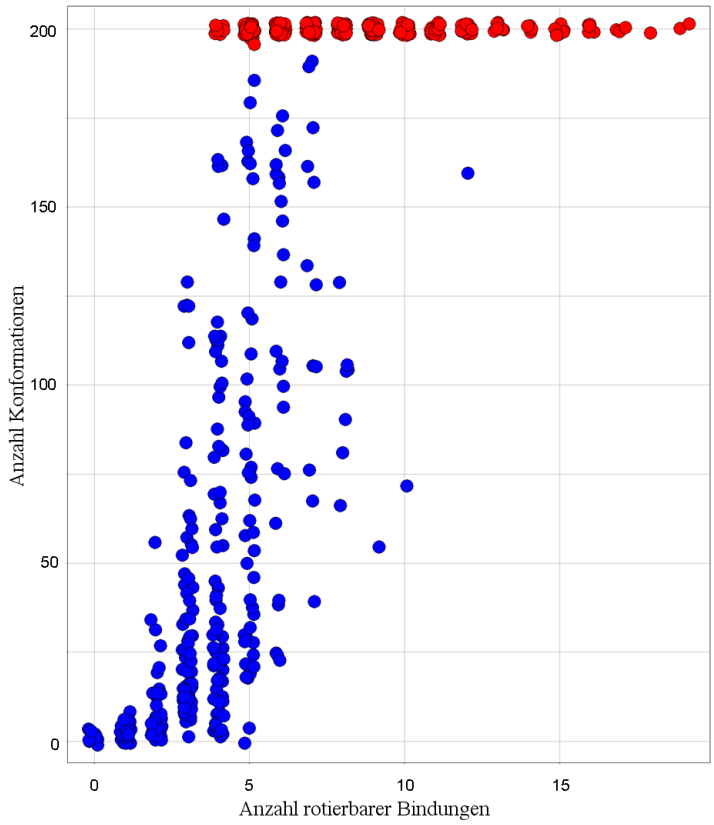


Abbildung 5.2.: Abhängigkeit der Anzahl der generierten Konformationen von der Anzahl der rotierbaren Bindungen. Die roten Kreise markieren Liganden, bei denen die maximale Anzahl von 200 Konformationen erreicht wurde.

Liganden mit vier rotierbaren Bindungen wurde für einige die maximale Anzahl an Konformationen erreicht. Für jeden Liganden, bei dem die Konformationsmenge nicht vollständig ist, steigt die Wahrscheinlichkeit, dass die Konformation, welche der bioaktiven Konformation am nächsten ist, nicht in der Konformationsmenge enthalten ist, was wiederum zur Folge hat, dass der durchschnittliche relative RMSD ansteigt.

Um genauer zu untersuchen, wie sich die normalisierten TFD-Werte gegenüber den nicht normalisierten RMSD-Werten verhalten, wurden drei Kohlenstoffketten unterschiedlicher Länge generiert: *kurz* (drei Bindungen), *mittel* (fünf Bindungen) und *lang* (7 Bindungen). Die beiden terminalen Bindungen werden dabei jeweils als nicht rotierbar angesehen. Für jede Kette wurden drei unterschiedliche Konformationen generiert, so dass die rotierbaren Bindungen der zweiten Konformationen jeweils um die Hälfte der maximal möglichen Abweichung von den rotierbaren Bindungen der

Tabelle 5.1.: TF für jeweils drei Konformationen von drei unterschiedlich langen Kohlenstoffketten.

Molekül	Torsion-Fingerprint
Kurz 1	(60.00 180.00 60.00)
Mittel 1	(60.00 180.00 180.00 180.00 60.00)
Lang 1	(60.00 180.00 180.00 180.00 180.00 180.00 60.00)
Kurz 2	(60.00 270.00 60.00)
Mittel 2	(60.00 270.00 90.00 90.00 60.00)
Lang 2	(60.00 270.00 270.00 270.00 270.00 270.00 60.00)
Kurz 3	(60.00 0.00 60.00)
Mittel 3	(60.00 0.00 0.00 0.00 60.00)
Lang 3	(60.00 0.00 0.00 0.00 0.00 0.00 60.00)

ersten Konformationen abweichen und die rotierbaren Bindungen der dritten Konformationen jeweils maximal von den rotierbaren Bindungen der ersten Konformationen abweichen. Die TF der neun Konformationen sind in Tabelle 5.1 angegeben. Für jede Gruppe wurden anschließend TFD- und RMSD-Werte berechnet und miteinander verglichen, wobei jeweils die erste Konformation als Referenzstruktur benutzt wurde.

5.2. Torsionsbibliothek

Die Evaluierung der Torsionsbibliothek gliedert sich in zwei Teile. Im ersten Teil wird untersucht, wie gut die Torsionssignaturen den für die Medizinalchemie relevanten chemischen Raum abdecken. Im zweiten Teil wird die Verteilung der Torsionswinkel in zwei in der Medizinalchemie häufig genutzten Datenbanken, CSD und PDB (siehe auch Abschnitt 2.3), miteinander verglichen.

5.2.1. Abdeckung des chemischen Raumes

Die Torsionsbibliothek umfasst etwa 500 Torsionssignaturen. Um zu untersuchen, in wie weit die 500 Signaturen den für die Medizinalchemie relevanten chemischen Raum abdecken, ob es systematische Lücken gibt und in welchen Fällen nur generische Muster zugeordnet werden, wurde als Datensatz eine möglichst diverse, aber dennoch repräsentative Teilmenge der

ChEMBL-Datenbank (Version 13) [114] verwendet. Die ChEMBL-Datenbank enthält mehr als eine Million wirkstoffähnliche bioaktive Moleküle, was diese Datenbank zu einem idealen Referenzdatensatz macht. Um die Teilmenge zu erstellen, wurde zuerst eine Suchanfrage nach Molekülen mit einer Molekülmasse zwischen 200 und 500 gestellt. Von den 908.007 resultierenden Molekülen wurde mit Pipeline Pilot [115] eine diverse Teilmenge von 100.000 Molekülen extrahiert. Der Funktionsumfang von PipelinePilot bietet die Möglichkeit, für eine Menge von Eigenschaften (wie zum Beispiel alogP, Molekülmasse, Anzahl der Donoren, Anzahl der Akzeptoren, Anzahl der rotierbaren Bindungen, Anzahl der Atome, Anzahl der Ringe, Anzahl der aromatischen Ringe) eine Teilmenge mit maximaler Diversität zu extrahieren. Anschließend wurde mit Hilfe des TorsionAnalyzers untersucht, welche Torsionssignatur wie oft einer rotierbaren Bindung des Datensatzes zugeordnet wurde.

5.2.2. Vergleich von CSD- und PDB-Histogrammen

Um die Verteilung von Torsionswinkeln in der CSD und der PDB miteinander zu vergleichen, wurden zwei verschiedene Datensätze generiert. Für den ersten Datensatz wurden mit ConQuest (Version 1.14) aus der CSD (Version 2011) alle Moleküle extrahiert, bei denen 3D-Koordinaten vorliegen und die mindestens ein Kohlenstoffatom beinhalten. Aus den ca. 580.000 resultierenden Molekülen wurden die folgenden Einträge gelöscht:

- Moleküle mit anderen Elementen als H, C, N, O, F, Cl, Br, I, S oder P
- Ionen
- mit Röntgen-Pulverdiffraktometrie (XRPD) aufgeklärte Strukturen
- Organometallische Verbindungen
- Strukturen mit einem R-Faktor $< 10\%$

Die ca. 145.000 verbliebenen Moleküle wurden im MOL2-Format exportiert und mit CORINA (Version 3.46, Optionen: *no3d*, *newtypes*, *rs*) weiter bearbeitet, um allen Molekülen konsistente Atomtypen zuzuweisen. Der finale CSD-Datensatz enthält etwa 140.000 Moleküle.

Für den zweiten Datensatz wurden Liganden aus allen HET-Einträgen, mit Ausnahme von Metallen und häufig vorkommenden Ionen, aus Protein-Ligand-Komplexen in Proasis2 [116], einer überarbeiteten Version der PDB, extrahiert. Liganden mit weniger als 5 oder mehr als 100 Atomen wurden

dabei verworfen. Der endgültige PDB-Datensatz enthält 77.065 Moleküle aus 24.163 PDB-Einträgen.

Aus beiden Datensätzen wurden mit dem TorsionAnalyzer für jede Torsionssignatur ein CSD- und ein PDB-Histogramm generiert und diese anschließend manuell untersucht.

5.3. CONFECT

Die Evaluierung der in Abschnitt 4.3 vorgestellten Methode zur Generierung von Konformationen gliedert sich in vier Teile, in denen folgende Fragen behandelt werden:

1. In wie weit ist die Methode in der Lage, die bioaktive Konformation zu reproduzieren? Wie viele Konformationen werden dabei durchschnittlich generiert? Wie ist die durchschnittliche Laufzeit pro Molekül?
2. Wie schneidet die Methode im Vergleich zu anderen Methoden ab?
3. Ist die Methode in der Lage, mehrere bioaktive Konformationen des gleichen Moleküls zu reproduzieren?
4. Wie ist die Stabilität und das Laufzeitverhalten der Methode bei einem größeren Datensatz?

5.3.1. Reproduktion der bioaktiven Konformation

Im ersten Teil der Evaluierung wird untersucht, in wie weit CONFECT in der Lage ist, die bioaktive Konformation (in diesem Fall die Kristallstruktur) zu reproduzieren. Dazu wurde zuerst für jede Konformation der RMSD-Wert zur Kristallstruktur berechnet (siehe auch Abschnitt 3.1.1) und anschließend die folgenden vier Prozentsätze bestimmt:

1. $P_{0,5}$: Prozentsatz der Liganden mit mindestens einer Konformation mit RMSD $\leq$ 0,5 Å
2. $P_{1,0}$: Prozentsatz der Liganden mit mindestens einer Konformation mit RMSD $\leq$ 1,0 Å
3. $P_{1,5}$: Prozentsatz der Liganden mit mindestens einer Konformation mit RMSD $\leq$ 1,5 Å

4. $P_{2,0}$: Prozentsatz der Liganden mit mindestens einer Konformation mit RMSD $\leq$ 2,0 Å

Zur Evaluierung wurden dabei zwei verschiedene Datensätze verwendet. Der erste Datensatz [37] enthält 100 Liganden aus PDB-Strukturen. Anhand der in der Publikation angegebenen PDB-IDs wurden die entsprechenden Protein-Ligand-Komplexe aus der PDB heruntergeladen. Die Liganden wurden anschließend extrahiert und analysiert (MOE) und, falls nötig, manuell korrigiert. Die so entstandenen Daten wurden später als bioaktive Konformationen für die RMSD-Berechnungen verwendet. Um eine von der Kristallstruktur unbeeinflusste Startkonformation zu erhalten, wurden für alle Liganden mit coord3d (siehe Abschnitt 3.3.5) neue 3D-Koordinaten erzeugt. Für die 100 Liganden wurden Konformationen mit den ersten sieben Qualitätsstufen (siehe auch Abschnitt 4.3.3) generiert und die Ergebnisse ($P_{0,5}$, $P_{1,0}$, $P_{1,5}$, $P_{2,0}$, Anzahl der generierten Konformationen und die Laufzeit) miteinander verglichen. Bei der Generierung der Konformationen wurden die folgenden Einstellungen (von hier an als Standardeinstellungen bezeichnet) benutzt:

- eingeschaltete Optimierung (siehe Abschnitt 4.3.7)
- Generierung von Ringkonformationen (siehe Abschnitt 4.3.2)
- TFD-Clustering mit einem Schwellenwert von 0,01 (siehe Abschnitt 4.3.6)

Als zweiter Datensatz wurde der Iridium-Datensatz [117] verwendet. Die Liganden wurden von der entsprechenden OpenEye-Webseite heruntergeladen. Liganden ohne rotierbare Bindungen wurden entfernt. Diese Daten wurden dann ebenfalls wieder als bioaktive Konformation für die RMSD-Berechnung verwendet. Mit der Qualitätsstufe 21 und den Standardeinstellungen wurden Konformationen für die 114 Liganden aus dem Datensatz generiert (Iridium deposited). Anschließend wurden mit coord3d neue 3D-Koordinaten für die Liganden erzeugt und Konformationen mit der Qualitätsstufe 21 und Standardeinstellungen generiert (Iridium refined). Beide Ergebnisse wurden dann mit den Ergebnissen aus dem vorherigen Experiment (Perola100) verglichen.

5.3.2. Vergleich mit anderen Methoden

Die Ergebnisse, die mit der Qualitätsstufe 21 für den Perola100-Datensatz aus dem vorherigen Experiment erreicht wurden, wurden mit den Ergebnissen von Catalyst, ICM, OMEGA und ConfGen verglichen. Die Ergebnisse der

vier Programme wurden aus einer Publikation von Watts und Kollegen [16] entnommen.

5.3.3. Reproduktion mehrerer bioaktiver Konformationen

Da es nur sehr wenige Daten zu Molekülen mit mehreren bioaktiven Konformationen gibt, wurde dieses Experiment am Beispiel von Adenosinmonophosphat (AMP) durchgeführt. AMP hat mehrere unterschiedliche Funktionen im Metabolismus und kommt daher in verschiedenen Protein-Ligand-Komplexen der PDB vor. Insgesamt gibt es 367 AMP-Strukturen in der PDB (Stand vom 11.12.2012), von denen für 343 Strukturen vollständige 3D-Koordinaten vorhanden sind. Diese Strukturen wurden aus der PDB heruntergeladen und später als bioaktive Konformationen für die RMSD-Berechnungen verwendet. Für eine der AMP-Strukturen wurden mit coord3d neue 3D-Koordinaten erzeugt und anschließend Konformationen mit der Qualitätsstufe 21 und Standardeinstellungen generiert. Für jede der 343 AMP-Strukturen wurde untersucht, ob es eine generierte Konformation mit einem RMSD ≤ 0.5 Å oder ≤ 1.0 Å gibt.

5.3.4. Laufzeitverhalten

Für den letzten Teil der Evaluierung von CONFECT wurde der ChEMBL-Datensatz aus der Evaluierung der Torsionsbibliothek (siehe Abschnitt 5.2.1) benutzt. Konformationen wurden mit der Qualitätsstufe 1 und Standardeinstellungen generiert. Anschließend wurde untersucht, ob für alle 908.007 Moleküle Konformationen generiert werden konnten, wie viele Konformationen durchschnittlich generiert wurden, und wie hoch die durchschnittliche Laufzeit pro Molekül ist.

Kapitel 6

6 Resultate und Diskussion

In diesem Kapitel werden die Ergebnisse aus der Evaluierung des TFD, der Torsionsbibliothek und der Konformationsgenerierung vorgestellt und diskutiert. Die Datensätze und Evaluierungsmethoden wurden bereits in Kapitel 5 beschrieben.

6.1. TFD

Zur Evaluierung des TFD wurden für die 627 Liganden des Datensatzes (siehe Abschnitt 5.1) insgesamt ca. 71.500 Konformationen generiert. Anschließend wurden TFD-, RMSD- und TanimotoShape-Werte berechnet und miteinander verglichen. Die Ergebnisse aus der Evaluierung werden im folgenden Abschnitt vorgestellt und diskutiert.

Bezogen auf alle Konformationen des Datensatzes lässt sich keine Korrelation zwischen TFD- und RMSD-Werten erkennen (Abbildung 6.1 oben, $R^2 = 0,1$). Es gibt sowohl Konformationen mit einem sehr kleinen TFD- ($< 0,25$) aber hohem RMSD-Wert ($> 4\,\text{Å}$) als auch Konformationen mit einem hohen TFD- ($> 0,8$), aber sehr kleinem RMSD-Wert ($< 1\,\text{Å}$). Werden die Konformationen nach ihrer Flexibilität eingeteilt, lassen sich allerdings leichte Tendenzen innerhalb der einzelnen Gruppen erkennen (Abbildung 6.1 unten). Die Ergebnisse zeigen außerdem, dass der maximale TFD-Wert von 1 für keines der Moleküle aus dem Datensatz erreicht wurde. Dies kann unterschiedliche Gründe haben. Das zur Generierung der Konformationen verwendete Programm OMEGA benutzt eine wissensbasierte Methode zur Generierung von Konformationen. Es kann also sein, dass Konformationen mit maximal abweichenden Torsionswinkeln nicht generiert werden, da

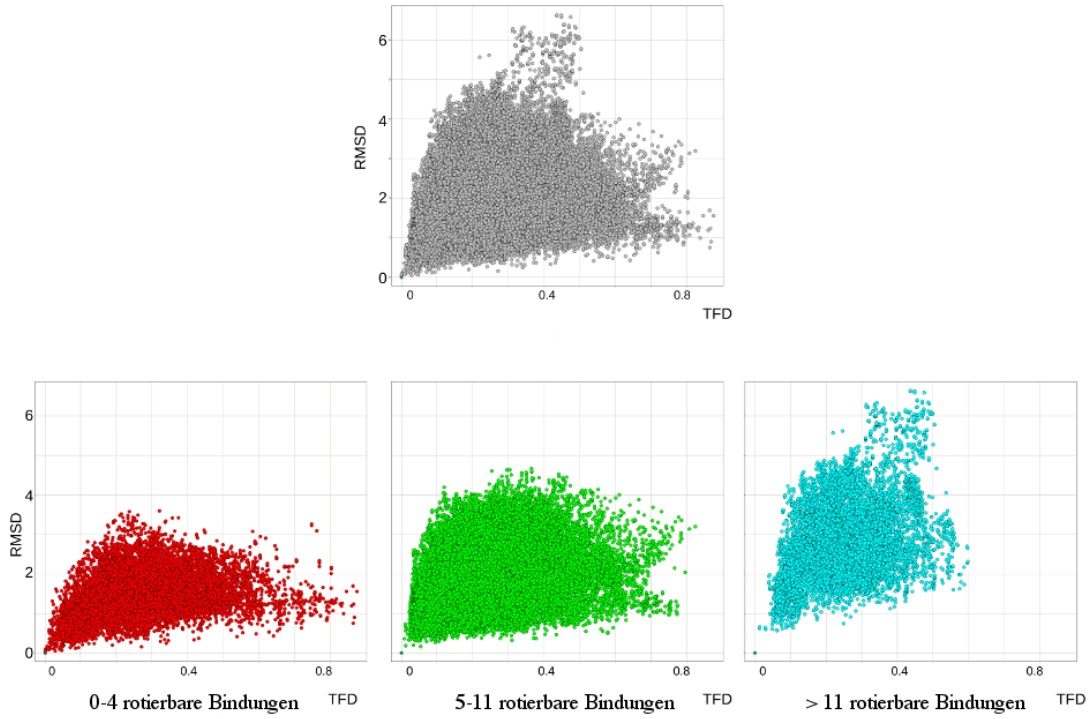


Abbildung 6.1.: Oben: TFD- und relative RMSD-Werte für alle ca. 71.500 Konformationen des Datensatzes. Unten: TFD- und relative RMSD-Werte aufgeteilt nach der Flexibilität (Anzahl der rotierbaren Bindungen) der Konformationen.

diese in der Wissensbasis nicht oder nur selten beobachtet wurden. Zudem können maximal abweichende Torsionswinkel zu stark überlappenden Atomen führen und die Konformationen deshalb verworfen werden. Die fehlende Korrelation zwischen TFD- und RMSD-Werten zeigt, dass beide Maße die Konformationen unterschiedlich bewerten. Welches Maß dabei die sinnvollere Bewertung liefert, lässt sich aus dieser allgemeinen Betrachtung allerdings noch nicht erkennen. Die TFD- und RMSD-Werte der vier folgenden Moleküle des Datensatzes sollen deswegen exemplarisch die Vor- und Nachteile des TFD gegenüber dem RMSD zeigen. Zudem wird geklärt, warum es keine Konformation mit maximalem TFD-Wert gibt.

6.1.1. 1acl

Der Ligand aus dem PDB-Komplex 1acl [118] ist mit 11 rotierbaren Bindungen sehr flexibel. Er besteht aus zwei positiv geladenen terminalen Ammonium-Gruppen, die durch eine lange aliphatische Kette mit 10 Kohlenstoffatomen verbunden sind (Abbildung 6.2). Die TFD- und relativen RMSD-Werte der Konformationen sind nicht korreliert ($R^2 = 0,0$). Die Konformationen lassen sich anhand des Torsionswinkels der zentralen Bindung (C-C-C-C) in drei Gruppen einteilen. Die Konformationen der ersten Gruppe sind mit einer Torsionswinkelabweichung von nur 7° der bioaktiven Konformation am nächsten. Dies spiegelt sich auch in den TFD-Werten wider, welche im allgemeinen niedriger sind als die der anderen beiden Gruppen. Die Konformationen der anderen beiden Gruppen weisen einen Torsionswinkel auf, der stark von dem der bioaktiven Konformation abweicht und mit einem TFD-Cutoff von 0,2 wird keine der Konformation als ähnlich zur bioaktiven Konformation klassifiziert. Torsionswinkelabweichungen bei der betrachteten Bindung haben den stärksten Einfluss auf die Konformation, was in der TFD-Berechnung durch eine stärkere Gewichtung von Abweichungen bei zentralen Torsionswinkeln abgebildet wird. Die Bewertung der Konformationen durch den TFD entspricht den Erwartungen. Die Überlagerung von zwei Konformationen mit der bioaktiven Konformation (Abbildung 6.2) zeigt allerdings eine Einschränkung des TFD. Beide Konformationen haben den gleichen, sehr niedrigen relativen RMSD-Wert (0,54 Å), aber einen unterschiedlich hohen TFD-Wert (0.1 und 0.4). Mit dem RMSD-Cutoff von 1,5 Å werden beide Konformationen, mit einem TFD-Cutoff von 0,2 hingegen nur eine Konformation als ähnlich zur bioaktiven Konformation klassifiziert. Die Überlagerung zeigt, dass bei allen drei Konformationen die positiv geladenen Ammonium-Gruppen trotz der starken Torsionswinkel-Abweichungen in der aliphatischen Kette sehr gut überlagern. Der TFD ist aufgrund der unabhängigen Betrachtung der einzelnen Torsionswinkel nicht in der Lage zu erkennen, dass sich die Torsionswinkel-Abweichungen hier gegenseitig aufheben und beide Ammonium-Gruppen am Ende somit fast wieder an der gleichen Stelle liegen. Der RMSD liefert hier also die sinnvollere Bewertung der Konformationen. Beim Vergleich von TFD- und TanimotoShape-Werten zeigt sich ebenfalls keine Korrelation ($R^2 = 0,0$). Die beiden mit der bioaktiven Konformation überlagerten Konformationen haben TanimotoShape-Werte knapp über dem Cutoff (0,76 und 0,77) und werden somit als ähnlich zur bioaktiven Konformation klassifiziert. Der TanimotoShape liefert hier also ebenfalls eine sinnvollere Bewertung als der TFD.

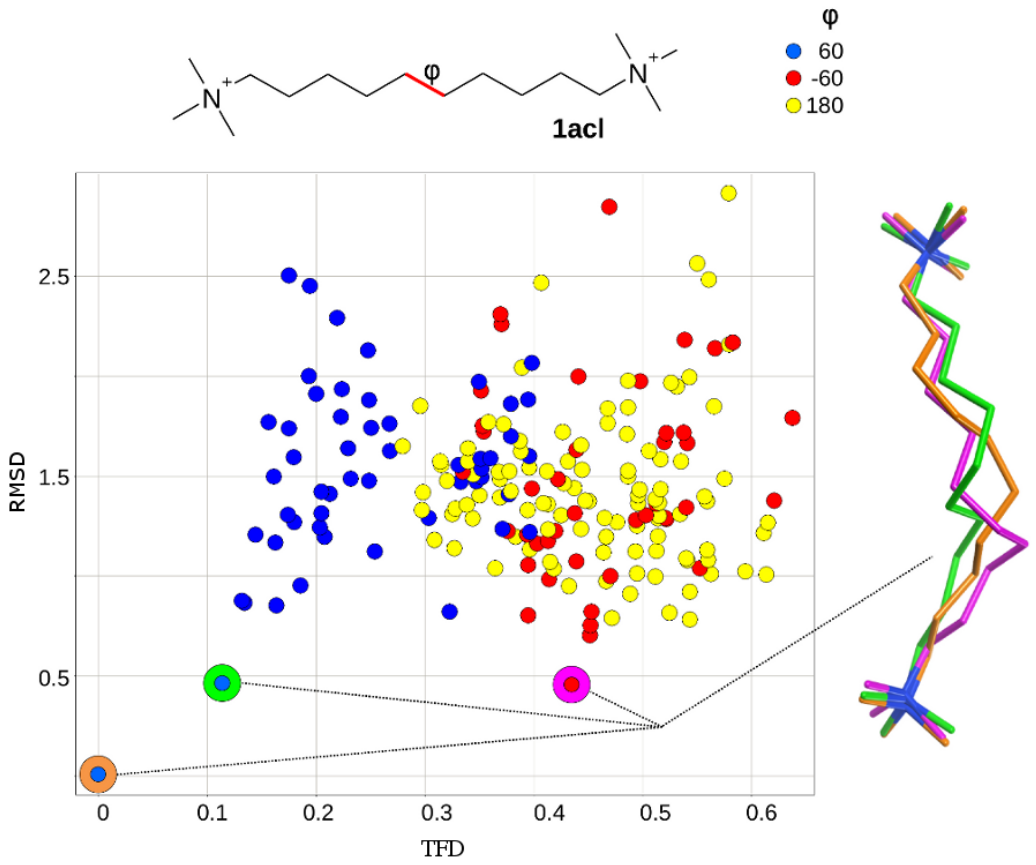


Abbildung 6.2.: TFD- und relative RMSD-Werte für den Liganden aus dem PDB-Komplex 1acl. Die Konformationen sind nach dem Torsionswinkel der rot markierten Bindung in drei Gruppen eingeteilt (blau: 60°, rot: −60°, gelb: 180°). Die bioaktive Konformation hat einen Torsionswinkel von 67°. Die Torsionswinkelabweichungen der einzelnen Gruppen betragen somit: 7°, 127° und 113°. Rechts ist die Überlagerung von zwei Konformationen mit der bioaktiven Konformation (orange gefärbte Struktur) dargestellt. $R^2 = 0,00$

6.1.2. 1gj5

Der Ligand aus dem PDB-Komplex 1gj5 [119] gehört zu den eher kompakteren und weniger flexiblen Molekülen. Der Ligand hat drei rotierbare Bindungen, so dass insgesamt nur neun Konformationen generiert wurden (Abbildung 6.3). Bei diesem Beispiel sind sowohl die TFD- und relativen RMSD-Werte, als auch die TFD- und TanimotoShape-Werte stark korreliert ($R^2 = 0,84$ und $R^2 = 0,81$). Es gibt lediglich eine Konformation, die von allen drei Maßen als unähnlich zur bioaktiven Konformation klassifiziert wird. Abbildung 6.3 zeigt die Überlagerung dieser Konformation mit der bioaktiven Konformation. Unter der Annahme, dass sowohl die Pi-Interaktion der äußeren Ringe als auch die direkte Interaktion via Wasserstoffbrückenbindung der Amidin-Gruppe wichtig für die Bindung an das Protein sind, zeigt die Überlagerung, dass diese Interaktionen für die ausgewählte Konformation wohl nicht zustande kommen. Die Konformation wurde also in diesem Fall durch alle drei Maße korrekt bewertet.

6.1.3. 1k7f

Im Gegensatz zu dem Liganden aus dem PDB-Komplex 1gj5 ist der Ligand aus dem PDB-Komplex 1k7f [120] sehr flexibel, so dass die maximale Anzahl von 200 generierten Konformationen erreicht wurde (Abbildung 6.4). Die TFD- und relativen RMSD-Werte sind bei diesem Beispiel leicht korreliert ($R^2 = 0,38$). Mit dem RMSD-Cutoff von 1,5 Å werden 50 Konformationen als ähnlich zur bioaktiven Konformation klassifiziert. Bei einem TFD-Cutoff von 0,2 werden hingegen nur 22 Konformationen als ähnlich zur bioaktiven Konformation klassifiziert. Abbildung 6.4 zeigt die Überlagerung der bioaktiven Konformation mit einer Konformation, die einen niedrigen relativen RMSD-Wert (1 Å) aber einen hohen TFD-Wert (0,5) hat. Die Richtung der Wasserstoffbrücke durch die zentrale Amid-Gruppe der ausgewählten Konformation weicht stark von der der bioaktiven Konformation ab. Somit ist eine Bindung an das Zielprotein sehr unwahrscheinlich. Die Ergebnisse zeigen, dass es noch weitere Konformationen gibt, deren Werte über dem TFD-, aber unter dem RMSD-Cutoff liegen. Umgekehrt gibt es keine einzige Konformation, deren Werte unter dem TFD-, aber über dem RMSD-Cutoff liegen. Hier liefert der TFD somit die sinnvollere Bewertung. Es besteht eine leichte Korrelation zwischen TFD- und Tanimoto-Shape-Werten ($R^2 = 0,3$) und fast alle Konformationen mit einem Wert über dem TFD-, aber unter dem RMSD-Cutoff haben auch einen TanimotoShape-Wert unter dem Cutoff von 0,75. Allerdings gibt es auch Ausnahmen: der TanimotoShape-Wert

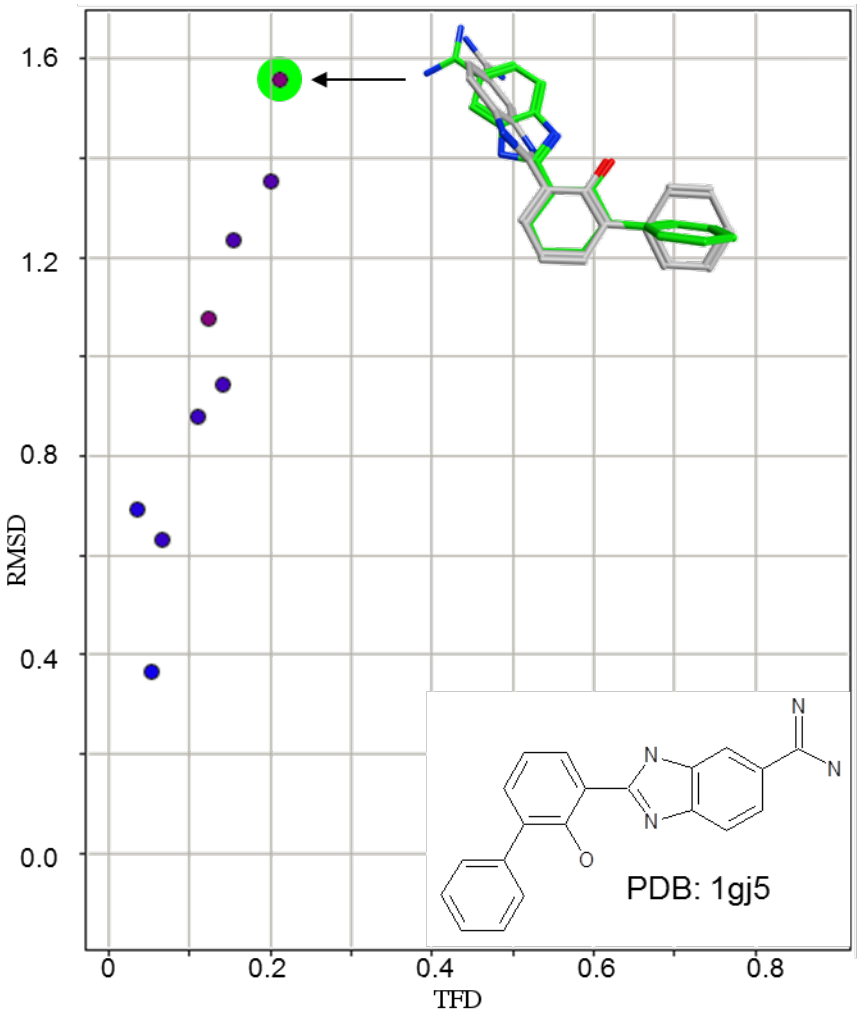


Abbildung 6.3.: TFD- und RMSD-Werte für den Liganden aus dem PDB-Komplex 1gj5. Die Punkte sind nach TanimotoShape (0: rot – 1: blau) eingefärbt. Zusätzlich wird die Überlagerung einer ausgewählten Konformation (grün gefärbte Struktur) mit der bioaktive Konformation (grau gefärbte Struktur) gezeigt. $R^2 = 0,84$

für die mit der bioaktiven Konformation überlagerte Konformation zum Beispiel liegt mit 0,83 über dem Cutoff von 0,75.

Dieses Beispiel zeigt außerdem, warum es keine Konformation mit einem TFD-Wert von 1 gibt. Der Ligand aus dem PDB-Komplex 1k7f hat eine zentrale Amid-Bindung. Da Torsionswinkelabweichungen an dieser Stelle den stärksten Einfluss auf den TFD haben, müsste eine Konformation mit einem TFD nahe an 1 hier eine maximale Abweichung von 180° haben. Bei einer solchen Konformation würde aus dem favorisierten trans-Amid ein cis-Amid werden, was allerdings durch eine entsprechende Regel in OMEGA ausgeschlossen ist und diese Konformation somit gar nicht erst generiert wird.

6.1.4. 1ela

Der Ligand aus dem PDB-Komplex 1ela [121] ist, ebenso wie der Ligand aus dem vorherige Beispiel, sehr flexibel, so dass die maximale Anzahl von 200 generierten Konformationen erreicht wurde (Abbildung 6.5). Auch hier lässt sich aus den Ergebnissen eine Korrelation zwischen TFD- und relativen RMSD-Werten erkennen ($R^2 = 0,68$). Es gibt eine Konformation, deren RMSD-Wert kurz unter dem RMSD-Cutoff von 1,5 Å liegt, der TFD-Wert aber leicht über dem Cutoff von 0,2 liegt. Ein großer Teil der Konformationen wird allerdings nur mit dem TFD als ähnlich zur bioaktiven Konformation klassifiziert. In Abbildung 6.5 sind zwei Überlagerungen mit der bioaktiven Konformation dargestellt. Mit dem TFD wird eine der Konformationen als ähnlich zur bioaktiven Konformation klassifiziert. Mit dem RMSD wird hingegen keine der beiden Konformationen als ähnlich zur bioaktiven Konformation klassifiziert. Die Interaktionsrichtung der zentralen Amid-Gruppe mit dem höheren TFD-Wert unterscheidet sich sehr stark von der Interaktionsrichtung der Amid-Gruppe in der bioaktiven Konformation. Die Interaktionen zeigen fast in entgegengesetzte Richtungen, so dass hier mit hoher Wahrscheinlichkeit keine Interaktion mit dem Zielprotein stattfinden kann. Bei der Konformation mit dem niedrigeren TFD-Wert ist die Abweichung der Interaktionsrichtung der Amid-Gruppe zur bioaktiven Konformation weniger stark, so dass hier eine Interaktion mit dem Zielprotein sehr wahrscheinlich stattfindet. Wie im vorherigen Beispiel liefert auch hier der TFD eine sinnvollere Bewertung als der RMSD. Die Korrelation zwischen TFD- und TanimotoShape-Werten ($R^2 = 0,53$) ist bei diesem Beispiel etwas schwächer als die Korrelation zwischen TFD- und RMSD-Werten. Mit dem TanimotoShape wird keine der beiden überlagerten Konformationen als ähnlich zur bioaktiven Konformation klassifiziert. Hier

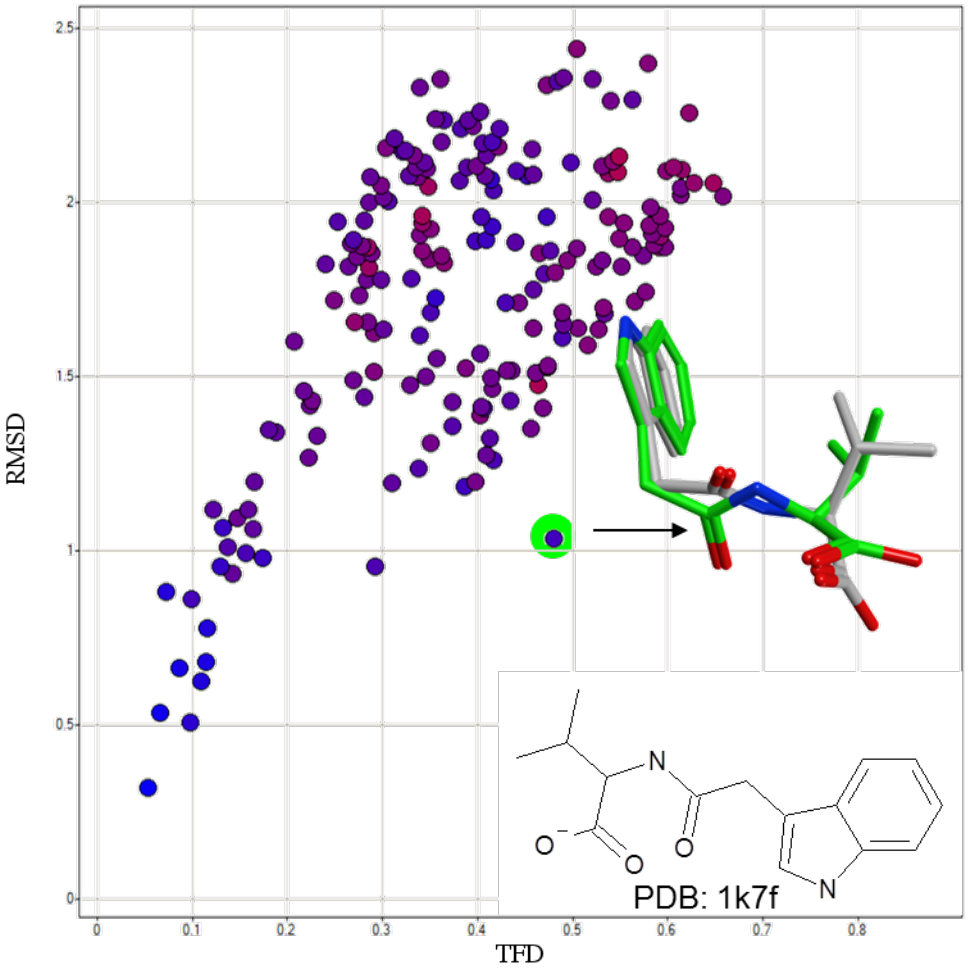


Abbildung 6.4.: TFD- und RMSD-Werte für den Liganden aus dem PDB-Komplex 1k7f. Die Punkte sind nach TanimotoShape (0: rot – 1: blau) eingefärbt. Zusätzlich wird die Überlagerung einer ausgewählten Konformation (grün gefärbte Struktur) mit der bioaktiven Konformation (grau gefärbte Struktur) gezeigt. $R^2 = 0,38$

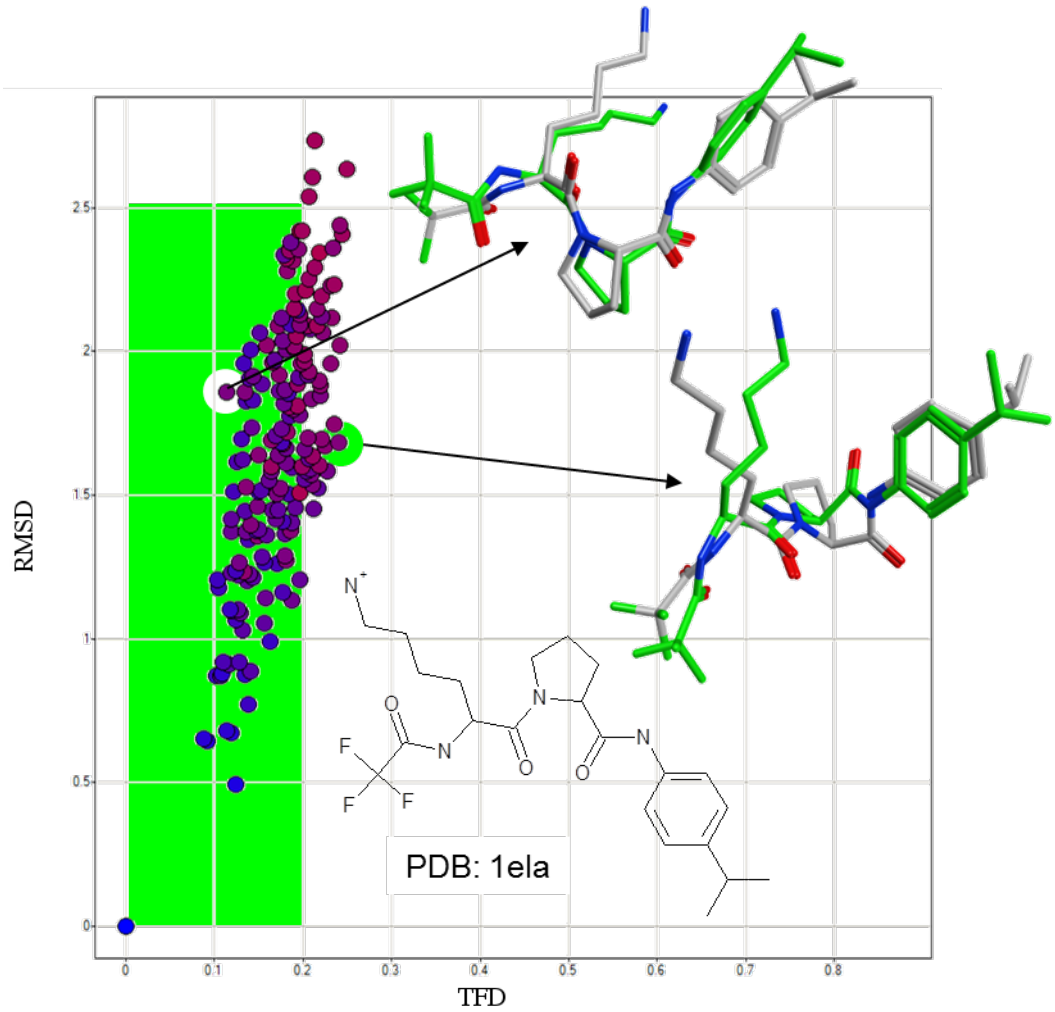


Abbildung 6.5.: TFD- und RMSD-Werte für den Liganden aus dem PDB-Komplex 1ela. Die Punkte sind nach TanimotoShape (0: rot – 1: blau) eingefärbt. Zusätzlich werden die Überlagerungen zweier ausgewählter Konformationen (grün gefärbte Strukturen) mit der bioaktiven Konformation (grau gefärbte Strukturen) gezeigt. $R^2 = 0,68$

liefert der TFD also nicht nur eine sinnvollere Bewertung als der RMSD, sondern auch eine sinnvollere Bewertung als der TanimotoShape.

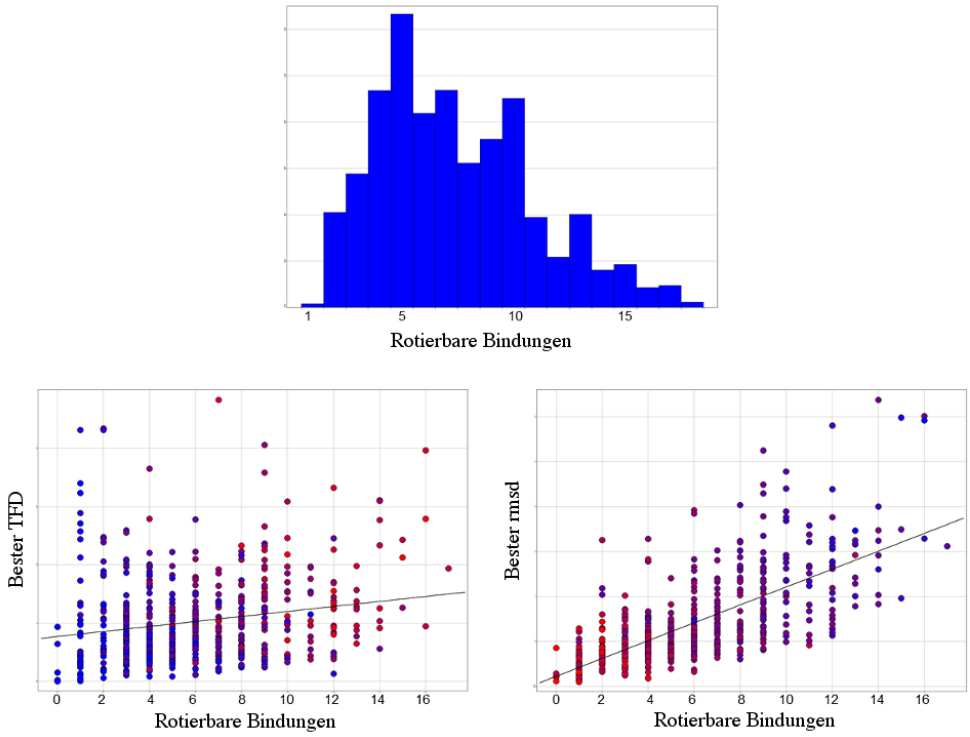


Abbildung 6.6.: Oben: Verteilung der Anzahl der rotierbaren Bindungen innerhalb des Datensatzes. Unten: bester durchschnittlicher TFD-Wert (links) und bester durchschnittlicher relativer RMSD-Wert (rechts) für jedes Molekül aus dem Datensatz, aufgeteilt nach der Anzahl der rotierbaren Bindungen. Die Einfärbung der Punkte erfolgte nach der Anzahl der Schweratome.

6.1.5. Einfluss der Datensatzzusammenstellung

Bei der Betrachtung der besten durchschnittlichen TFD- bzw. RMSD-Werte für jedes Molekül lässt sich erkennen, dass, im Gegensatz zum RMSD, der TFD nicht abhängig von der Anzahl der rotierbaren Bindungen ist (Abbildung 6.6; RMSD: $R^2 = 0,47$; TFD: $R^2 = 0,04$). Auch wenn die Daten statt nach der Anzahl der rotierbaren Bindungen nach der Anzahl der Schweratome aufgeteilt werden, ergibt sich, im Gegensatz zum RMSD, keine Abhängigkeit für den TFD. Hieraus lässt sich schließen, dass die Zusammenstellung des Datensatzes keinen Einfluss auf den TFD hat, wohingegen der RMSD durch Hinzufügen von Molekülen mit wenigen rotierbaren Bindungen und Entfernen von Molekülen mit vielen rotierbaren Bindungen manipuliert werden kann.

Tabelle 6.1.: Vergleich von normalisierten TFD-Werten mit relativen RMSD-Werten für drei unterschiedlich lange Kohlenstoffketten: *Kurz*, *Mittel*, *Lang*. Für die Berechnung der Werte wurde jeweils die erste Konformation als Referenzstruktur der entsprechenden Gruppe verwendet.

Molekül	Torsion-Fingerprint	TFD	RMSD
Kurz 1	(60.00 180.00 60.00)	0	0
Mittel 1	(60.00 180.00 180.00 180.00 60.00)	0	0
Lang 1	(60.00 180.00 180.00 180.00 180.00 180.00 60.00)	0	0
Kurz 2	(60.00 270.00 60.00)	0.50	0.54
Mittel 2	(60.00 270.00 90.00 90.00 60.00)	0.50	1.02
Lang 2	(60.00 270.00 270.00 270.00 270.00 270.00 60.00)	0.50	0.78
Kurz 3	(60.00 0.00 60.00)	0.99	0.92
Mittel 3	(60.00 0.00 0.00 0.00 60.00)	0.99	2.05
Lang 3	(60.00 0.00 0.00 0.00 0.00 0.00 60.00)	0.99	3.08

Die TFD- und RMSD-Werte für die drei unterschiedlich langen Kohlenstoffketten (siehe Abschnitt 5.1) sind in Tabelle 6.1 aufgelistet. Die TFD-Werte für die zweiten Konformationen jeder Gruppe sind genau gleich, wohingegen die RMSD-Werte zwischen den unterschiedlich langen Molekülen variieren. Die Variation der RMSD-Werte der dritten Konformationen ist sogar noch stärker als bei denen der zweiten Konformationen. Die Werte variieren dabei von 0,9 Å bis zu über 3 Å. Im Gegensatz dazu sind die TFD-Werte auch hier gleich. Dieses Beispiel zeigt, dass der TFD auf Grund seiner Normierung sehr gut geeignet ist, um Konformationen von Molekülen unterschiedlicher Größe und Flexibilität miteinander zu vergleichen.

6.1.6. Laufzeitverhalten

Die Berechnung der TFD-Werte erfolgte auf einem Rechner mit acht CPU-Kernen (Intel(R) Core(TM) i7-2600 CPU 3.40 GHz), 8 GB Arbeitsspeicher und SuSe Linux 11.3.

Abbildung 6.7 zeigt die akkumulierte Laufzeit der kompletten TFD-Berechnung inklusive TF-Berechnung und Gewichtung für alle Konformationen des Liganden aus dem PDB-Komplex 1k7f. Die Berechnung des TFD für die erste Konformation dauert etwa 18 Millisekunden. Für alle weiteren Konformationen dauert die Berechnung zwischen 0 und 1 Millisekunde. Bei jeder TFD-Berechnung wird zuerst der TF der Konformation berechnet

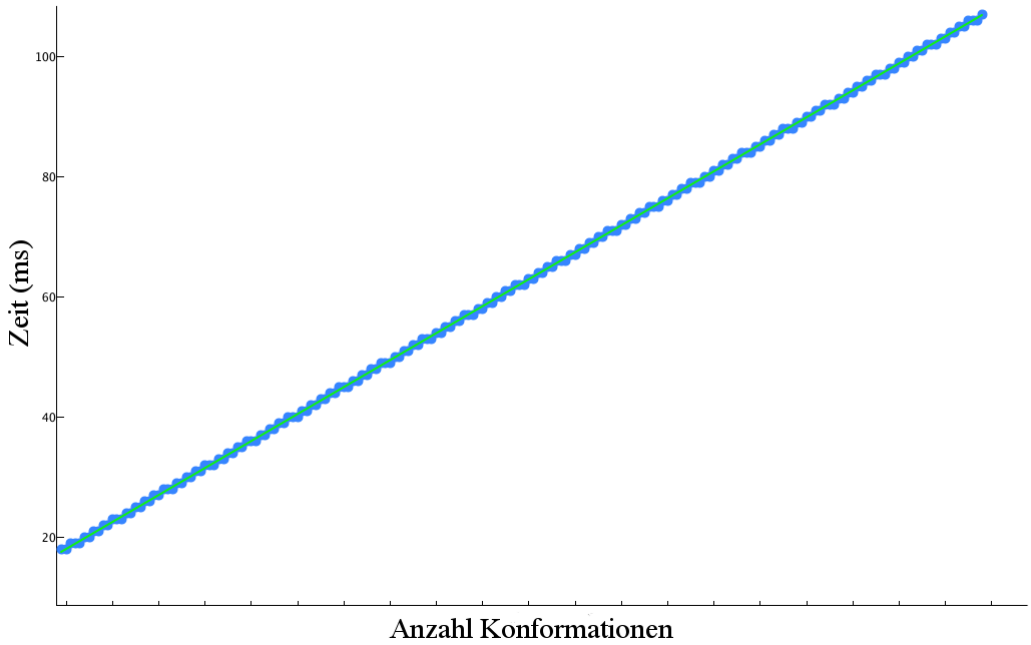


Abbildung 6.7.: Kumulative Laufzeit der TFD-Berechnung (inklusive TF-Berechnung und Gewichtung) für alle Konformationen des Liganden aus dem PDB-Komplex 1k7f.

und danach der TFD zur Referenzstruktur. Die Berechnung der Gewichtung erfolgt dagegen nur ein Mal am Anfang anhand der Referenzstruktur. Die hohe Laufzeit für die erste TFD-Berechnung ist somit auf die Berechnung der Gewichtung zurückzuführen. Die Laufzeit der TF- und TFD-Berechnung steigt ansonsten linear mit der Anzahl der Konformationen.

Abbildung 6.8 zeigt die Laufzeit der TF-Berechnung und Gewichtung in Abhängigkeit der Größe des TF für alle Liganden des Datensatzes. Das Diagramm zeigt deutlich, dass die Laufzeit nicht von der Größe des TF abhängt ($R^2 = 0,01$). Aufgrund der generell kurzen Laufzeiten im Millisekundenbereich (1–60 ms) ist davon auszugehen, dass sich die beobachteten Schwankungen in der Laufzeit in erster Linie aus der jeweiligen Systemsituation (Hintergrundprozesse etc.) ergeben. Die Laufzeit der TFD-Berechnung für jede einzelne Konformation des Datensatzes ist in Abbildung 6.9 gezeigt. Die Zeit für eine TFD-Berechnung liegt zwischen 0 und 2 Millisekunden. Auch hier lässt sich keine Abhängigkeit der Laufzeit von der TF-Größe feststellen ($R^2 = 0,06$).

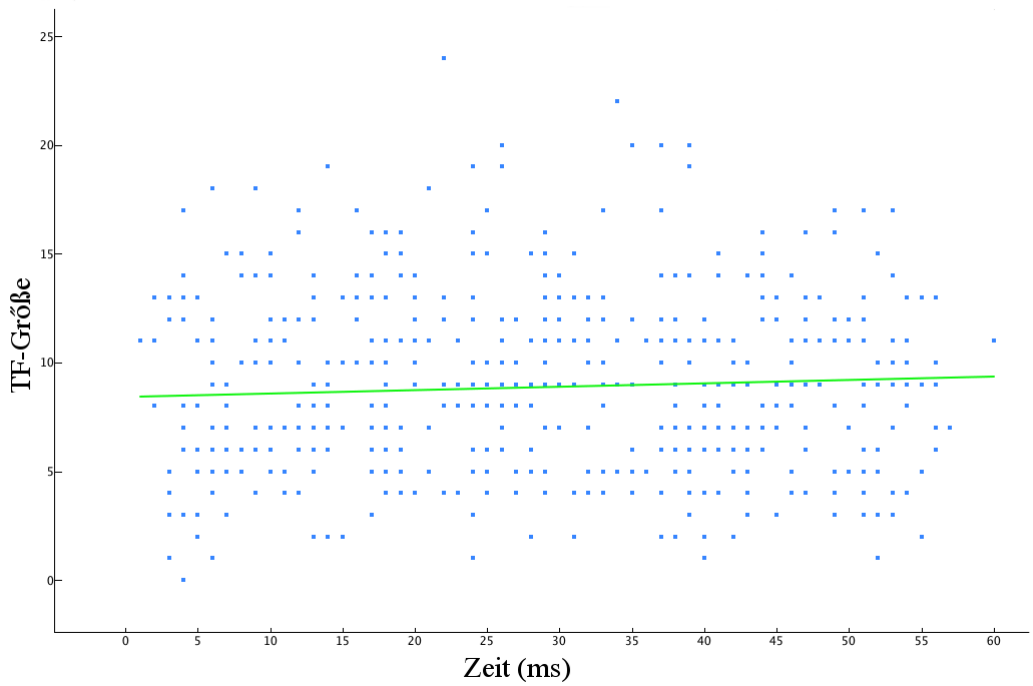


Abbildung 6.8.: Laufzeit TF-Berechnung inklusive Gewichtung in Abhängigkeit von der Größe des TF. $R^2 = 0,01$ (grüne Linie)

6.2. Torsionsbibliothek

Die Torsionsbibliothek besteht aus 97 generischen und 393 spezifischen Torsionssignaturen, die in sieben Hauptklassen (GG, CC, CN, CO, CS, NS, SS) und 35 Subklassen aufgeteilt sind. Die generische GG-Hauptklasse deckt zum einen alle generischen Torsionsmuster der anderen sechs Hauptklassen ab und enthält zum anderen generische Muster weiterer Heteroatom-Kombinationen. Die Torsionsatome 1 und 4 sind dabei jeweils mit der SMARTS-Primitive $*$ beschrieben. Die GG-Hauptklasse beinhaltet alle möglichen Kombinationen der Elemente C, N, O, S sowohl in der aliphatischen als auch in der aromatischen Version und zusätzlich noch in allen möglichen Valenzzuständen. Des weiteren enthält die Klasse noch generische Torsionsmuster, die nur die Hybridisierung beschreiben, (z.B. `[*:1]~[*^3:2]!@[*^2:3]~[*:4]`) und eine Subklasse für rotierbare Bindungen zwischen zwei aromatischen Ringsystemen (`[a:2]!@[a:3]`).

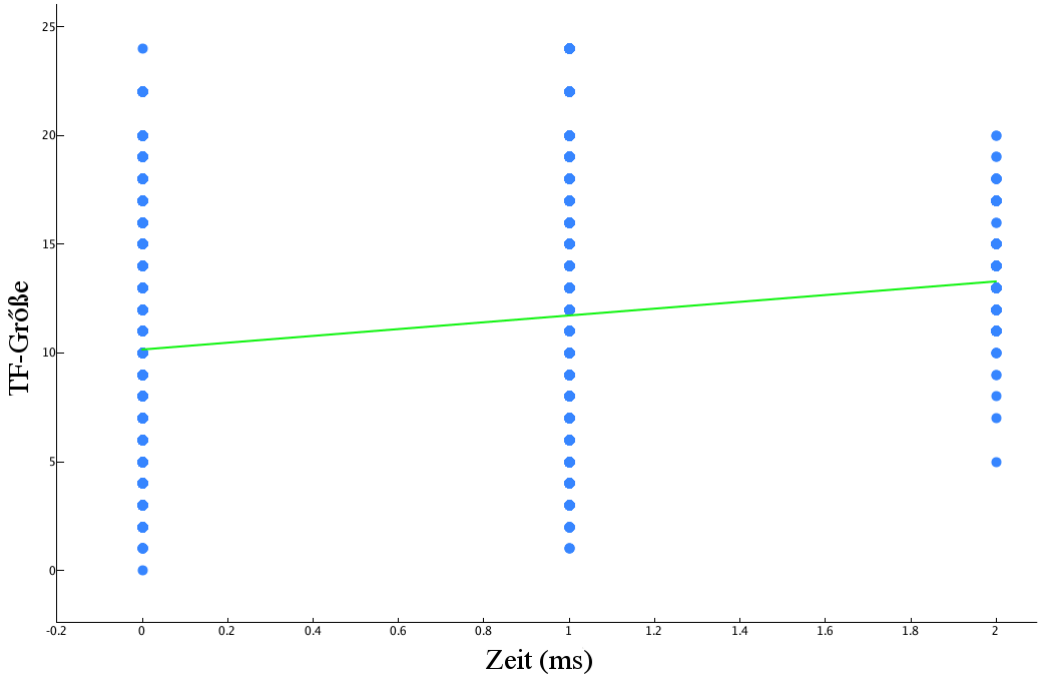


Abbildung 6.9.: Laufzeit TFD-Berechnung in Abhängigkeit von der Größe des TF. $R^2 = 0,06$ (grüne Linie)

6.2.1. Abdeckung des chemischen Raumes

Allen rotierbaren Bindungen (> 5 Millionen) der 100.000 Moleküle aus dem ChEMBL-Datensatz konnte eine Signatur aus der Torsionsbibliothek zugeordnet werden. Dabei wurden 96,3% der rotierbaren Bindungen eine spezifische und 3,7% eine generische Torsionssignatur zugewiesen. Die fünf am häufigsten benutzten spezifischen Torsionssignaturen beschreiben aliphatische Ketten, primäre Amide, Arylether und Benzylgruppen. Vier der zugehörigen Torsionshistogramme (siehe Abbildung 6.10) zeigen klare Peaks, was auf eine starke Präferenz für die entsprechende Konformation hindeutet. Einzig das Histogramm für die Benzylgruppen (5) zeigt keine klaren Peaks, was bedeutet, dass hier keine Präferenz für eine bestimmte Konformation vorliegt. Einige der sehr spezifischen Torsionssignaturen konnten keiner rotierbaren Bindung aus dem Datensatz zugeordnet werden. Diese Signaturen beschreiben zum Beispiel Arylether mit zweifacher ortho-Substitution in Kombination mit einem aliphatischen Ether-Kohlenstoffatom, dass an vier weitere Schweratome gebunden ist. Den 10 meistbenutzten generischen Torsionssignaturen wurden etwa 2% der rotierbaren Bindungen zugeordnet. Die an erster Stelle stehende Torsionssignatur (0,6% der

rotierbaren Bindungen) beschreibt ein aliphatisches sp2 hybridisiertes Kohlenstoffatom, welches an ein aliphatisches Stickstoffatom gebunden ist. Alle 10 Torsionshistogramme (siehe Abbildung 6.11) zeigen klare Präferenzen für bestimmte Konformationen, und wurden deshalb nicht weiter in spezifische Torsionssignaturen unterteilt. Aus den Ergebnissen lässt sich schließen, dass die 10 meistbenutzten generischen Torsionssignaturen zusammen mit den spezifischen Torsionssignaturen mehr als 98% des für die Medizinalchemie relevanten chemischen Raumes abdecken.

6.2.2. Vergleich von CSD- und PDB-Histogrammen

Allgemein lassen sich keine systematischen Unterschiede zwischen den aus der CSD und aus der PDB abgeleiteten Histogrammen erkennen. Dies bestätigt die Aussage, dass die Präferenz für eine bestimmte Konformation nur sehr selten durch die Packung im Kristall verändert wird [122,123]. CSD- und PDB-Histogramme weisen häufig die gleichen Peaks auf, allerdings sind die Peaks der PDB-Histogramme im allgemeinen breiter. Der Ligand spielt beim Einpassen der Komplex-Kristallstruktur in die Elektronendichte nur eine untergeordnete Rolle, so dass hier größere Abweichungen bei den Torsionswinkeln akzeptiert werden können. Da die Bestimmung der Toleranzen der Torsionssignaturen auf den schmaleren Peaks der CSD-Histogramme beruhen, werden bei der Analyse von Liganden aus PDB-Komplexen die Torsionswinkel häufig als *grenzwertig* oder *selten* klassifiziert. Dazu passt, dass etwa 66% der Strukturen aus dem PDB-Datensatz mindestens einen Torsionswinkel außerhalb des zweiten Toleranzbereichs aufweisen. Im Gegensatz dazu gilt dies nur für etwa 25% der Strukturen aus dem CSD-Datensatz.

Ein typisches Beispiel für PDB-Histogramme mit breiteren Peaks sind Liganden mit langen aliphatischen Kohlenstoffketten. Torsionswinkel entlang dieser Ketten haben Energieminima bei ±60° (windschiefe Konformation) und 180° (gestaffelte Konformation), was sich auch in den entsprechenden CSD- und PDB-Histogrammen widerspiegelt (Abbildung 6.12 a). Die PDB-Histogramme zeigen allerdings sehr breite, ineinander übergehende Peaks. Die Torsionswinkel zwischen den Peaks repräsentieren die energiereicheren ekliptischen Konformationen. Die Analyse von Strukturen mit mehreren ekliptischen Torsionswinkeln, wie zum Beispiel der Ligand aus dem PDB-Komplex 1cvu [124] (Abbildung 6.12 a) zeigt, dass schon mit einer leichten Minimierung des Liganden innerhalb der Bindetasche eine Konformation mit energetisch günstigeren Torsionswinkeln entsteht (orange gefärbte Struktur), die immer noch in die experimentell bestimmte Elektronendichte passt.

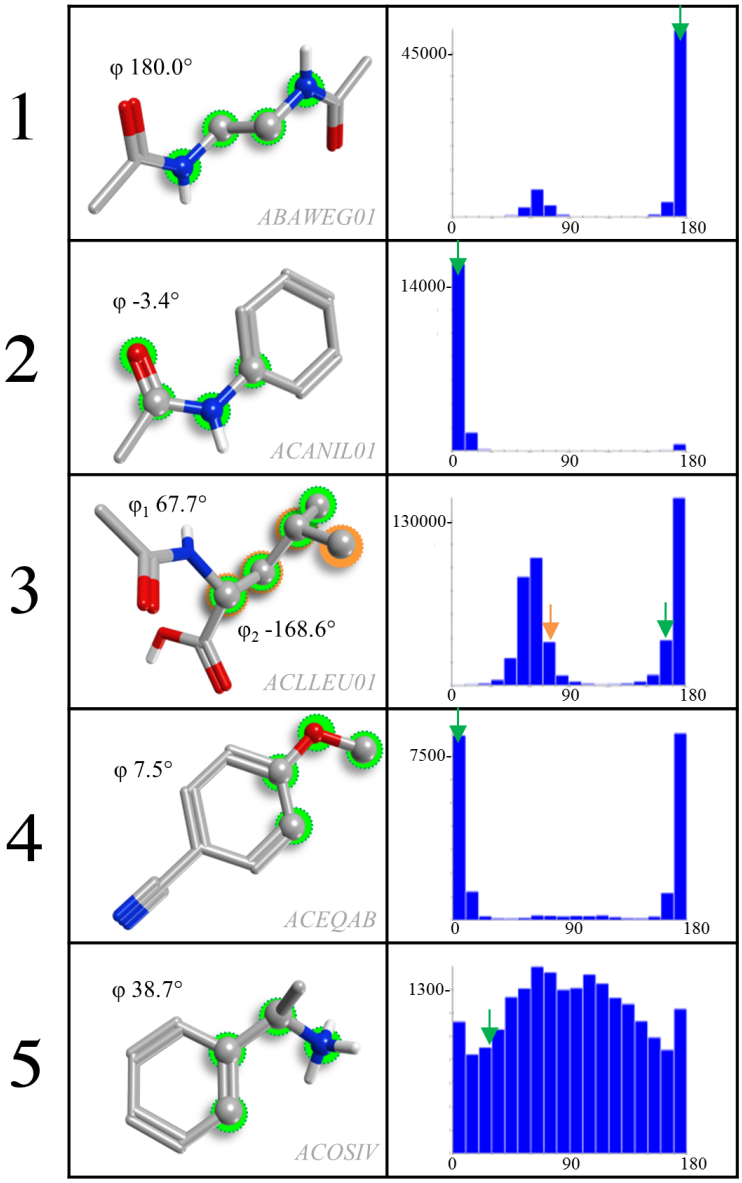


Abbildung 6.10.: Die fünf meistbenutzten spezifischen Torsionssignaturen beschreiben aliphatische Ketten (1 und 3), primäre Amide (2), Arylether (4) und Benzylgruppen mit Stickstoff- oder Sauerstoffsubstituenten (5). Für jede Signatur ist ein Beispielmolekül aus der CSD gezeigt. Diejenigen vier Atome, die den entsprechenden Torsionswinkel beschreiben, sind im Molekül hervorgehoben und der gemessene Wert ist durch einen Pfeil im Torsionshistogramm angezeigt.

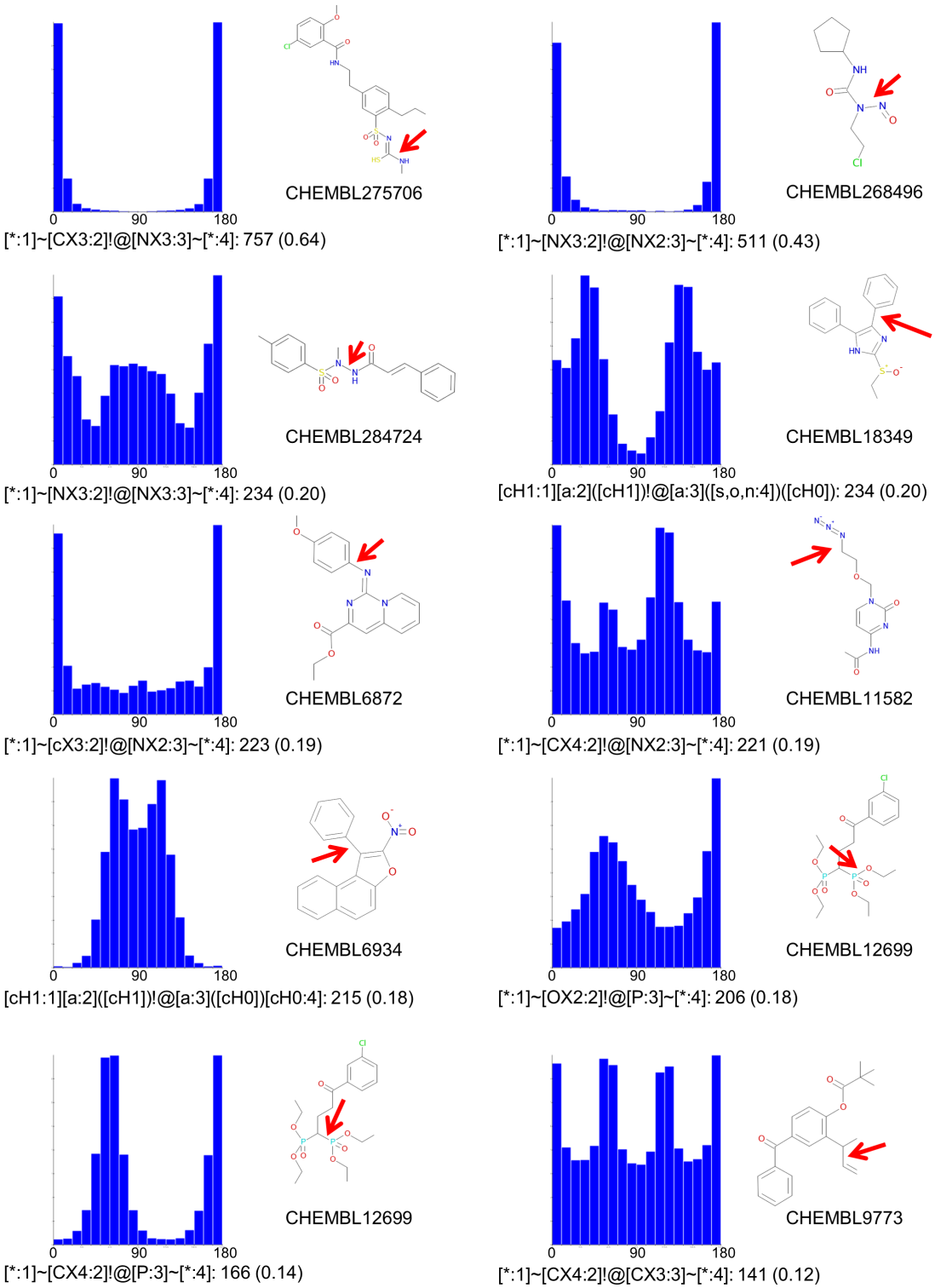


Abbildung 6.11.: Die 10 meistbenutzten generischen Torsionssignaturen. Für jede Signatur ist die Häufigkeit angegeben und ein Beispielmolekül aus dem ChEMBL-Datensatz. Der entsprechende Torsionswinkel im jeweiligen Molekül ist durch einen roten Pfeil hervorgehoben.

Ein weiteres Beispiel für breitere Peaks in PDB-Histogrammen sind Ester-Konformationen. Die Alkoxygruppe und die Carbonylgruppe nehmen dabei immer eine cis-Konfiguration ein, was sich auch in dem dazugehörigen CSD-Histogramm zeigt (Abbildung 6.12 b). Die Verteilung im PDB-Histogramm zeigt zwar die gleiche Präferenz für den 0°-Torsionswinkel, aber der Peak ist wesentlich breiter und geht sogar noch über 90° hinaus. Beim Liganden aus dem PDB-Komplex 3rxw [125] hat die Ester-Gruppe einen Torsionswinkel von 73°. Der Komplex hat zwar eine hohe Auflösung (1,26 Å), aber um den Ester-Substituenten ist keine Elektronendichte definiert. Eine genaue Analyse des Liganden zeigt, dass dieser Teil im Lösungsmittel liegt und nicht mit dem Protein interagiert. Es ist daher anzunehmen, dass die Einpassung des Liganden in die Elektronendichte nicht korrekt ist.

Genau wie Ester zeigen auch primäre Amide eine stark ausgeprägte Präferenz für eine planare Konformation mit einem Torsionswinkel von 0°, was sowohl im CSD- als auch im PDB-Histogramm zu beobachten ist (Abbildung 6.12 c). Im Gegensatz zum Ester weisen allerdings beide Histogramme etwa gleich schmale Peaks auf, was eine direkte Konsequenz aus der Benutzung typischer Kraftfelder zur Einpassung der Koordinaten in die Elektronendichte sein könnte. Die mit MMFF94 [73] berechnete Energiedifferenz für das 90°-Amid ist zum Beispiel etwa doppelt so hoch wie die des Esters. Am Liganden aus dem PDB-Komplex 3ke1 [126] lässt sich beispielhaft zeigen, dass viele der aus der Ebene gedrehten Amid-Konformationen in die bevorzugte Konformation mit dem 0°-Torsionswinkel überführt werden können, ohne dabei die experimentell bestimmte Elektronendichte zu verletzen.

Die vorherigen drei Beispiel veranschaulichen nicht nur den Unterschied zwischen den CSD- und PDB-Histogrammen, sondern zeigen auch, wie sich einzelne Torsionswinkel mit Hilfe der CSD-Histogramme optimieren lassen. Die nächsten beiden Beispiele sollen nun zeigen, wie Liganden aus PDB-Komplexen mit Hilfe der Torsionsbibliothek aufbereitet werden können.

Der Ligand aus dem PDB-Komplex 3tv7 [127] hat sieben rotierbare Bindungen, von denen drei als *selten* klassifiziert werden (siehe Abbildung 6.13). Die Amid-Bindung der Harnstoffgruppe steht fast senkrecht zur benachbarten Bindung (1), die benachbarte CN-Bindung hat eine ekliptische Konformation (2) und die terminale Methoxy-Gruppe ist aus der Ebene des angrenzenden Phenylrings gedreht (3). Die gemessenen Torsionswinkel der drei Bindungen liegen außerhalb der Peaks und Toleranzbereiche der zugeordneten CSD-Histogramme (siehe rote Pfeile in Abbildung 6.13). Laut der Annotation in der PDB-Struktur interagiert das NH der Harnstoffgruppe über eine Wasserstoffbrückenbindung mit der Seitenkette von Asp 216. Eine

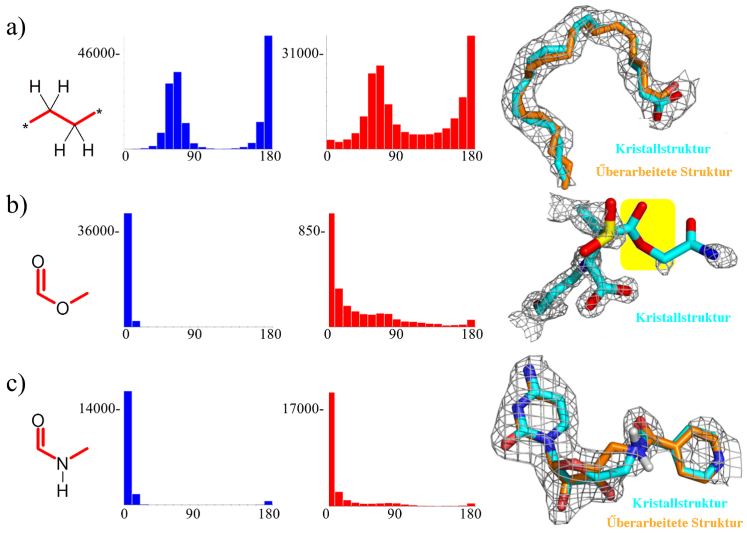


Abbildung 6.12.: Drei Beispiel-Torsionssignaturen für den Vergleich von CSD- und PDB-Histogrammen. Jede Signatur ist durch ein Beispielmolekül aus der PDB veranschaulicht. a) Flexible aliphatische Kette, Ligand aus PDB-Komplex 1cvu (Auflösung: 2,4 Å). b) Ester, Ligand aus PDB-Komplex 3rxw (Auflösung: 1,26 Å). c) Primäre Amide, Ligand aus PDB-Komplex 3ke1 (Auflösung: 2,4 Å). „Verzerrte" Konformationen (blau gefärbte Kristallstrukturen) können in energetisch günstigere Konformationen (orange gefärbte Strukturen) umgewandelt werden, ohne dabei die experimentell bestimmte Elektronendichte zu verletzen.

genauere Überprüfung des Liganden zeigt allerdings, dass alle Parameter außerhalb des Bereichs für eine Wasserstoffbrückenbindung liegen:

- Abstand N...O: 3,4 Å, optimaler Abstand: 2,8–3,1 Å
- Winkel N-H-O: 129° , optimaler Winkel: >150°
- NH etwa 80° aus der Ebene des sp2-Akzeptor-Atoms gedreht, optimal: in der Ebene

Die drei Torsionswinkel lassen sich manuell so anpassen, dass sie hinterher in den Peak-Bereichen der Torsionshistogramme liegen (siehe blaue Pfeile in Abbildung 6.13) und der Ligand nach wie vor in die experimentell bestimmte Elektronendichte passt. Das NH der Harnstoffgruppe interagiert weiterhin nicht mit dem Protein, ist jetzt aber dem Lösungsmittel zugänglich. Der Phenylring verändert seine Lage dabei nicht, so dass die an dieser Stelle wichtige hydrophobe Interaktion mit dem Protein bestehen bleibt. Die einzig signifikante Änderung ist die Orientierung der Methylgruppe am

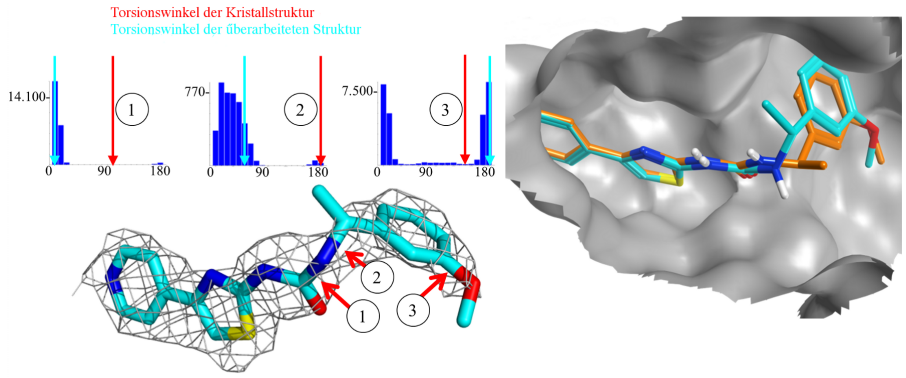


Abbildung 6.13.: Drei der Bindungen des Liganden aus dem PDB-Komplex 3tv7 wurden mit der Torsionsbibliothek als *selten* klassifiziert (1–3, rote Pfeile). Die gemessenen Torsionswinkel sind in den entsprechenden Torsionshistogrammen durch rote Pfeile markiert. Durch Rotation gelangen alle Torsionswinkel in den Peak-Bereich (blaue Pfeile in den Torsionshistogrammen). Die so überarbeitete Struktur passt dabei nach wie vor in die experimentell bestimmte Elektronendichte.

tertiären Kohlenstoffatom. Allerdings wurde genau an dieser Stelle keine Elektronendichte gemessen.

Der Ligand aus dem PDB-Komplex 1gwx [128] weist sechs als *selten* klassifizierte rotierbare Bindungen auf (siehe Abbildung 6.14). Durch manuelle Anpassung des Liganden können alle sechs Torsionswinkel so eingestellt werden, dass sie innerhalb des ersten Toleranzbereichs der zugeordneten Torsionssignatur liegen und der Ligand dabei nicht die Elektronendichte verletzt. Die beiden Torsionswinkel am tertiären Amid ($\varphi 1$ und $\varphi 2$ in Abbildung 6.14) zeigen einen Torsionswinkel von etwa 120°, welcher außerhalb der zweiten Toleranz (90°±20°) liegt. Die beiden Winkel wurden manuell auf 78° und 108° eingestellt. Obwohl die Anpassungen an dieser Stelle nicht besonders groß sind, kann der gleiche Effekt mit alternativen Methoden nicht erreicht werden. Eine Kraftfeldminimierung würde hier zum Beispiel zu einer antiperiplanaren Konformation für mindestens einen der Stickstoffsubstituenten führen.

Das nächste Beispiel zeigt einen der wenigen systematischen Unterschiede zwischen CSD- und PDB-Histogrammen. Die beiden Torsionshistogramme in Abbildung 6.15 zeigen die Verteilung von Torsionswinkeln in acetylierten Aminopyrimidinen. Das CSD-Histogramm zeigt sowohl die cis- also auch die trans-Konfiguration für die Amidbindung, wohingegen das PDB-Histogramm nur die trans-Konfiguration zeigt. Eine Analyse der dem

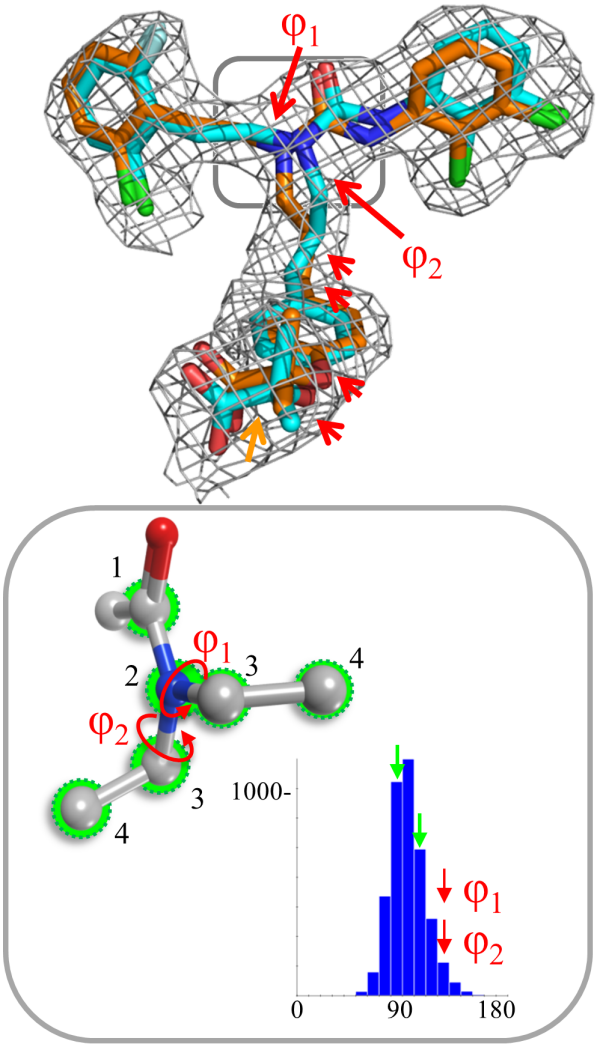


Abbildung 6.14.: Oben: Der gebundene Ligand aus dem PDB-Komplex 1gwx (orange gefärbte Struktur) wurde manuell überarbeitet (blau gefärbte Struktur). Sechs Bindungen der Kristallstruktur wurden als *selten* klassifiziert (rote Pfeile). Die nur leicht überarbeitete Struktur passt immer noch in die experimentell bestimmte Elektronendichte. Die terminale Säuregruppe (oranger Pfeil) wurde unverändert gelassen, um die Interaktion mit dem Protein beizubehalten. Unten: Torsionshistogramm für Substituenten des Typs -CH2R am sekundären Amid.

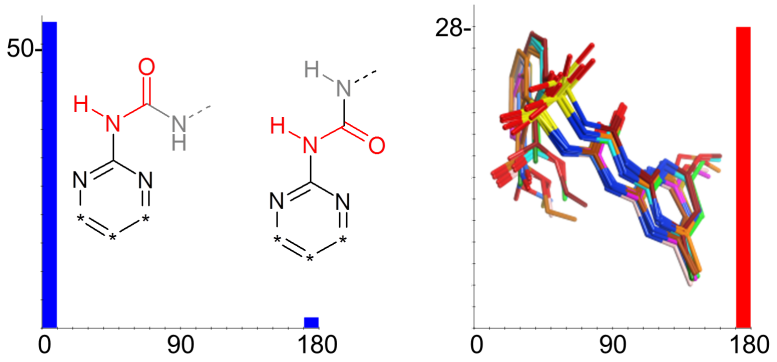


Abbildung 6.15.: Unterschiede in der Torsionswinkelverteilung bei acetylierten Aminopyrimidinen. Links: CSD-Histogramm, rechts: PDB-Histogramm. Die dem PDB-Histogramm zugrunde liegenden Liganden binden alle an das gleiche Enzym (Acetohydroxyacid-Synthase).

PDB-Histogramm zugrunde liegenden Liganden zeigt, dass alle Liganden eine ähnliche Struktur haben und an das gleiche Enzym (Acetohydroxyacid-Synthase) binden. Das PDB-Histogramm zeigt also nur eine sehr einseitige Torsionswinkelverteilung und kein allgemeines Bild.

6.2.3. Anwendungsbeispiele

Die folgenden drei Beispiele sollen das praktische Arbeiten mit der Torsionsbibliothek veranschaulichen.

Beispiel: Faktor-Xa

Abbildung 6.16 zeigt zwei Faktor-Xa-Inhibitoren **1** und **2** [129]. Die beiden Inhibitoren unterscheiden sich ausschließlich in einer einzelnen Methyl-Gruppe. Dabei weist Faktor-Xa-Inhibitor **1** eine schlechtere Bindungsaffinität (20nM) als Faktor-Xa-Inhibitor **2** (5,3nM) auf. Aus dem Bindungsmodus von Faktor-Xa-Inhibitor **2** im PDB-Komplex 1ksn [130] lässt sich erkennen, dass die zusätzliche Methyl-Gruppe dem Lösungsmittel zugänglich ist und keine Wechselwirkung mit dem Protein ausbildet. Der Gewinn an Bindungsaffinität von Faktor-Xa-Inhibitor **2** im Vergleich mit **1** lässt sich also nicht durch zusätzliche Wechselwirkungen in der Bindestelle erklären. Mit Hilfe der Torsionsbibliothek lassen sich jedoch energetische Unterschiede

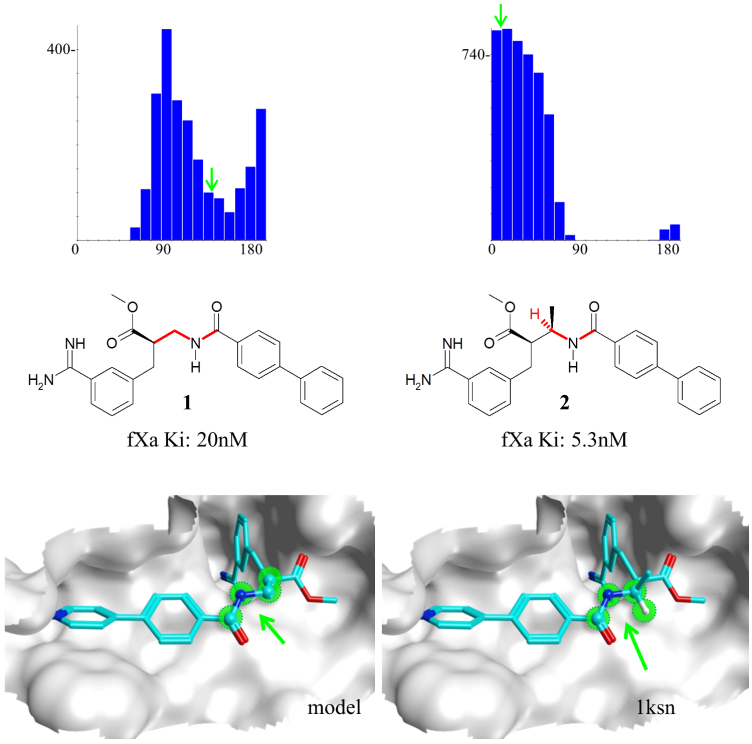


Abbildung 6.16.: Anwendungsbeispiel Faktor-Xa-Inhibitoren **1** und **2**. Oben: aus der CSD abgeleitete Torsionshistogramme für die durch die rot markierten Bindungen definierten Torsionswinkel. Unten links: modellierte Konformation von Faktor-Xa-Inhibitor **1**. Unten rechts: Konformation von Faktor-Xa-Inhibitor **2** im PDB-Komplex 1ksn. Die in beiden Konformationen gemessenen Torsionswinkel sind durch einen grünen Pfeil im entsprechenden Torsionshistogramm angedeutet.

in den Konformationen der beiden Inhibitoren identifizieren. Der Faktor-Xa-Inhibitor **1** weist eine Konformation auf, deren Torsionswinkel genau zwischen zwei Peaks im dazugehörigen Torsionshistogramm liegt (siehe linkes Histogramm in Abbildung 6.16), und somit energetisch ungünstig ist. Der Faktor-Xa-Inhibitor **2** weist hingegen eine Konformation auf, deren Torsionswinkel direkt im Peak des dazugehörigen Histogramms liegt (siehe rechtes Histogramm in Abbildung 6.16), was energetisch günstiger ist und somit zu einer erhöhten Bindungsaffinität gegenüber **1** führt.

Beispiel: Arylether

Das in der GG-Hauptklasse einsortierte generische Torsionsmuster für Arylether `[*:1]~[cX3:2]!@[OX2:3]~[*:4]` beschreibt ein Sauerstoffatom, das mit genau zwei weiteren Atome verbunden ist, wobei eines davon ein aromatisches Kohlenstoffatom ist. Das dazugehörige, aus der CSD abgeleitete Torsionshistogramm (Abbildung 6.17 oben), zeigt zwei klare Peaks bei 0°und 180°, was bedeutet, dass der Ether-Substituent bevorzugt in einer Ebene mit dem aromatischen Ring liegt. Das Histogramm zeigt aber noch einen weiteren, nur schwach ausgeprägten Peak bei 90°. Es gibt also Strukturen, bei denen der Ether-Substituent im rechten Winkel zur Ringebene liegt. Aus dem Torsionshistogramm der generischen Signatur für Arylether lässt sich nicht erkennen, ob sterische oder elektronische Effekte von benachbarten Substituenten für die einzelnen Peaks verantwortlich sind. Abbildung 6.17 zeigt acht spezifische Arylether-Torsionsmuster (a–h) für unterschiedliche aromatische ortho-Substituenten und für Substituenten, die einen Einfluss auf die Konformation des Arylethers haben. Die in der Hierarchie der Torsionsbibliothek ganz unten stehenden Torsionsmuster f–h beschreiben den allgemeinen Fall für einen, zwei oder keinen ortho-Substituenten. Die hierarchische Reihenfolge der Torsionsmuster untereinander ist irrelevant, da sie sich gegenseitig ausschließen. Die Torsionshistogramme zeigen, dass bei zweifacher ortho-Substitution der Ether-Substituent im rechten Winkel zur Ringebene steht. Bei einfacher ortho-Substitution liegt der Ether-Substituent dagegen in der Ringebene und zeigt in die entgegengesetzte Richtung des ortho-Substituenten. Die Torsionsmuster d und e sind etwas spezifischer und beschreiben eine ortho-Substitution durch Sauerstoffatome. Die dazugehörigen Histogramme zeigen die gleichen Peaks wie die Histogramme in f und h und sind daher redundant. Sie bieten allerdings zusätzliche Informationen. Ein Vergleich von d und f zeigt, dass die gegenseitige Abstoßung der freien Elektronenpaare der beiden benachbarten Ether-Sauerstoffe die Präferenz für einen bestimmten Torsionswinkel nicht verändert. Es macht also in diesem Fall keinen Unterschied, ob der ortho-Substituent ein Sauerstoffatom, oder ein beliebiges anderes Schweratom ist. In einigen Fällen sind die Peaks im Torsionshistogramm zwar an der gleichen Stelle, sie unterscheiden sich aber in ihrer Breite. Ein Beispiel dafür ist das Torsionsmuster a im Vergleich zu den Mustern e und h. Ein sehr sperriger Ether-Substituent sorgt für eine striktere Präferenz für den 90°-Torsionswinkel. Dementsprechend unterscheiden sich die zugewiesenen Toleranzen: 20° und 30° für a, 30° und 40° für e und 30° und 35° für h. Das Torsionshistogramm für das Pyridin-Derivat (c) zeigt einen einzelnen Peak bei 180°, was darauf hindeutet, dass die freien Elektronenpaare des Sauerstoffatoms und Stickstoffatoms eine

gegenseitige Abstoßung vermeiden. Bei Pyrimidin-Derivaten (b) lässt sich die gegenseitige Abstoßung hingegen nicht vermeiden. Das dazugehörige Torsionshistogramm zeigt eine klare Präferenz für die Anordnung des Ether-Substituenten in einer Ebene mit dem Ringsystem. Die Beispiele zeigen, dass sich die Präferenzen für einen bestimmten Torsionswinkel nicht immer durch die Berücksichtigung von sterischen oder elektronischen Effekten erklären lassen.

Beispiel: Anilide und Benzamide

Das folgende Beispiel zeigt, genau wie das vorherige Beispiel, dass eine Veränderung der chemischen Umgebung zu einer nicht gleich offensichtlichen Veränderung der Präferenz für einen bestimmten Torsionswinkel führt. Die Torsionshistogramme der beiden oberen generischen Torsionssignaturen in Abbildung 6.18 zeigen, dass die Substituenten an Anilin-Stickstoffatomen und an Systemen mit konjugierten Doppelbindungen häufig in einer Ebene mit dem Ringsystem liegen. Es gibt allerdings auch eine signifikante Anzahl an Strukturen mit einem Torsionswinkel zwischen 0° und 180°. Die Torsionshistogramme der generischen Signaturen für Anilide und Benzamide (Abbildung 6.18 d) zeigen eine ähnliche Häufigkeitsverteilung, das Histogramm für die Benzamide weist allerdings zwei zusätzliche benachbarte Peaks bei 0° und 180° auf. Die Torsionshistogramme der spezifischeren Torsionssignaturen (c) zeigen keine Konformation, in der die Substituenten im rechten Winkel zum Ring liegen und sie verdeutlichen die leicht unterschiedliche Lage der Peaks. Während die Anilid-Substituenten in einer Ebene mit dem Ring liegen, liegen die Substituenten der Benzamide etwas außerhalb der Ringebene.

Interessant ist der unterschiedliche Einfluss von einfacher ortho-Substitution durch Fluor- oder Chloratome auf die Konformation von Aniliden und Benzamiden. Aus den Torsionshistogrammen (Abbildung 6.18 b) lässt sich erkennen, dass die Fluoratome in einer Ebene und auf der selben Seite wie das Amid-Wasserstoffatom liegen. Dies gilt sowohl für die Anilide als auch für die Benzamide. Ein Unterschied zeigt sich bei der ortho-Substitution durch Chloratome (Abbildung 6.18 a). Bei den Aniliden tendieren die Chloratome dazu, in einer Ebene und auf der selben Seite wie das Amid-Wasserstoffatom zu liegen. Dies lässt sich für die Benzamide nicht beobachten. Dort ist das Amid-Stickstoffatom stark aus der Ringebene herausgedreht. Vier der Torsionshistogramme in Abbildung 6.18 (a) und (b) wurden zusätzlich noch mit berechneten Torsionsprofilen (Maestro 9.1, Relaxed Coordinate Scan, HF,

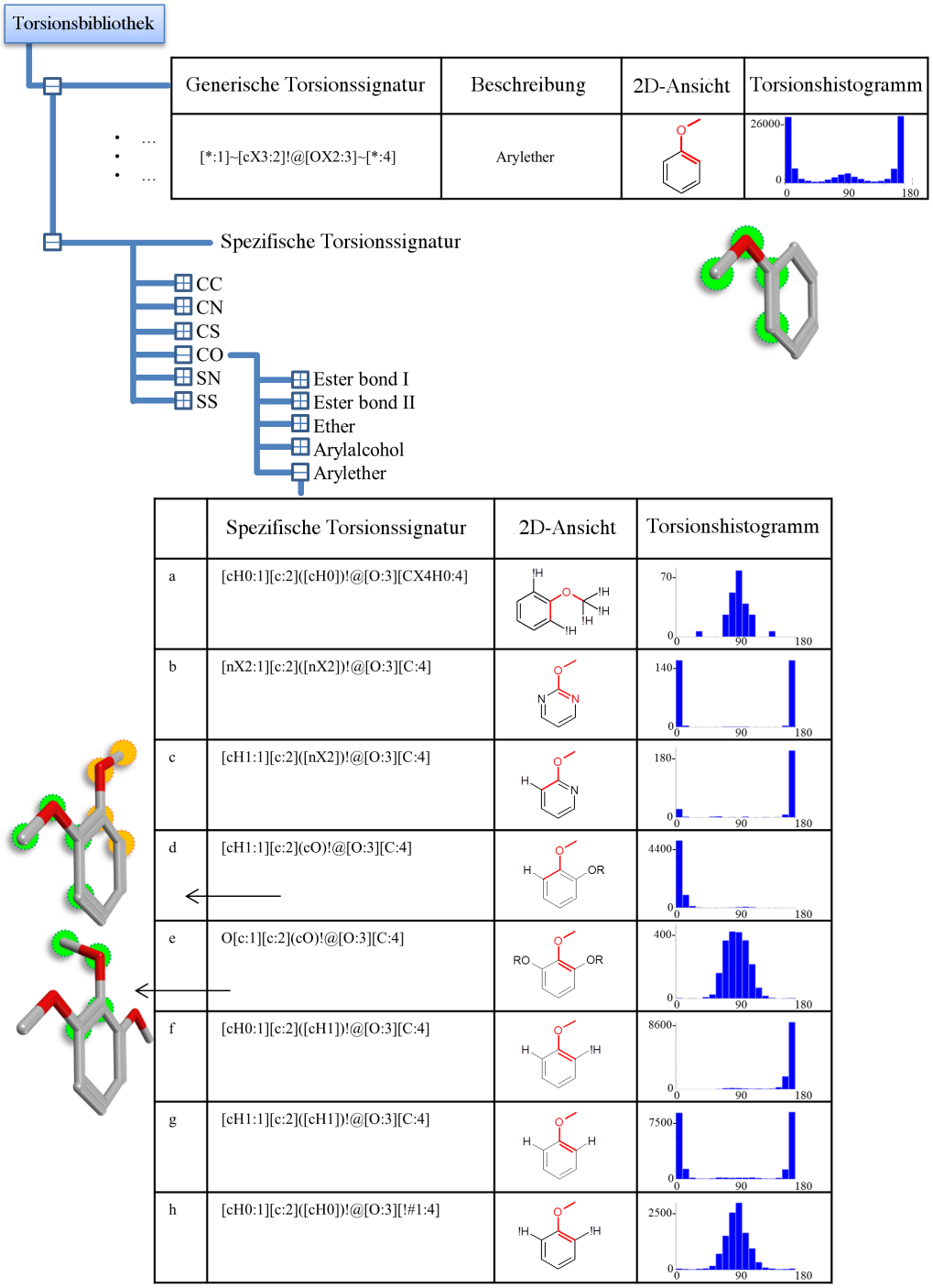


Abbildung 6.17.: Arylether: das obere generische Torsionsmuster beschreibt ein Sauerstoffatom, das mit genau zwei weiteren Atome verbunden ist, eines davon ein aromatisches Kohlenstoffatom. Die unter der CO-Hauptklasse einsortierte Subklasse enthält spezifische Torsionssignaturen für Arylether a–h. Die abgebildeten Torsionshistogramme wurden aus der CSD abgeleitet.

6-31G) überlagert. Die berechneten lokalen Minima stimmen mit den Torsionshistogrammen überein, was für die Benutzung der CSD-Histogramme zur Analyse von Torsionswinkeln spricht.

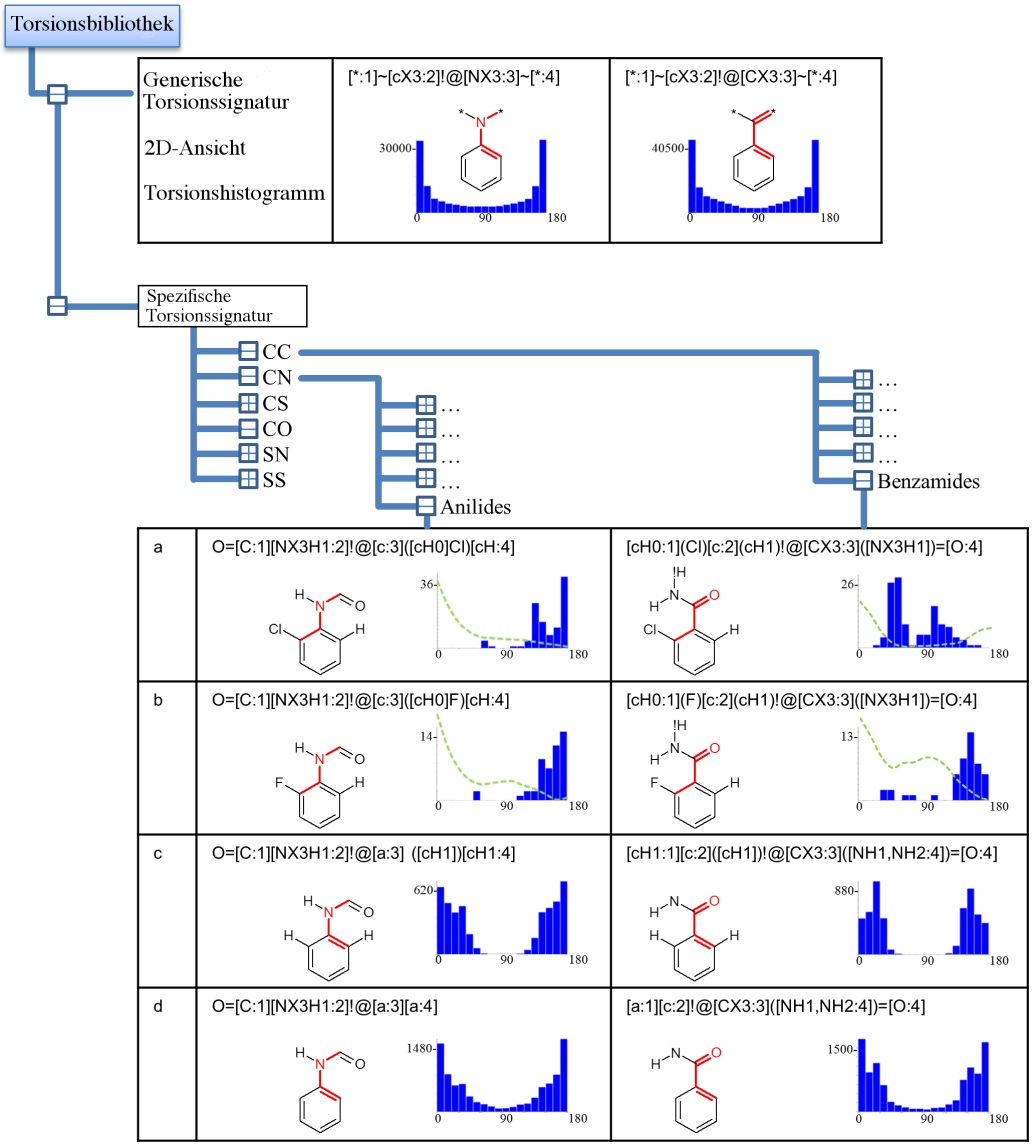


Abbildung 6.18.: Anilide und Benzamide: die beiden oberen generischen Torsionsmuster beschreiben ein Stickstoff-, bzw. Kohlenstoffatom, das mit genau drei weiteren Atomen verbunden ist, eines davon ein aromatisches Kohlenstoffatom. Die Torsionshistogramme der spezifischeren Torsionsmuster aus der Subklasse für Anilide bzw. Benzamide zeigen mehr Details zu den Präferenzen der Torsionswinkel (a–d). Die Torsionshistogramme in a und b enthalten jeweils zusätzliche Torsionsprofile aus QM-Berechnungen.

6.3. CONFECT

Der folgende Abschnitt beschreibt die Ergebnisse aus der Evaluierung der Konformationsgenerierung. Die verwendete Vorgehensweise ist in Abschnitt 5.3 beschrieben.

6.3.1. Reproduktion der bioaktiven Konformation

Perola100-Datensatz

Zur Evaluierung von CONFECT wurden zuerst die Ergebnisse für den Perola100-Datensatz, die mit den unterschiedlichen Qualitätsstufen erzielt wurden, miteinander verglichen. Was die Reproduktion der bioaktiven Konformation angeht, unterscheiden sich die Ergebnisse der unterschiedlichen Qualitätsstufen nur minimal (siehe Tabelle 6.2 und Abbildung 6.19 oben). Lediglich die Qualitätsstufe 1 schneidet bei den Konformationen $< 0,5$Å und $< 1,0$Å schlechter ab, als die anderen Stufen. Anscheinend ist es also nicht ausreichend, die Torsionswinkel nur anhand der Peaks, ohne Berücksichtigung der Toleranzen, einzustellen. Bei allen Qualitätsstufen konnte für mehr als 60% der Moleküle eine Konformationen mit einem RMSD $< 1,0$Å und für fast 80% der Moleküle eine Konformation mit einem RMSD $< 1,5$Å zur bioaktiven Konformation generiert werden.

Die durchschnittliche Größe des generierten Ensembles liegt zwischen 47 (Stufe 1) und 284 (Stufe 5) Konformationen, wobei die Ensemblegröße mit zunehmender Qualitätsstufe ansteigt (siehe Tabelle 6.2 und Abbildung 6.19 unten links). Dies entspricht den Erwartungen, da mit zunehmender Qualitätsstufe mehr Winkel pro rotierbarer Bindung (Stufe 1: 1 Winkel; Stufe 2 und 21: 3 Winkel; Stufe 3, 4 und 41: 5 Winkel; Stufe 5: 9 Winkel) eingestellt werden und die maximale Anzahl zu generierender Konformationen stufenweise hoch gesetzt wird (Stufe 1: 250 Konformationen; Stufe 2 und 21: 500 Konformationen; Stufe 3, 4 und 41: 1000 Konformationen; Stufe 5: 2000 Konformationen; siehe auch Abschnitt 4.3.3). Die durchschnittliche Laufzeit steigt, wie erwartet, ebenfalls mit zunehmender Qualitätsstufe an und liegt zwischen 0,7s (Stufe 1) und 5,9s (Stufe 2; siehe Tabelle 6.2 und Abbildung 6.19 unten links). Unter Berücksichtigung der Laufzeit und der Ensemblegröße erzielt CONFECT mit der Qualitätsstufe 21 die besten Ergebnisse.

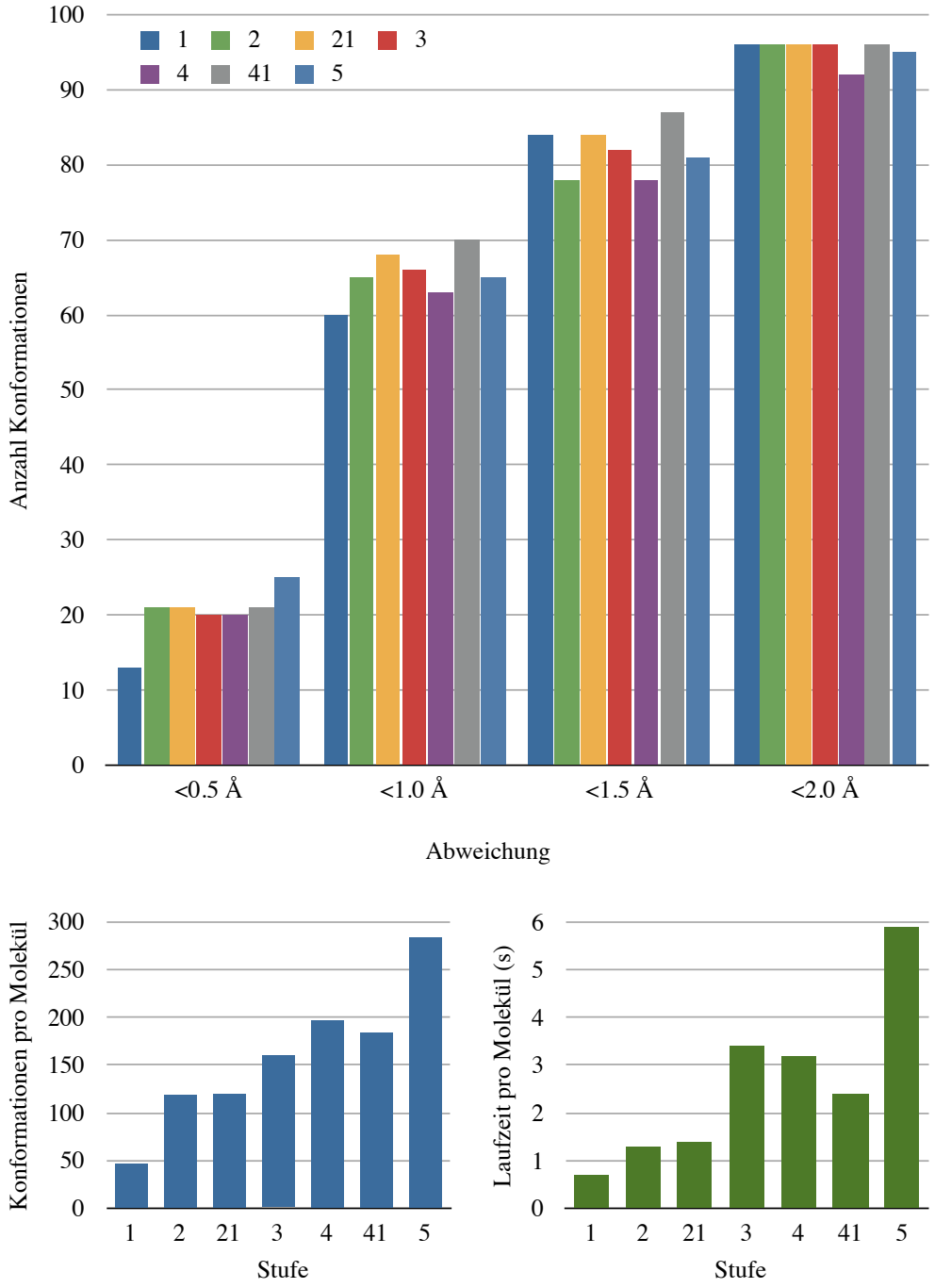


Abbildung 6.19.: CONFECT-Ergebnisse für die unterschiedlichen Qualitätsstufen. Oben: Reproduktion der bioaktiven Konformation. Unten links: durchschnittliche Größe des Konformationsensembles. Unten rechts: durchschnittliche Laufzeit.

Tabelle 6.2.: CONFECT-Ergebnisse für die unterschiedlichen Qualitätsstufen. „Konf pro Mol“ gibt die durchschnittliche Anzahl der generierten Konformationen und „Zeit pro Mol“ die durchschnittliche Laufzeit an.

Stufe	gefundene bioaktive Konformationen (%)				Konf pro Mol	Zeit pro Mol (s)
	$\leq$ 0.5 Å	$\leq$ 1.0 Å	$\leq$ 1.5 Å	$\leq$ 2.0 Å		
1	13	60	84	96	47	0,7
2	21	65	78	96	119	1,3
21	21	68	84	96	120	1,4
3	20	66	82	96	160	3,4
4	20	63	78	92	197	3,2
41	21	70	87	96	184	2,4
5	25	65	81	95	284	5,9

Iridium-Datensatz

Die Strukturen aus dem Iridium-Datensatz sind von höherer experimenteller Qualität als die Strukturen des Perola100-Datensatzes. Ein Vergleich der CONFECT-Ergebnisse (Qualitätsstufe 21, Standardeinstellungen) der beiden Datensätze zeigt, dass CONFECT wesentlich besser in der Lage ist, die bioaktive Konformation für die Moleküle des Iridium-Datensatzes zu reproduzieren (siehe Abbildung 6.20). Dies zeigt sich vor allem für die RMSD-Werte $< 0,5$Å. Beim Iridium-Datensatz wurden doppelt so viele Konformationen mit einem RMSD <0,5Å zur bioaktiven Konformation generiert wie beim Perola100-Datensatz. Insgesamt konnten für mehr als 90% der Moleküle des Iridium-Datensatzes Konformationen mit einem RMSD $< 1,5$Å zur bioaktiven Konformation generiert werden. Die besseren Ergebnisse für den Iridium-Datensatz liegen wahrscheinlich an der höheren Qualität der Strukturen und einer niedrigeren durchschnittlichen Anzahl rotierbarer Bindungen (Iridium: 5,3; Perola100: 6,5).

Keinen nennenswerten Unterschied gibt es dagegen zwischen den Ergebnissen von Iridium deposited und Iridium refined. Auch das liegt wahrscheinlich an der besseren Qualität der Strukturen. Die ursprünglichen 3D-Koordinaten sind bereits so gut, dass die Erzeugung neuer Koordinaten keine weitere Verbesserung bringt.

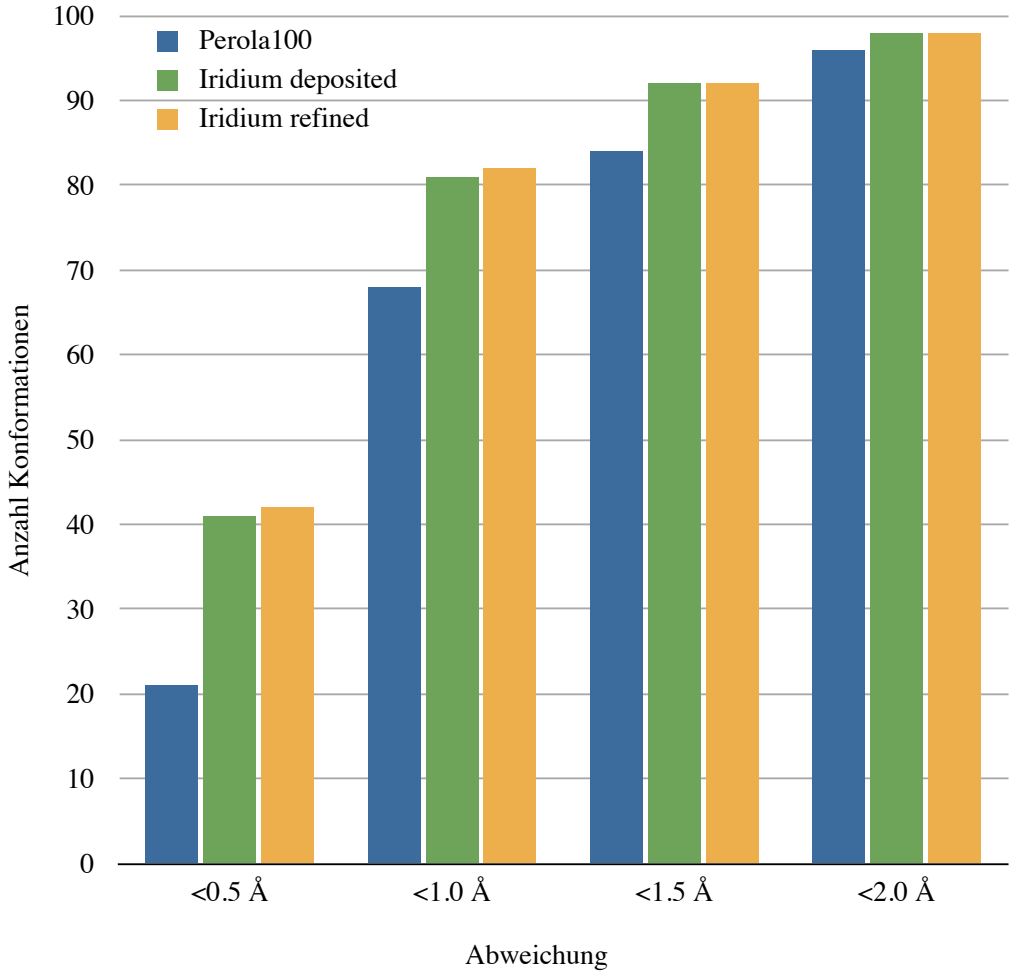


Abbildung 6.20.: Vergleich der CONFECT-Ergebnisse zur Reproduktion der bioaktiven Konformationen für den Iridium- und den Perola100-Datensatz. Iridium deposited: ursprüngliche 3D-Koordinaten wurden beibehalten, Iridium refined: neue 3D-Koordinaten wurden mit coord3d erzeugt.

6.3.2. Vergleich mit anderen Methoden

Die Ergebnisse für den Perola100-Datensatz, die mit der Qualitätsstufe 21 erzielt wurden, wurden benutzt, um CONFECT mit Catalyst, ICM, OMEGA und ConfGen zu vergleichen. Die Ergebnisse des Vergleichs zeigen, dass CONFECT genau so gut in der Lage ist, die bioaktive Konformation zu reproduzieren, wie die anderen vier Programme (siehe Tabelle 6.3 und Abbildung 6.21 oben). Lediglich bei den Konformationen $< 1,5$Å schneidet CONFECT etwas schlechter ab als die anderen. Wird nur die Fähigkeit zur Reproduktion der bioaktiven Konformation betrachtet, bietet CONFECT zunächst einmal keine Vorteile gegenüber den anderen Methoden. Dies ändert sich, wenn zusätzlich die durchschnittliche Ensemblegröße und die durchschnittliche Laufzeit betrachtet werden. Die von CONFECT generierten Ensembles sind im allgemeinen kleiner als die der anderen Programme (siehe Tabelle 6.3 und Abbildung 6.21 unten links). Im Vergleich zu Catalyst, OMEGA und ConfGen generiert CONFECT etwa 30% weniger Konformationen. Auch bei der Betrachtung der durchschnittlichen Laufzeit schneidet CONFECT besser ab als die anderen vier Programme (siehe Tabelle 6.3 und Abbildung 6.21 unten rechts). Hier ist ICM mit 151,8 Sekunden mit Abstand am langsamsten. Unter Berücksichtigung der Ensemblegröße und der Laufzeit zeigen die Ergebnisse, dass CONFECT bei der Reproduktion der bioaktiven Konformationen genau so gut abscheidet wie Catalyst, ICM, OMEGA und ConfGen, dafür aber weniger Zeit und Konformationen benötigt.

6.3.3. Reproduktion mehrerer bioaktiver Konformationen

Für AMP wurden mit CONFECT (Qualitätsstufe 21, Standardeinstellungen) insgesamt 130 Konformationen generiert. Werden alle Konformationen anhand des Adenins überlagert, entsteht der Eindruck, dass der Konformationsraum sehr gleichmäßig abgedeckt wurde (siehe Abbildung 6.22). Die Überlagerung der 343 in der PDB gefundenen Strukturen zeigt allerdings ein anderes Bild. Hier lässt sich eine deutliche Präferenz für bestimmte Konformationen erkennen. Wodurch diese Präferenz zustande kommt (zum Beispiel ähnlicher Bindungsmodus, sterische Hinderung aufgrund ähnlich geformter Bindetaschen) ließe sich allerdings nur mit einer gründlichen Analyse aller Protein-Ligand-Komplexe herausfinden. Diese Analyse wurde aus Zeitgründen nicht gemacht.

Für 226 (65,9%) der 343 bioaktiven Konformationen konnte eine Konformation mit einem RMSD $< 0,5$Å generiert werden. Nach Anhebung des

Tabelle 6.3.: Vergleich der CONFECT-Ergebnisse mit anderen Methoden zur Konformationsgenerierung. „Konf pro Molekül“ gibt die durchschnittliche Anzahl der generierten Konformationen pro Molekül an. „Zeit pro Molekül“ gibt die durchschnittliche Laufzeit pro Molekül an.

Programm	gefundene bioaktive Konformationen (%)				Konf pro Molekül	Zeit pro Molekül (s)
	$\leq$ 0.5 Å	$\leq$ 1.0 Å	$\leq$ 1.5 Å	$\leq$ 2.0 Å		
Catalyst/FAST	23	65	93	97	190	4,9
ICM	18	61	85	97	141	151,8
OMEGA	24	67	87	94	179	1,7
ConfGen comprehensive	17	65	88	98	162	9,1
ConfGen combined	17	70	93	100	181	9,8
CONFECT q21	21	68	84	96	120	1,4

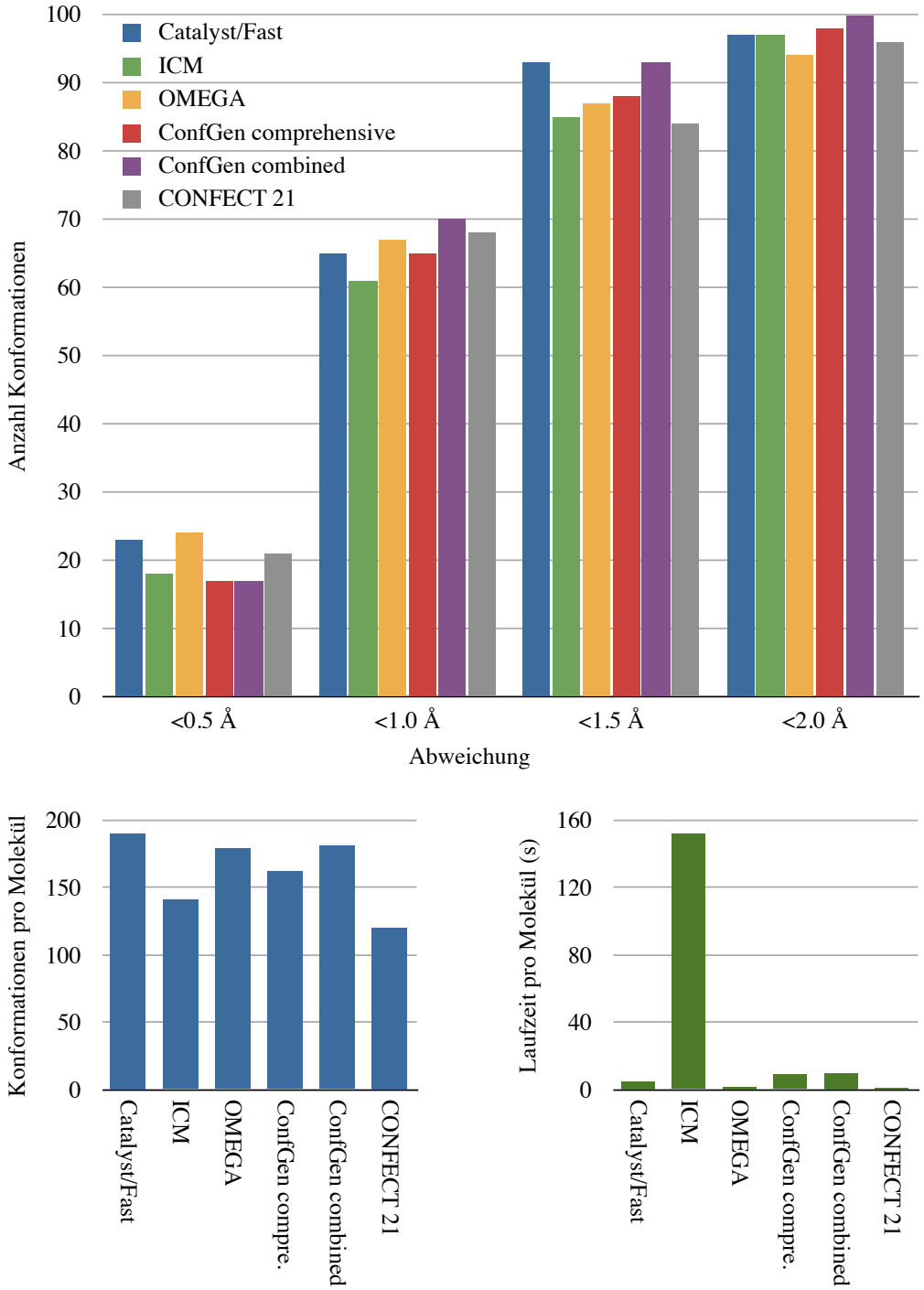


Abbildung 6.21.: Vergleich der CONFECT-Ergebnisse mit den Programmen Catalyst, ICM, OMEGA und ConfGen. Oben: Reproduktion der bioaktiven Konformation. Unten links: durchschnittliche Größe des Konformationsensembles. Unten rechts: durchschnittliche Laufzeit pro Molekül.

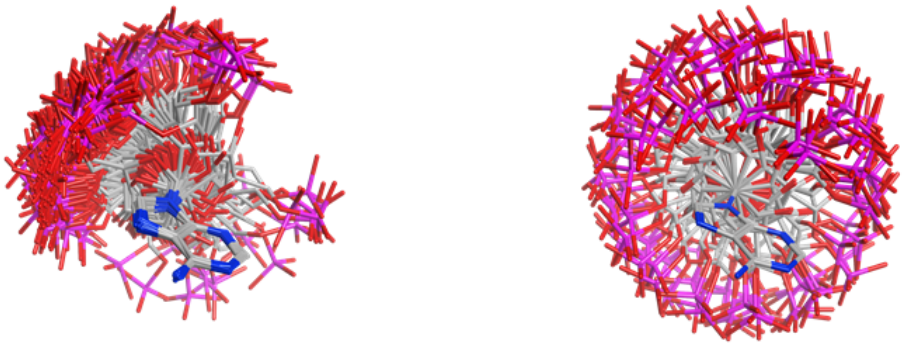


Abbildung 6.22.: Konformationen von Adenosinmonophosphat. Links: 343 AMP-Konformationen aus Protein-Ligand-Komplexen der PDB. Rechts: von CONFECT generiertes Konformationsensemble aus 130 Konformationen. Die Konformationen sind jeweils anhand des Adenosins überlagert.

Grenzwertes auf 1,0Å, konnten bis auf eine alle bioaktiven Konformationen reproduziert werden. Die bioaktive Konformation, die nicht reproduziert werden konnte, stammt aus dem PDB-Komplex 4eq5 [131], welcher nur in einer sehr niedrigen Auflösung von 2,85Å vorliegt. Die Ergebnisse zeigen, dass CONFECT (für AMP) sehr gut in der Lage ist, den Konformationsraum der bioaktiven Konformationen zu reproduzieren.

6.3.4. Laufzeitverhalten

Um die Stabilität und Laufzeit von CONFECT zu testen, wurden Konformationen für 908.007 Moleküle aus der ChEMBL-Datenbank generiert (Qualitätsstufe 1). Die Konformationsgenerierung dauerte 48,7 Stunden auf einem Rechner mit acht CPU-Kernen (Intel(R) Xeon(R) CPU, 2.53GHz) und 32 GB Arbeitsspeicher. Von allen Molekülen konnten nur zwei (zwei zyklische Phosphate) nicht von CONFECT verarbeitet werden. Für 694 Moleküle konnten erst nach einem erneuten Lauf mit der Qualitätsstufe 21 Konformationen generiert werden. Für weitere 130 Moleküle konnten überhaupt keine Konformationen generiert werden. Eine genauere Analyse mit dem TorsionAnalyzer hat gezeigt, dass für diese Moleküle Konformationen ohne überlappende Atome mindestens einen als *selten* klassifizierten Torsionswinkel enthalten würden (siehe Abbildung 6.23). Durch Hinzufügen neuer spezifischer Torsionssignaturen könnte das Problem allerdings behoben werden.

Die Ergebnisse zeigen, dass sowohl die Laufzeit als auch die Ensemblegröße

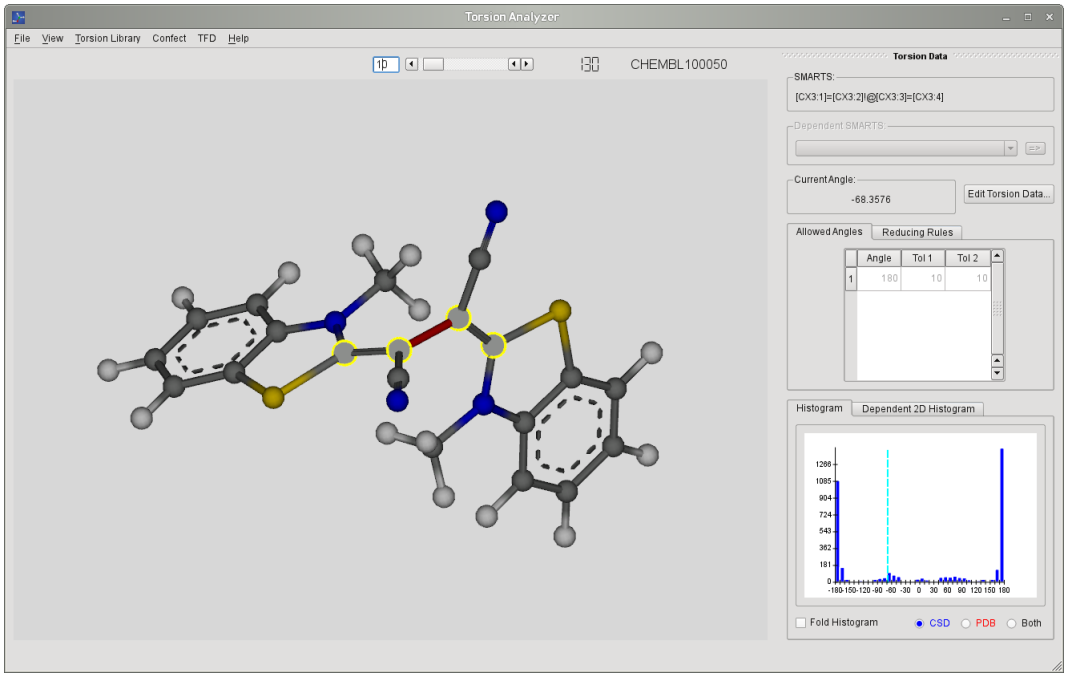


Abbildung 6.23.: Beispielmolekül aus dem ChEMBL-Datensatz, für das keine Konformation mit CONFECT generiert werden konnte. Für dieses Moleküle würde eine Konformation ohne überlappende Atome mindestens einen als *selten* klassifizierten Torsionswinkel enthalten

von der Anzahl der rotierbaren Bindungen abhängt (siehe Abbildung 6.24). Die durchschnittliche Laufzeit für Moleküle mit 0–2 rotierbaren Bindungen ist höher als die Laufzeit für die flexibleren Moleküle. Ein Großteil der Moleküle dieser Klasse enthält sehr flexible Ringsysteme, so dass der größte Teil der Laufzeit für die Generierung von Ringkonformationen verwendet wird. Insgesamt gesehen sorgen die Parametereinstellungen von CONFECT allerdings dafür, dass Laufzeit und Ensemblegröße mit zunehmender Flexibilität nur mäßig ansteigen.

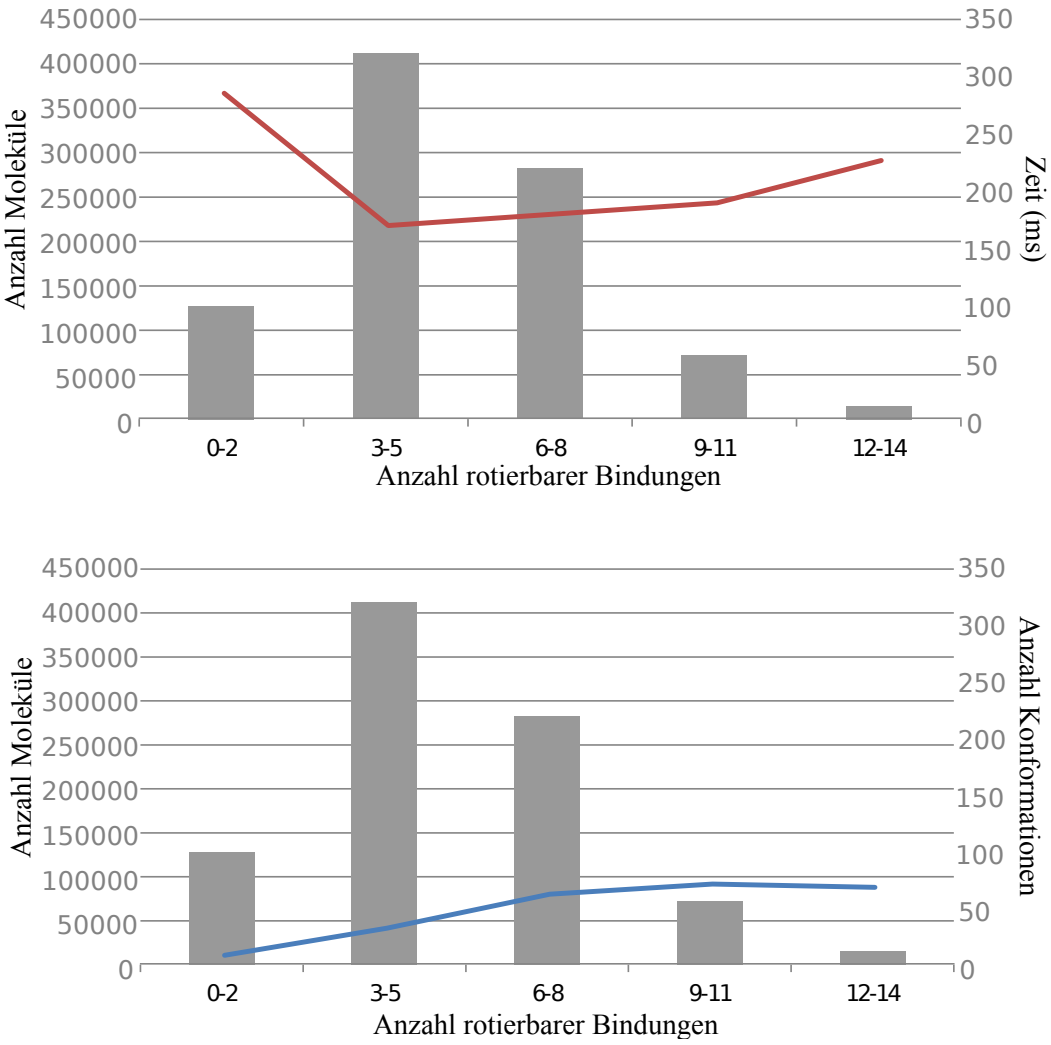


Abbildung 6.24.: Ergebnisse für den ChEMBL-Datensatz. Oben: Laufzeit in Abhängigkeit von der Molekülflexibilität. Unten: Ensemblegröße in Abhängigkeit von der Molekülflexibilität. Balken: Anzahl der Molekül mit einer bestimmten Anzahl rotierbarer Bindungen.

Kapitel 7

7 Zusammenfassung und Ausblick

In dieser Arbeit wurden neue Methoden zur Analyse und Generierung von Molekülkonformationen vorgestellt, die im folgenden Abschnitt noch einmal zusammengefasst werden. Des Weiteren werden Grenzen der Methoden aufgezeigt und Verbesserungs- und Erweiterungsmöglichkeiten vorgeschlagen.

7.1. TFD

Konformationen unterscheiden sich hauptsächlich in ihren Torsionswinkeln. Der *TFD* nutzt diese Eigenschaft, um Konformationen eines Moleküls miteinander zu vergleichen. Dazu wird für jede Konformation zuerst ein *Torsion-Fingerprint (TF)* erstellt, welcher alle Torsionswinkel und Ringtorsionen der Konformation beschreibt. Zur Berechnung der Differenz (TFD) zwischen dem TF einer Referenz-Konformation und dem TF einer vorhergesagten Konformation des selben Moleküls wird zuerst für jede Bindung und jedes Ringsystem die Differenz der Torsionswinkel berechnet. Anschließend wird diese Differenz auf eine Zahl zwischen 0 und 1 normalisiert. Die normalisierten Differenzen werden mit Hilfe einer Gauss-Funktion gewichtet, um sicher zu stellen, das Abweichungen an topologisch zentralen Bindungen und Ringen einen höheren Einfluss auf den TFD ausüben als Abweichungen an terminalen Bindungen und Ringen. Um den endgültigen

TFD zu berechnen, wird die Summe der gewichteten Differenzen durch die Summe der Gauß-Gewichte geteilt.

Zur Evaluierung wurde der TFD mit dem relativen RMSD verglichen. Der RMSD ist ein weit verbreitetes Maß zum Vergleich von Konformationen, dessen Vorteile seine universelle Einsetzbarkeit, seine Objektivität und seine einfache und automatisierbare Berechnung sind. Allerdings hat der RMSD-Vergleich auch gravierende Nachteile. Die Ergebnisse der Evaluierung haben gezeigt, dass man durch die Nutzung des TFD diese Nachteile umgehen kann. Die Erstellung, Gewichtung und der Vergleich von TFs ist einfach zu Implementieren und daher ebenso universell einsetzbar wie der RMSD. Durch die einfache implementierung kann die Berechnung des TFD, genau wie die Berechnung des RMSD automatisiert werden und bietet somit die gleiche Objektivität. TFD- und RMSD-Werte für Konformationen eines Moleküls sind oft korreliert, was die Interpretation der Ergebnisse für RMSD-Benutzer sehr einfach macht. Die Normalisierung des TFD macht es zudem einfacher, einen Grenzwert festzulegen, der die Konformationen in ähnlich und unähnlich zur bioaktiven Konformation einteilt. Des Weiteren ist der TFD unabhängig von der Größe des Moleküls. Dies, zusammen mit der Normalisierung, stellt sicher, dass TFD-Werte über einen großen Datensatz gemittelt werden können, ohne das Gesamtergebnis zu verzerren.

Grenzen der Methode

Wie bereits am Beispiel des Liganden aus dem PDB-Komplex 1acl gezeigt wurde (siehe Abschnitt 6.1.1), sind der Anwendbarkeit des TFD für sehr flexible Moleküle mit langen gleichförmigen Ketten Grenzen gesetzt. Die hohe Gewichtung der zentralen Bindung und die fehlende Berücksichtigung des Effekts der gegenseitigen Aufhebung der Torsionswinkelabweichungen führen in diesem Beispiel zu einer falschen Bewertung durch den TFD. Wenn allerdings eines der zentralen Kohlenstoffatome des 1acl-Liganden durch ein mit dem Protein interagierendes Amid ersetzt werden würde, würde eine starke Abweichung des zentralen Torsionswinkels dazu führen, dass das Amid nicht mehr mit dem Protein interagieren kann. In diesem Fall wäre die Bewertung durch den TFD wieder korrekt. Die Entscheidung, welche Atome oder funktionellen Gruppen für eine Bindung an ein Protein wichtig sind, ist ohne die Bindetasche des Proteins oder eine bekannte Menge bioaktiver Moleküle nicht möglich. Da der TFD allerdings mit der Absicht entwickelt wurde, unabhängig vom Protein zu sein, werden funktionelle Gruppen bei der TFD-Berechnung nicht berücksichtigt.

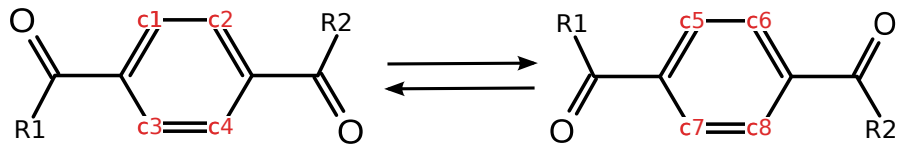


Abbildung 7.1.: Da für die Berechnung des TFD nur eine der möglichen Zuordnungen der Atome der beiden Konformationen benutzt wird, ergibt sich anstatt des erwarteten TFD von 0 ein TFD von 1.

Zur Berechnung des TFD müssen die Konformationen zwar nicht aufwendig überlagert werden, aber es wird eine Zuordnung von Atomen benötigt, um sicherzustellen, dass die richtigen Torsionswinkel miteinander verglichen werden. Die Zuordnung der Atome erfolgt anhand der unique SMILES jeder Konformation. Dadurch ergibt sich pro Konformation nur eine einzige Zuordnung, was zu Problemen führen kann. Der erwartete TFD für die beiden Konformationen in Abbildung 7.1 ist 0. Der berechnete TFD liegt allerdings nahe an 1, da bei der Zuordnung der Atome c1 zu c5, c2 zu c6, c3 zu c7 und c4 zu c8 zugeordnet werden. Dies führt bei beiden Torsionswinkeln zu einer maximalen Abweichung von 180°.

Ausblick

Die Gauß-Funktion sorgt für eine sinnvolle Gewichtung der einzelnen Torsionswinkel. Um allerdings noch besser die Bewertung der Torsionswinkel durch einen erfahrenen Modeller widerspiegeln zu können, könnte die Gewichtungsfunktion so gestaltet werden, dass sie durch den Benutzer austauschbar oder erweiterbar ist. So könnten dann zum Beispiel Größenunterschiede oder die Symmetrie von terminalen Rotoren besser berücksichtigt werden. Eine weitere Verbesserung wäre die Einbeziehung der benachbarten Torsionswinkelabweichung bei der Betrachtung einer rotierbaren Bindung. So könnten Probleme wie bei dem 1acl-Liganden behoben werden. Eine Lösung des Problems in Abbildung 7.1 wäre eine Entscheidung für den minimalen TFD, der sich bei der Berechnung des TFDs aller möglichen Atom-Zuordnungen ergibt. Für das Beispiel würde sich eine weitere Zuordnung ergeben, welche die Atome c1 zu c7, c2 zu c8, c3 zu c5 und c4 zu c6 zuordnet. Dies würde dann zu dem erwarteten TFD von 0 führen.

7.2. Torsionsbibliothek

Die *Torsionsbibliothek* ist eine Sammlung von *Torsionssignaturen*, welche sich aus einem *Torsionsmuster*, einer Liste von häufig vorkommenden Torsionswinkeln und einem oder zwei *Torsionshistogrammen* zusammensetzen. Das Torsionsmuster beschreibt in der SMARTS-Notation die an einem Torsionswinkel beteiligten Atome und, je nach Spezifität, mehr oder weniger die weitere chemische Umgebung. Das Torsionshistogramm zeigt die Verteilung der vorkommenden Werte des beschriebenen Torsionswinkels in einem gegebenen Datensatz. Die am häufigsten vorkommenden Werte (*Peaks*) ergeben die Liste mit Torsionswinkeln. Jedem Torsionswinkel der Liste sind zusätzlich noch zwei Toleranzen zugeordnet, die die Breite der Peaks beschreiben. Die Torsionsbibliothek gliedert sich in Haupt- und Subklassen, wobei innerhalb jeder Klasse oder Subklasse die Torsionssignaturen nach abnehmender Spezifität angeordnet sind.

Bei der Analyse einer Konformation wird jeder rotierbaren Bindung mit Hilfe eines SMARTS-Matching-Algorithmus eine passende Torsionssignatur zugeordnet. Anhand der einzelnen Daten der Torsionssignaturen lassen sich die Torsionswinkel der rotierbaren Bindungen dann als *häufig*, *grenzwertig* oder *selten* klassifizieren.

Die Evaluierung hat gezeigt, dass die etwa 400 spezifischen Torsionssignaturen der Torsionsbibliothek den für die Medizinalchemie relevanten chemischen Raum bereits zu ca. 96% abdecken. Der Vergleich von CSD- und PDB-Histogrammen hat ergeben, dass sich die Präferenzen für bestimmte Torsionswinkel nicht unterscheiden. CSD- und PDB-Histogramme weisen häufig die gleichen Peaks auf, allerdings sind die Peaks der PDB-Histogramme im allgemeinen breiter. Anhand von Beispielen wurde außerdem gezeigt, wie mit Hilfe der Torsionsbibliothek, Liganden aus PDB-Komplexen aufbereitet werden können.

Grenzen der Methode

Die Torsionsbibliothek enthält nur sehr wenige Signaturen für abhängige Torsionswinkel, welche zusätzlich noch auf direkt benachbarte rotierbare Bindungen beschränkt sind. Analysen haben allerdings gezeigt, dass es deutlich mehr abhängige Torsionswinkel gibt, wie zum Beispiel Aryl-X-Aryl-Systeme [44]. Ein weiterer wichtiger Teil der Konformationsanalyse ist die Analyse von Ringkonformationen. Das Modell der Torsionsbibliothek

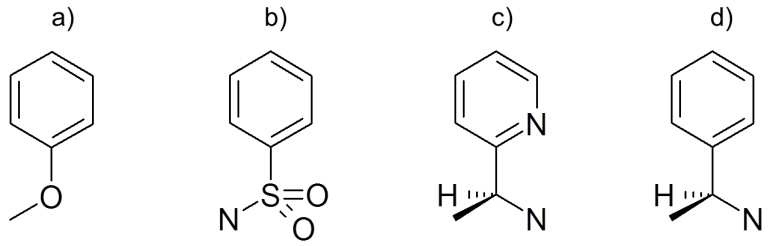


Abbildung 7.2.: Die Chiralität muss bei der Definition von Torsionsmustern berücksichtigt werden: a) achirale, b) prochirale, c) und d) chirale Substruktur.

erlaubt momentan nur die Definition von Torsionssignaturen für azyklische rotierbare Bindungen, so dass eine Analyse von Ringkonformationen ausgeschlossen ist.

Alle Torsionsmuster wurden als achirale Substrukturen definiert, da der verwendete SMARTS-Matching-Algorithmus Chiralität nicht unterstützt. In den meisten Fällen können chirale Moleküle auch mit achiralen Torsionsmustern analysiert werden. Es gibt allerdings auch Fälle, bei denen die Chiralität eine Rolle bei der Definition des Torsionsmusters spielt, obwohl die betrachtete Substruktur eigentlich achiral ist (siehe Abbildung 7.2). Für die Definition des Torsionswinkels eines Arylethers (a) gibt es nur eine Möglichkeit. Beim Sulfonamid (b) hingegen ist eine eindeutige Definition des Torsionswinkels nur über das freie Elektronenpaar des Stickstoffatoms möglich. Die Definition über eines der beiden Sauerstoffatome wäre ohne die Einführung von Chiralität mehrdeutig. Bei anderen Substrukturen (c und d) lässt sich die Chiralität nicht umgehen. Hier müssten chirale Torsionsmuster definiert werden, um eindeutige Torsionshistogramme zu erhalten.

Ausblick

Im Rahmen des dieser Dissertation zugrunde liegenden Projektes wurde bereits eine systematische Suche nach abhängigen Torsionswinkeln durchgeführt. Dazu wurden alle Signaturen der Torsionsbibliothek paarweise anhand eines CSD-Subsets von etwa 73.000 Molekülen analysiert. Die Analyse bestand aus zwei Teilen. Im ersten Teil mussten die Torsionswinkel direkt benachbart sein, wohingegen im zweiten Teil genau eine Bindung dazwischen liegen musste. Aus den resultierenden 2D-Histogrammen wurden im ersten Schritt leere Histogramme herausgefiltert. In einem zweiten

Schritt wurden dann Histogramme mit weniger als 100 Datenpunkten entfernt. Die verbliebenen Histogramme wurden manuell auf Abhängigkeiten untersucht (Abbildung 7.3). Der Nachteil der manuellen Analyse ist, dass sie sehr langsam und die Einteilung in abhängige und unabhängige Torsionssignaturen nicht objektiv ist. Durch die Einführung eines objektiven Maßes zur Bestimmung von Abhängigkeit könnte die Suche nach abhängigen Torsionssignaturen automatisiert werden. Analog zu den Torsionshistogrammen könnten aus den 2D-Histogrammen anschließend automatisch Abhängigkeitsregeln abgeleitet werden.

Das Programm Mogul [83] verfügt über eine veröffentlichte Methode zur automatischen Bewertung von Ringkonformationen [84]. Der Nachteil dieser Methode ist allerdings, dass die Bewertung einer Ringkonformation immer jeweils relativ zu anderen Ringkonformationen vorgenommen wird. Die Häufigkeitsverteilung kann also nicht vorab bestimmt, sondern muss für jede Ringkonformation neu berechnet werden. Zur Analyse von Ringkonformationen mit Hilfe der Torsionsbibliothek müsste ein neuer Deskriptor zur Beschreibung von Ringkonformationen entwickelt werden, der dem Torsionswinkel einer azyklischen rotierbaren Bindungen entspricht. Aufbauend auf diesem Deskriptor könnten dann, analog zu den Torsionssignaturen, Ringsignaturen entwickelt werden. Die Ringsignaturen sollten sich dabei in das Hierarchiekonzept der Torsionsbibliothek einfügen.

Die fehlende Möglichkeit Torsionssignaturen für chirale Substrukturen zu definieren, könnte durch Erweiterung oder Austausch des SMARTS-Matching-Algorithmus gelöst werden.

7.3. CONFECT

CONFECT ist eine wissensbasierte Methode zur Generierung von Molekülkonformationen. Das Molekül wird dazu zuerst in einzelne Komponenten zerlegt und jeder rotierbaren Bindung wird eine passende Torsionssignatur aus der Torsionsbibliothek zugeordnet. Um Konformationen zu erzeugen, werden ausgehend von einer Startkomponente und unter Berücksichtigung der Listen mit häufig vorkommenden Torsionswinkeln sukzessive alle Komponenten wieder hinzugefügt. Ringkonformationen werden mit einem kombinierten Ansatz aus Ringtemplaten und kraftfeldbasierter Optimierung generiert. Generierte Konformationen, bei denen Atome überlappen, werden mit Hilfe eines chemischen Kraftfeldes optimiert. Die Konformationsmenge lässt sich am Ende entweder mit einem TFD- oder

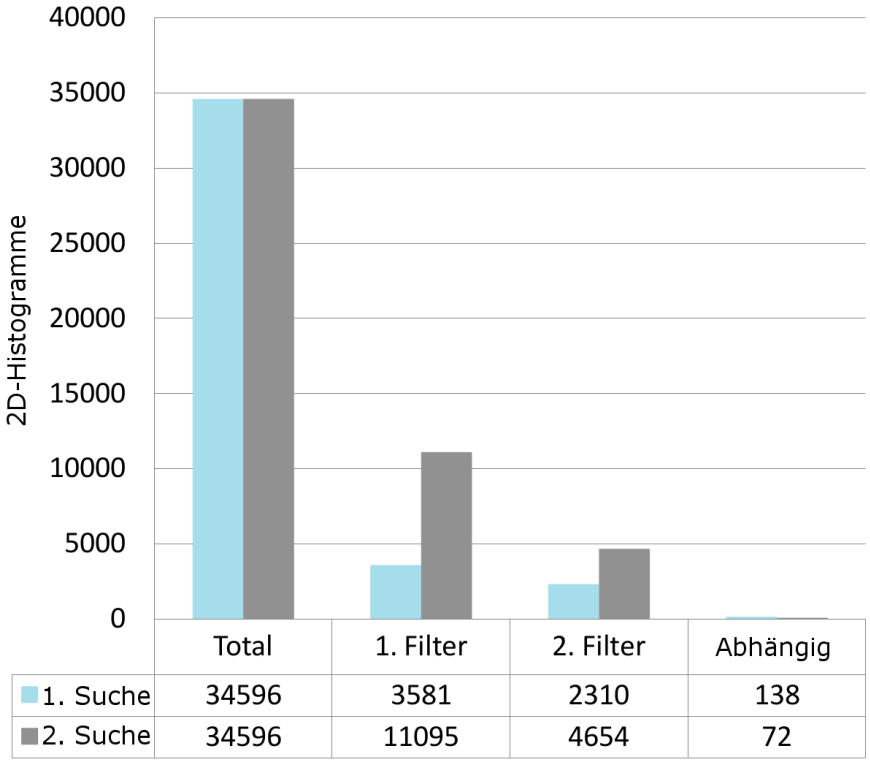


Abbildung 7.3.: Ergebnisse der systematischen Suche nach abhängigen Torsionssignaturen. 1. Suche: Torsionswinkel sind direkt benachbart. 2. Suche: zwischen den Torsionswinkeln liegt genau eine Bindung.

RMSD-Clustering reduzieren. CONFECT besitzt acht verschiedene Qualitätsstufen, um die Abdeckung des Konformationsraums der Moleküle zu steuern. Die Stufen legen zum einen fest, wie viele Konformationen maximal für ein Molekül generiert werden sollen und zum anderen, wie viele Torsionswinkel pro rotierbarer Bindung eingestellt werden. Zur Steuerung der Konformationsgenerierung und zur Sortierung der generierten Konformationen wird eine Bewertungsfunktion verwendet, die auf den relativen Häufigkeiten der Torsionshistogramme basiert.

Die Evaluierung und vor allem der Vergleich der Ergebnisse mit Catalyst, ICM, OMEGA und ConfGen hat gezeigt, das CONFECT genau so gute Ergebnisse bei der Reproduktion der bioaktiven Konformation erzielt wie die anderen Methoden, dabei aber weniger Konformationen generiert und schneller ist. Am Beispiel von AMP wurde gezeigt, dass CONFECT in der Lage ist, mehrere bioaktive Konformationen zu reproduzieren. Die

vorhandenen Möglichkeiten zur Parametrisierung von CONFECT sorgen dafür, dass die Laufzeit und die Ensemblegröße nur mäßig ansteigen.

Grenzen der Methode

Beim Zusammenbau der Konformationen und auch bei der anschließenden Optimierung werden nur die Torsionswinkel eingestellt, die Bindungslängen und Bindungswinkel werden nicht verändert. Bei Strukturen, deren Bindungslängen und -winkel nicht optimal sind, kann es passieren, dass Atome überlappen und dies nicht durch alleinige Rotation von Bindungen aufgelöst werden kann. Dadurch werden eventuell Teilkonformationen, die sehr ähnlich zur bioaktiven Konformation sind, verworfen.

Ein weiteres Problem ist die Bewertung von Ringkonformationen, da es keine zu den Torsionshistogrammen äquivalente Häufigkeitsverteilungen für Ringkonformationen gibt. Die Ringkonformationen werden in der Reihenfolge bewertet, wie sie durch den Ringgenerator entstehen. Der erste Ring bekommt den höchsten Score und jede weitere Ringkonformation die Hälfte des Scores der vorherigen Konformation. Die Reihenfolge der Ringkonformationen entspricht dabei nicht unbedingt den bevorzugten Präferenzen.

Ausblick

Eine Verbesserungsmöglichkeit von CONFECT wäre die Integration von coord3d. Dadurch könnten einerseits die Probleme mit schlechten Bindungslängen und -winkeln der Eingabestruktur gelöst werden und andererseits wäre CONFECT dann nicht mehr auf 3D-Eingabestrukturen angewiesen, sondern könnte auch für 2D-Strukturen Konformationen erzeugen. Auch coord3d würde von der Integration profitieren. Die von coord3d eingestellten Torsionswinkel entsprechen nicht unbedingt den bevorzugten Torsionswinkeln, was durch die Anbindung an CONFECT in Verbindung mit der Qualitätsstufe 6 behoben werden könnte.

Durch das Hinzufügen von Ringsignaturen zur Torsionsbibliothek wäre auch eine sinnvollere Bewertung der Ringkonformationen in CONFECT möglich, was wiederum zu besseren Ergebnissen bei der Reproduktion der bioaktiven Konformation führen würde. Auch die Erweiterung der Torsionsbibliothek um mehr Torsionssignaturen für abhängige Torsionswinkel könnte zu einer weiteren Verbesserung der CONFECT-Ergebnisse beitragen.

7.4. TorsionAnalyzer

Der *TorsionAnalyzer* ist ein graphisches Softwarewerkzeug, mit dem sich nicht nur Molekülkonformationen analysieren lassen, sondern mit dem auch Torsionsbibliotheken angezeigt, erstellt bearbeitet und und abgespeichert werden können. Zur Konformationsanalyse werden eine oder mehrere Konformationen in den TorsionAnalyzer geladen. Beim Laden der Konformationen wird automatisch jeder rotierbaren Bindung eine passende Torsionssignatur zugeordnet. Die rotierbaren Bindungen werden anschließend anhand der Klassifizierung (*häufig*, *grenzwertig*, *selten*) eingefärbt. Der Benutzter kann so auf den ersten Blick erkennen, ob eine der Konformationen seltene Torsionswinkel enthält. Der TorsionAnalyzer bietet außerdem Funktionen zur automatischen Generierung und Analyse von Torsionshistogrammen, zur Berechnung des TFD oder RMSD und zur Generierung von Konformationen.

Im Gegensatz zu Mogul und ConQuest ist der TorsionAnalyzer bei der Erstellung von Torsionshistogrammen nicht auf die CSD beschränkt. Es können beliebige Datensätze zur Analyse der Präferenzen von Torsionswinkeln verwendet werden. Des weiteren ist aus der Torsionssignatur klar ersichtlich, welche Umgebung bei der Analyse einer rotierbaren Bindung verwendet wurde und das Torsionsmuster kann notfalls angepasst werden. Bei Mogul erfolgt die Definition der Umgebung dagegen automatisch und kann durch den Benutzer nicht modifiziert werden.

Grenzen der Methode

Der TorsionAnalyzer wurde ursprünglich zur Ansicht und Bearbeitung der Torsionsbibliothek entwickelt. Die Erweiterungen zur Analyse und Generierung von Konformationen und zur TFD-Berechnung kamen erst im Laufe des Projektes hinzu. Momentan spricht der TorsionAnalyzer zwei verschiedene Benutzergruppen an: zum einen den Chemiker, der wissen möchte, ob die Torsionswinkel seiner Moleküle in Ordnung sind und zum anderen den Modeller, der die Torsionsbibliothek bearbeiten und Konformationen erzeugen möchte. Der TorsionAnalyzer wurde zwar bereits sehr früh von einigen Mitarbeitern der Firma F. Hoffmann-La Roche Ltd. getestet und eingesetzt, für eine breitere Benutzung müssten die graphische Oberfläche und die Benutzerführung allerdings durch einen Experten für Softwareergonomie überarbeitet werden.

Ausblick

Die meisten Benutzer, die den TorsionAnalyzer gesehen und getestet haben, empfanden die Einfärbung der rotierbaren Bindungen sehr hilfreich. Diese Komponente könnte relativ einfach auch in andere Programme eingebaut werden. Möglich wäre zum Beispiel die Kombination mit Hyde [132] in LeadIT [133] oder der Einbau in MOE [23].

Literaturverzeichnis

[1] G. Schneider and H.-J. Böhm. Virtual screening and fast automated docking methods. *Drug Discov. Today*, 7(1):64–70, 2002.

[2] T. Lengauer, C. Lemmen, M. Rarey, and M. Zimmermann. Novel technologies for virtual screening. *Drug Discov. Today*, 9(1):27–34, 2004.

[3] G Klebe. *Wirkstoffdesign: Entwurf und Wirkung von Arzneistoffen*. Spektrum Akademischer Verlag Heidelberg, 2 edition, 2009.

[4] D. B. Kitchen, H. Decornez, J. R. Furr, and J. Bajorath. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nat. Rev. Drug Discovery*, 3(11):935–949, 2004.

[5] T. Tuccinardi. Docking-based virtual screening: recent developments. *Comb. Chem. High T. Scr.*, 12(3):303–14, 2009.

[6] A. Breda, L. A. Basso, D. S. Santos, and W. F. de Azevedo Jr. Virtual Screening of Drugs: Score Functions, Docking, and Drug Design. *Curr. Comput.-Aided Drug Des.*, 4(4):265–272, 2008.

[7] J. A. Grant, M. A. Gallardo, and B. T. Pickup. A fast method of molecular shape comparison: A simple application of a Gaussian description of molecular shape. *J. Comput. Chem.*, 17(14):1653–1666, 1996.

[8] C. H. Schwab. Conformations and 3D pharmacophore searching. *Drug Discov. Today: Technologies*, 7(4):e245–e253, 2010.

[9] J. Gasteiger, editor. *Handbook of Chemoinformatics*, volume 1, chapter 7 & 8. WILEY-VHC Verlag GmbH & Co. KGaA, 2003.

[10] A. R. Leach and K. Prout. Automated conformational analysis: Directed conformational search using the A* algorithm. *J. Comput. Chem.*, 11(10):1193–1205, 1990.

[11] Accelrys, Inc., 9685 North Scranton Road, San Diego, CA 92121, U.S.A. Catalyst. `http://accelrys.com` (Abruf am 08.03.2012).

[12] M. J. Vainio and M. S. Johnson. Generating conformer ensembles using a multiobjective genetic algorithm. *J. Chem. Inf. Model.*, 47(6):2462–2474, 2007.

[13] J. Li, T. Ehlers, J. Sutter, S. Varma-O'Brien, and J. Kirchmair. CAESAR: a new conformer generation algorithm based on recursive buildup and local rotational symmetry consideration. *J. Chem. Inf. Model.*, 47(5):1923–1932, 2007.

[14] A. Smellie, R. Stanton, R. Henne, and S. Teig. Conformational analysis by intersection: CONAN. *J. Comput. Chem.*, 24(1):10–20, 2003.

[15] N. M. O'Boyle, T. Vandermeersch, C. J. Flynn, A. R. Maguire, and G. R. Hutchison. Confab - Systematic generation of diverse low-energy conformers. *J. Cheminf.*, 3(1):8, 2011.

[16] K. S. Watts, P. Dalal, R. B. Murphy, W. Sherman, R. A. Friesner, and J. C. Shelley. ConfGen: a conformational search method for efficient generation of bioactive conformers. *J. Chem. Inf. Model.*, 50(4):534–546, 2010.

[17] R.S. Pearlman and R. Balducci. National Meeting of the American Chemical Society, New Orleans, LA, USA, 1998.

[18] X. Liu, F. Bai, S. Ouyang, X. Wang, H. Li, and H. Jiang. Cyndi: a multi-objective evolution algorithm based method for bioactive molecular conformational generation. *BMC Bioinformatics*, 10(1):1–14, 2009.

[19] T. B. Leite, D. Gomes, M. A. Miteva, J. Chomilier, B. O. Villoutreix, and P. Tuffery. Frog: a FRee Online druG 3D conformation generator. *Nucleic Acids Res.*, 35:W568–W572, 2007.

[20] M. A. Miteva, F. Guyon, and P. Tuffery. Frog2: Efficient 3D conformation ensemble generator for small compounds. *Nucleic Acids Res.*, 38(suppl 2):W622–W627, 2010.

[21] F. Mohamadi, N. G. J. Richards, W. C. Guida, R. Liskamp, M. Lipton, C. Caufield, G. Chang, T. Hendrickson, and W. C. Still. MacroModel—an integrated software system for modeling organic and bioorganic molecules using molecular mechanics. *J. Comput. Chem.*, 11(4):440–467, 1990.

[22] G. Klebe and T. Mietzner. A fast and efficient method to generate biologically relevant conformations. *J. Comput.-Aided Mol. Des.*, 8(5):583–606, 1994.

[23] Chemical Computing Group. Molecular operating environment moe, version 2010.10. `http://www.chemcomp.com` (Abruf am 08.03.2012).

[24] P. C. D. Hawkins, A. G. Skillman, G. L. Warren, B. A. Ellingson, and M. T. Stahl. Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and Cambridge Structural Database. *J. Chem. Inf. Model.*, 50(4):572–584, 2010.

[25] Open-source cheminformatics. Rdkit: Cheminformatics and machine learning software. `http://rdkit.org` (Abruf am 08.03.2012).

[26] S. Renner, C. H. Schwab, J. Gasteiger, and G. Schneider. Impact of Conformational Flexibility on Three-Dimensional Similarity Searching Using Correlation Vectors. *J. Chem. Inf. Model.*, 46(6):2324–2332, 2006.

[27] A. Griewel, O Kayser, J. Schlosser, and M. Rarey. Conformational sampling for large-scale virtual screening: accuracy versus ensemble size. *J. Chem. Inf. Model.*, 49(10):2303–2311, 2009.

[28] D. Young. *Computational Chemistry. A Practical Guide for Applying Techniques to Real World Problems*, chapter 21. WILEYINTERSCIENCE, 2001.

[29] B. Musafia and H. Senderowitz. Biasing conformational ensembles towards bioactive-like conformers for ligand-based drug design. *Expert Opin. Drug Discov.*, 5(10):943–959, 2010.

[30] T. Schulz-Gasch, C. Schärfer, W. Guba, and M. Rarey. TFD: Torsion Fingerprints As a New Measure To Compare Small Molecule Conformations. *J. Chem. Inf. Model.*, 52(6):1499–1512, 2012.

[31] C. Schärfer, T. Schulz-Gasch, H.-C. Ehrlich, W. Guba, M. Rarey, and M. Stahl. Torsion Angle Preferences in Drug-like Chemical Space: A Comprehensive Guide. *J. Med. Chem.*, 56(5):2016–2028, 2013.

[32] C. Schärfer, T. Schulz-Gasch, J. Hert, B. Schulz, T. Inhester, M. Stahl, and M. Rarey. Confect: Conformations from an Expert Collection of Torsion Patterns. *in Vorbereitung*.

[33] A. R. Leach and V. J. Gillet. *An Introduction to Chemoinformatics*, chapter 2. Springer, 2007.

[34] M. Vieth, J. D. Hirst, and C. L. Brooks. Do active site conformations of small ligands correspond to low free-energy solution structures? *J. Comput.-Aided Mol. Des.*, 12(6):563–572, 1998.

[35] Gareth R Stockwell and Janet M Thornton. Conformational Diversity of Ligands Bound to Proteins. *J. Mol. Biol.*, 356(4):928–944, 2006.

[36] J Boström, P O Norrby, and T Liljefors. Conformational energy penalties of protein-bound ligands. *J. Comput.-Aided Mol. Des.*, 12(4):383–383, 1998.

[37] E. Perola and P.S. Charifson. Conformational analyses of drug-like molecules bound to proteins: An extensive study of ligand reorganization upon binding. *J. Med. Chem.*, 47(10):2499–2510, 2004.

[38] Marc C Nicklaus, Shaomeng Wang, John S Driscoll, and George W A Milne. Conformational changes of small molecules binding to proteins. *Bioorg. Med. Chem.*, 3(4):411–428, 1995.

[39] S Günther, C Senger, E Michalsky, A Goede, and R Preissner. Representation of target-bound drugs by computed conformers: implications for conformational libraries. *BMC Bioinformatics*, 7(1):293, 2006.

[40] R. Mannhold, editor. *Molecular Drug Properties*, volume 37, chapter 8. WILEY-VHC Verlag GmbH & Co. KGaA, 2008.

[41] F H Allen. The Cambridge Structural Database: a quarter of a million crystal structures and rising. *Acta Cryst.*, B58:380–388, 2002.

[42] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res.*, 28(1):235–242, 2000.

[43] A. R. Leach. *Molecular Modelling: Principles and Applications; Second Edition*, chapter 9. Pearson Education Limited, 2001.

[44] K. A. Brameld, B. Kuhn, D. C. Reuter, and M. Stahl. Small molecule conformational preferences derived from crystal structure data. A medicinal chemistry focused analysis. *J. Chem. Inf. Model.*, 48(1):1–24, 2008.

[45] C. Bissantz, B. Kuhn, and M. Stahl. A Medicinal Chemist's Guide to Molecular Interactions. *J. Med. Chem.*, 53(14):5061–5084, 2010.

[46] B. Kuhn, P. Mohr, and M. Stahl. Intramolecular Hydrogen Bonding in Medicinal Chemistry. *J. Med. Chem.*, 53(6):2601–2611, 2010.

[47] 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, and F. Diederich. Systematic Investigation of Halogen Bonding in Protein-Ligand Interactions. *Angew. Chem. Int. Ed.*, 50(1):314–318, 2011.

[48] B. Kuhn, J. E. Fuchs, M. Reutlinger, M. Stahl, and N. R. Taylor. Rationalizing Tight Ligand Binding through Cooperative Interaction Networks. *J. Chem. Inf. Model.*, 51(12):3180–3198, 2011.

[49] J. Liebeschuetz, J. Hennemann, T. Olsson, and C. R. Groom. The good, the bad and the twisted: a survey of ligand geometry in protein crystal structures. *J. Comp.-Aided Mol. Des.*, 26(2):169–183, 2012.

[50] The Cambridge Crystallographic Data Centre. CSD (Cambridge Structural Database), Version 5.34. `http://www.ccdc.cam.ac.uk/Solutions/CSDSystem/Pages/CSD.aspx` (Abruf am 21.03.2013).

[51] W. Borchardt-Ott. *Kristallographie: eine Einführung für Naturwissenschaftler*. Springer Verlag, Berlin, 1997.

[52] C. G. Shull. Early development of neutron scattering. *Rev. Mod. Phys.*, 67:753–757, 1995.

[53] D. Williams. *Nuclear magnetic resonance spectroscopy*. John Wiley and Sons Inc.,New York, NY, 1986.

[54] Research Collaboratory for Structural Bioinformatics (RCSB). PDB (Protein Data Bank). `http://www.rcsb.org` (Abruf am 21.03.2013).

[55] Research Collaboratory for Structural Bioinformatics (RCSB). Yearly Growth of Total Structures. `http://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=total&seqid=100` (Abruf am 21.03.2013).

[56] J. Kirchmair, P. Markt, S. Distinto, G. Wolber, and T. Langer. Evaluation of the performance of 3D virtual screening protocols: RMSD comparisons, enrichment assessments, and decoy selection–what can we learn from earlier mistakes? *J. Comput.-Aided Mol. Des.*, 22(3-4):213–228, 2008.

[57] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, 32(5):922–923, 1976.

[58] J. Kirchmair, G. Wolber, C. Laggner, and T. Langer. Comparative performance assessment of the conformational model generators omega and catalyst: a large-scale survey on the retrieval of protein-bound ligand conformations. *J. Chem. Inf. Model.*, 46(4):1848–1861, 2006.

[59] R. T. Kroemer, A. Vulpetti, J. J. McDonald, D. C. Rohrer, J-Y. T. F. Giordanetto, S. Cotesta, C. Mcmartin, M. Kihlen, and P. F. W. Stouten. Assessment of docking poses: interactions-based accuracy classification (IBAC) versus crystal structure deviations. *J. Chem. Inf. Comput. Sci.*, 44(3):871–881, 2004.

[60] J. C. Baber, D. C. Thompson, J. B. Cross, and C. Humblet. GARD: A Generally Applicable Replacement for RMSD. *J. Chem. Inf. Model.*, 49(8):1889–1900, 2009.

[61] G. J. Kleywegt. Validation of protein crystal structures. *Acta Crystallogr., Sect. D: Biol. Crystallogr.*, 56(3):249–265, 2000.

[62] C.-I. Bränden and T. A. Jones. Between objectivity and subjectivity. *Nature*, 343(6260):687–689, 1990.

[63] A. M. Davis, S. J. Teague, and G. J. Kleywegt. Application and limitations of X-ray crystallographic data in structure-based ligand and drug design. *Angew. Chem., Int. Ed.*, 42(24):2718–2736, 2003.

[64] D. Yusuf, A. M. Davis, G. J. Kleywegt, and S. Schmitt. An alternative method for the evaluation of docking performance: RSR vs RMSD. *J. Chem. Inf. Model.*, 48(7):1411–1422, 2008.

[65] OpenEye Scientific Software. ROCS. `http://www.eyesopen.com/rocs` (Abruf am 08.03.2012).

[66] J. Gasteiger, C. Rudolph, and J. Sadowski. Automatic generation of 3D-atomic coordinates for organic molecules. *Tetrahedron Comput. Methodol.*, 3(6):537–547, 1990.

[67] A. Rusinko III, R. P. Sheridan, and R. Nilakantan. Using CONCORD to construct a large database of three-dimensional coordinates from connection tables. *J. Chem. Inf. Comput. Sci.*, 29(4):251–255, 1989.

[68] Therese Inhester. Generation of Small-Molecule 3d Coordinates for High-Throughput Applications. Master's thesis, ZBH Center for Bioinformatics, University of Hamburg, 2012.

[69] J.-P. Ebejer, G. M. Morris, and C. M. Deane. Freely Available Conformer Generation Methods: How Good Are They? *J. Chem. Inf. Model.*, 52(5):1146–1158, 2012.

[70] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM - A Program for Macromolecular Energy, Minimization, and Dynamics Calculations. *J. Comput. Chem.*, 4(2):187–217, 1983.

[71] I.-J. Chen and N. Foloppe. Conformational Sampling of Druglike Molecules with MOE and Catalyst: Implications for Pharmacophore Modeling and Virtual Screening. *J. Chem. Inf. Model.*, 48(9):1773–1791, 2008.

[72] A Smellie, S L Teig, and P Towbin. Poling: promoting conformational variation. *J. Comput. Chem.*, 16(2):171–187, 2004.

[73] T. A. Halgren. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.*, 17(5):490–519, 1996.

[74] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, D. E. Shaw, P. Francis, and P. S. Shenkin. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J. Med. Chem.*, 47(7):1739–1749, 2004.

[75] T. A. Halgren, R. B. Murphy, R. A. Friesner, H. S. Beard, L. L. Frye, W. T. Pollard, and J. L. Banks. Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *J. Med. Chem.*, 47(7):1750–1759, 2004.

[76] W L Jorgensen and J Tirado-Rives. The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.*, 110:165, 1988.

[77] W L Jorgensen, D S Maxwell, and J TiradoRives. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc.*, 118:11225–11236, 1996.

[78] Jay L Banks, Hege S Beard, Yixiang Cao, Art E Cho, Wolfgang Damm, Ramy Farid, Anthony K Felts, Thomas A Halgren, Daniel T Mainz, Jon R Maple, Robert Murphy, Dean M Philipp, Matthew P Repasky, Linda Y Zhang, Bruce J Berne, Richard A Friesner, Emilio Gallicchio, and Ronald M Levy. Integrated Modeling Program, Applied Chemical Theory (IMPACT). *J. Comput. Chem.*, 26(16):1752–1780, 2005.

[79] I.-J. Chen and N. Foloppe. Drug-like Bioactive Structures and Conformational Coverage with the LigPrep/ConfGen Suite: Comparison to Programs MOE and Catalyst. *J. Chem. Inf. Model.*, 50(5):822–839, 2010.

[80] B. Cordero, V. Gomez, A. E. Platero-Prats, M. Reves, J. Echeverria, E. Cremades, F. Barragan, and S. Alvarez. Covalent radii revisited. *Dalton Trans.*, 0:2832–2838, 2008.

[81] The Cambridge Crystallographic Data Centre. ConQuest - The Interface for the CSD System, Version 1.14. `http://www.ccdc.cam.ac.uk/products/csd_system/conquest/` (Abruf am 15.10.2012).

[82] I J Bruno, J C Cole, M Kessler, J Luo, W D S Motherwell, L H Purkis, B R Smith, R Taylor, R I Cooper, S E Harris, and A G Orpen. Retrieval of Crystallographically-Derived Molecular Geometry Information. *J. Chem. Inf. Model.*, 44(6):2133–2144, 2004.

[83] The Cambridge Crystallographic Data Centre. Mogul - A Knowledge Base of Molecular Geometry, Version 3.0. `http://www.ccdc.cam.ac.uk/products/csd_system/mogul/` (Abruf am 15.10.2012).

[84] S. J. Cottrell, T. S. G. Olsson, R. Taylor, and J. C. Cole. Validating and Understanding Ring Conformations Using Small Molecule Crystallographic Data. *J. Chem. Inf. Model*, 52(4):956–962, 2012.

[85] D. W. Banner and P. Hadváry. Crystallographic analysis at 3.0-A resolution of the binding to human thrombin of four active site-directed inhibitors. *J. Biol. Chem.*, 266(30):20085–20093, 1991.

[86] P. Vismara. Union of all the minimum cycle bases of a graph. *Electron. J. Combin.*, 4:1–15, 1997.

[87] R. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6), 1962.

[88] D. Weininger, A. Weininger, and J. Weininger. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.*, 29(2):97–101, 1989.

[89] D. Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36., 1988.

[90] OpenEye Scientific Software. SMARTS Pattern Matching. http://www.eyesopen.com/docs/toolkits/current/html/OEChem_TK-python/SMARTS.html (Abruf am 06.08.2012).

[91] Open Babel. SMARTS. http://openbabel.org/wiki/SMARTS (Abruf am 06.08.2012).

[92] World Wide Web Consortium (W3C). Extensible Markup Language (XML). http://www.w3.org/XML/ (Abruf am 07.01.2013).

[93] World Wide Web Consortium (W3C). XML Schema. http://www.w3.org/XML/Schema.html (Abruf am 07.01.2013).

[94] L. C. Ray and R. A. Kirsch. Finding Chemical Records by Digital Computers. *Science*, 126(3278):814–819, 1957.

[95] E. H. Sussenguth. A Graph-Theoretic Algorithm for Matching Chemical Structures. *J. Chem. Doc.*, 5(1):36–43, 1965.

[96] J. R. Ullmann. An algorithm for subgraph isomorphism. *J. Assoc. Comput. Mach.*, 23(1):31–42, 1976.

[97] L.P. Cordella, P. Foggia, C. Sansone, and M. Vento. Performance evaluation of the VF graph matching algorithm. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 1172 –1177, Venice, Italy, 1999. IEEE Computer Society.

[98] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 26(10):1367–1372, 2004.

[99] H-C Ehrlich and M Rarey. Systematic benchmark of substructure search in molecular graphs — From Ullmann to VF2. *J. Cheminformatics*, 4(1):13, 2012.

[100] J. Sadowski and J. Boström. MIMUMBA revisited: torsion angle rules for conformer generation derived from X-ray structures. *J. Chem. Inf. Model.*, 46(6):2305–2309, 2006.

[101] M. Rarey, B. Kramer, T. Lengauer, and G. Klebe. A fast flexible docking method using an incremental construction algorithm. *J. Mol. Biol.*, 261(3):470–489, 1996.

[102] A. Kolodzik, S. Urbaczek, and M. Rarey. Unique Ring Families: A Chemically Meaningful Description of Molecular Ring Topologies. *J. Chem. Inf. Model.*, 52(8):2013–2021, 2012.

[103] Ole Kayser. Efficient Methods for the Generation of Bioactive Conformers of Small Molecules. Master's thesis, ZBH Center for Bioinformatics, University of Hamburg, 2007.

[104] Tobias Lippert. *Pharmakophorbasierte Leitstruktursuche in chemischen Fragmenträumen mit stochastischen Methoden*. PhD thesis, ZBH Center for Bioinformatics, University of Hamburg, 2011.

[105] L. Heinzerling, R. Klein, and M. Rarey. Fast force field-based optimization of protein–ligand complexes with graphics processor. *J. Comput. Chem.*, 33(32):2554–2565, 2012.

[106] M. K. Safo, C. M. Moure, J. C. Burnett, G. S. Joshi, and D. J. Abraham. High-resolution crystal structure of deoxy hemoglobin complexed with a potent allosteric effector. *Protein Sci.*, 10(5):951–957, May 2001.

[107] Tripos. Tripos Mol2 File Format. `http://www.tripos.com/tripos_resources/fileroot/pdfs/mol2_format.pdf` (Abruf am 07.01.2013).

[108] Accelrys. About Accelrys CTfile formats. `http://download.accelrys.com/freeware/ctfile-formats/ctfile-formats.zip` (Abruf am 07.01.2013).

[109] K. Schomburg, H.-C. Ehrlich, K. Stierand, and M. Rarey. From Structure Diagrams to Visual Chemical Patterns. *J. Chem. Inf. Model.*, 50(9):1529–1535, 2010.

[110] J. Boström, J. R. Greenwood, and J. Gottfries. Assessing the performance of OMEGA with respect to retrieving bioactive conformations. *J. Mol. Graphics Modell.*, 21(5):449–462, 2003.

[111] J. M. Chen, S. L. Xu, Z. Wawrzak, G. S. Basarab, and D. B. Jordan. Structure-based design of potent inhibitors of scytalone dehydratase: displacement of a water molecule from the active site. *Biochemistry*, 37(51):17735–17744, 1998.

[112] W. Bode and D. Turk. Geometry of binding of the benzamidine- and arginine-based inhibitors N alpha-(2-naphthyl-sulphonyl-glycyl)-DL-p-amidinophenylalanyl-pipe ridine (NAPAP) and (2R,4R)-4-methyl-1-[N alpha-(3-methyl-1,2,3,4-tetrahydro-8- quinolinesulphonyl)-L-arginyl]-2-piperidine carboxylic acid (MQPA) to human alpha-thrombin. X-ray crystallographic determination of the NAPAP-trypsin complex and modeling of NAPAP-thrombin and MQPA-thrombin. *Eur. J. Biochem.*, 193:175–182, 1990.

[113] R. Bone, T. Lu, C. R. Illig, R. M. Soll, and J. C. Spurlino. Structural analysis of thrombin complexed with potent inhibitors incorporating a phenyl group as a peptide mimetic and aminopyridines as guanidine substitutes. *J. Med. Chem.*, 41(12):2068–2075, 1998.

[114] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. P. Overington. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.*, 40:D1100–D1107, 2012.

[115] Accelrys. Pipeline Pilot, Version . http://accelrys.com/products/pipeline-pilot/ (Abruf am 01.11.2012).

[116] Desert Scientific Software. Proasis2. http://www.desertsci.com/ (Abruf am 16.10.2012).

[117] OpenEye Scientific Software. Iridium: A Highly Trustworthy Protein-Ligand Structure Database. http://www.eyesopen.com/iridium (Abruf am 15.10.2012).

[118] M. Harel, I. Schalk, L. Ehret-Sabatier, F. Bouet, M. Goeldner, C. Hirth, P.H. Axelsen, I. Silman, and J.L. Sussman. Quaternary ligand binding to aromatic residues in the active-site gorge of acetylcholinesterase. *Proc. Natl. Acad. Sci. USA*, 90(19):9031–9035, 1993.

[119] B.A. Katz, P.A. Sprengeler, C. Luong, E. Verner, K. Elrod, M. Kirtley, J. Janc, J.R. Spencer, J.G. Breitenbucher, H. Hui, D. McGee, D. Allen, A. Martelli, and R.L. Mackman. Engineering inhibitors highly selective for the S1 sites of Ser190 trypsin-like serine protease drug targets. *Chem. Biol.*, 8(11):1107–1121, 2001.

[120] M. Weyand, I. Schlichting, A. Marabotti, and A. Mozzarelli. Crystal structures of a new class of allosteric effectors complexed to tryptophan synthase. *J. Biol. Chem.*, 277(12):10647–10652, 2002.

[121] C. Mattos, B. Rasmussen, X. Ding, G.A. Petsko, and D. Ringe. Analogous inhibitors of elastase do not always bind analogously. *Nat. Struct. Biol.*, 1(1):55–58, 1994.

[122] A. J. Cruz-Cabeza, J. W. Liebeschuetz, and F. H. Allen. Systematic conformational bias in small-molecule crystal structures is rare and explicable. *Cryst. Eng. Comm.*, 14(20):6797–6811, 2012.

[123] R. A. Pascal, C. M. Wang, G. C. Wang, and L. V. Koplitz. Ideal Molecular Conformation versus Crystal Site Symmetry. *Crystal Growth and Design*, 12(9):4367–4376, 2012.

[124] J. R. Kiefer, J. L. Pawlitz, K. T. Moreland, R. A. Stegeman, W. F. Hood, J. K. Gierse, A. M. Stevens, D. C. Goodwin, S. W. Rowlinson, L. J. Marnett, W. C. Stallings, and R. G. Kurumbail. Structural insights into the stereochemistry of the cyclooxygenase reaction. *Nature*, 405(6782):97–101, 2000.

[125] W. Ke, C.R. Bethel, K.M. Papp-Wallace, S.R. Pagadala, M. Nottingham, D. Fernandez, J.D. Buynak, R.A. Bonomo, and F. van den Akker. Crystal structures of KPC-2 beta-lactamase in complex with 3-nitrophenyl boronic acid and the penam sulfone PSR-3-226. *Antimicrob. Agents Chemother.*, 56(5):2713–2718, 2012.

[126] Edwards, T.E. and Davies, D.R. and Hartley, R. and Zeller, W. Crystal structure of 2C-methyl-D-erythritol 2,4-cyclodiphosphate synthase from Burkholderia pseudomallei in complex with a fragment-nucleoside fusion D000161829. `http://www.rcsb.org/pdb/explore/explore.do?structureId=3ke1` (Abruf am 04.12.2013).

[127] R. Pireddu, K. D. Forinash, N. N. Sun, M. P. Martin, S.-S. Sung, B. Alexander, J.-Y. Zhu, W. C. Guida, E. Schonbrunn, S. M. Sebti, and N. J. Lawrence. Pyridylthiazole-Based Ureas as Inhibitors of Rho Associated Protein Kinases (ROCK1 and 2). *Med. Chem. Commun.*, 3(6):699–709, 2012.

[128] H.E. Xu, M.H. Lambert, V.G. Montana, D.J. Parks, S.G. Blanchard, P.J. Brown, D.D. Sternbach, J.M. Lehmann, G.B. Wisely, T.M. Willson, S.A. Kliewer, and M.V. Milburn. Molecular Recognition of Fatty Acids by Peroxisome Proliferator–Activated Receptors. *Mol. Cell*, 3(3):397–403, 1999.

[129] M. Czekaj, S. I. Klein, K. R. Guertin, C. J. Gardner, A. L. Zulli, H. W. Pauls, A. P. Spada, D. L. Cheney, K. D. Brown, D. J. Colussi, V. Chu, R. J. Leadley, and Dunwiddie C. T. Optimization of the beta-Aminoester class of factor Xa inhibitors. part 1: P4 and side-Chain modifications for improved In vitro potency. *Bioorg. Med. Chem. Lett.*, 12(12):1667–1670, 2002.

[130] K. R. Guertin, C. J. Gardner, S. I. Klein, A. L. Zulli, M. Czekaj, Y. Gong, A. P. Spada, D. L. Cheney, S. Maignan, J. P. Guilloteau, K. D. Brown, D. J. Colussi, V. Chu, C. L. Heran, S. R. Morgan, R. G. Bentley, C. T. Dunwiddie, R. J. Leadley, and H. W. Pauls. Optimization of the beta-Aminoester class of factor Xa inhibitors. part 2: Identification of FXV673 as a potent and selective inhibitor with excellent In vivo anticoagulant activity. *Bioorg. Med. Chem. Lett.*, 12(12):1671–1674, 2002.

[131] T. E. Petrova, E. Y. Bezsudnova, B. D. Dorokhov, E. S. Slutskaya, K. M. Polyakov, P. V. Dorovatovskiy, N. V. Ravin, K. G. Skryabin, M. V. Kovalchuk, and V. O. Popov. Expression, purification, crystallization and preliminary crystallographic analysis of a thermostable DNA ligase from the archaeon *Thermococcus sibiricus*. *Acta Cryst.*, 68(2):163–165, 2012.

[132] Nadine Schneider, Gudrun Lange, Sally Hindle, Robert Klein, and Matthias Rarey. A consistent description of HYdrogen bond and DEhydration energies in protein-ligand complexes: methods behind the HYDE scoring function. *J. Comput.-Aided Mol. Des.*, 27(1):15–29, 2013.

[133] BioSolveIT GmbH. Leadit. `http://www.biosolveit.de/LeadIT/` (Abruf am 01.04.2013).

[134] S. Urbaczek, A. Kolodzik, J. R. Fischer, T. Lippert, S. Heuser, I. Groth, T. Schulz-Gasch, and M. Rarey. NAOMI: On the Almost Trivial Task of Reading Molecules from Different File formats. *J. Chem. Inf. Model.*, 51(12):3199–3207, 2011.

Anhang

Anhang A

A Benutzung der Software

Im Rahmen dieser Dissertation sind mehrere Programme zur Analyse und Generierung von Molekülkonformationen entstanden. Im folgenden Kapitel wird die Benutzung der einzelnen Programme beschrieben. Die Reihenfolge, in der die Programme beschrieben werden, folgt dabei einem typischen Arbeitsablauf zur Analyse und Generierung von Konformationen:

1. Analyse des Moleküls mit dem *TorsionAnalyzer* und gegebenenfalls Anpassung der Torsionsbibliothek,
2. Generierung von Konformationen mit dem Kommandozeilenprogramm *Confect*,
3. Berechnung der TFD-Werte für die generierten Konformationen mit dem Kommandozeilenprogramm *TFDCalculator*,
4. Überprüfung der eingestellten Torsionswinkel der generierten Konformationen mit dem Kommandozeilenprogramm *TorsionChecker*.

Für eine kleinere Menge an Eingabemolekülen können alle oben beschriebenen Schritte auch komplett mit dem *TorsionAnalyzer* durchgeführt werden. Die Kommandozeilentools eignen sich vor allem für große Datensätze und die Nutzung von Computerclustern.

A.1. TorsionAnalyzer

Der TorsionAnalyzer ist ein graphisches Softwarewerkzeug zur Anzeige, Erstellung und Bearbeitung von Torsionsbibliotheken, zur automatischen Generierung und Analyse von Torsionshistogrammen und zur Analyse

von Molekülkonformationen. Die Benutzung des TorsionAnalyzers wurde bereits in Abschnitt 4.4 beschrieben. Mit dem folgenden Beispielaufruf lässt sich der TorsionAnalyzer von der Kommandozeile aus starten:

```
./torsionanalyzer --torlib torsionLibrary.xml \
--molecule 1dwd_ref.sdf
```

Beim Aufruf können optional eine Torsionsbibliothek im XML-Format und ein Molekül im MOL2- oder SDF-Format direkt mit übergeben werden. Wird keine Torsionsbibliothek angegeben, wird die Standardbibliothek, die im selben Verzeichnis wie der TorsionAnalyzer liegt, verwendet.

Optionen:

`--help`
Verfügbare Optionen anzeigen

`--torlib`
Torsionsbibliothek im XML-Format

`--molecule`
Zu analysierende Liganden im MOL2- oder SDF Format

A.2. Confect

Mit dem Kommandozeilenprogramm `Confect` lassen sich Konformationen für ein oder mehrere Moleküle im MOL2- oder SDF-Format generieren. Das Programm wird wie folgt aufgerufen:

```
./confect --torLib torsionlibrary.xml --mol 1dwd_ref.sdf \
--conformations 1dwd_confs.sdf --statusFile results.csv
```

Beim Aufruf muss die Datei mit dem Eingabemolekül, die Torsionsbibliothek im XML-Format, der Ausgabedateiname für die generierten Konformationen im MOL2- oder SDF-Format und der Ausgabedateiname für eine Statusdatei im CSV-Format angegeben werden. Die Statusdatei enthält Informationen zur Konformationsgenerierung wie zum Beispiel den Dateinamen des Eingabemoleküls und die verwendeten Optionen (siehe auch Abbildung A.1). Beim Aufruf des Programms können noch mehrere zusätzliche Optionen angegeben werden, die im Folgenden beschrieben werden.

Obligatorisch:

`--torLib`
Torsionsbibliothek im XML-Format

`--mol`
Molekül(e) im MOL2- oder SDF-Format

`--conformations`
Ausgabedatei mit generierten Konformationen im MOL2- oder SDF-Format

`--statusFile`
Ausgabedatei mit Informationen zur Konformationsgenerierung im CSV-Format

Optionen:

`--help`
Verfügbare Optionen anzeigen

`--quality`
Qualitätsstufe [1, 2, 21, 3, 4, 41, 5, 6]
Voreingestellt: 1

`--clustering`
Schwellenwert für das TFD- oder rmsd-Clustering
Voreingestellt: 0,01

`--optimierung`
Anschalten der Optimierung [0: aus, 1: an]
Voreingestellt: 1

`--ringConfs`
Ringkonformationen generieren [0: nein, 1: ja]
Voreingestellt: 1

`--initRing`
Ringkonformationen generieren und Ringkonformation des Eingabemoleküls beibehalten [0: nein, 1: ja]
Voreingestellt: 1

`--writeInput`
Ausgabe des Eingabemoleküls [0: nein, 1: ja]
Voreingestellt: 1

`--recalcHydrogens`
Koordinaten der Wasserstoffatome neu berechnen [0: nein, 1: ja]
Voreingestellt: 1

`--defaultProt`
Standardprotonierung berechnen [0: nein, 1: ja]
Voreingestellt: 1

`--tfdCluster`
TFD-Clustering für Konformationen benutzen [0: nein, 1: ja]
Voreingestellt: 1

`--rmsdCluster`
rmsd-Clustering für Konformationen benutzen [0: nein, 1: ja]
Voreingestellt: 0

`--nofConf`
Maximale Anzahl zu generierender Konformationen [-1: Standardeinstellungen der Qualitätsstufen]
Voreingestellt: -1

A.3. TFDCalculator

Mit dem Kommandozeilenprogramm `TFDCalculator` lassen sich TFD-Werte zwischen ein oder mehreren Referenzmolekülen im MOL2- oder SDF-Format und den dazugehörigen Konformationen im MOL2- oder SDF-Format berechnen. Das Programm wird wie folgt aufgerufen:

```
./tfdcalculator --refMol 1dwd_ref.sdf \
--conformations 1dwd_confs.sdf --output results.csv
```

Beim Aufruf des Programms müssen die Dateien für Referenzmolekül und Konformationen, sowie der Ausgabedateiname für die Ergebnisse im CSV-Format angegeben werden. In der Ausgabedatei stehen unter anderem

```
Name Number_of_Conformations Time(ms)
1dwd_ref 67 4177
#-------------------------------------------------------
#Input File: 1dwd_ref.sdf
#OutputFile: 1dwd_confs.sdf
#Use tfd clustering 1
#Use rmsd clustering 0
#cluster Threshold: 0.01
#Quality Level: 1
#Optimization: 1
#Generate Ring Conformations: 1
#Use initial ring conformation: 1
#Write Input Structure: 1
#Recalculate Hydrogens: 1
#Default Protonation: 1
#Max Number of conformations: 250
#Total running time (ms): 4326
```

Abbildung A.1.: Ausgabedatei von *Confect* mit Informationen zur Konformationsgenerierung

Namen und TF der Referenzmoleküle und Konformationen, die berechneten TFD-Werte und die dafür benötigte Zeit (siehe auch Abbildung A.2).

Obligatorisch:

`--refMol`
Referenzmolekül(e) im MOL2- oder SDF-Format

`--conformations`
Konformationen im MOL2- oder SDF-Format

`--output`
Ergebnisdatei im CSV-Format

Optionen:

`--help`
Verfügbare Optionen anzeigen

```
ID, Name, TF, Contribution, TFD, Time(ms)
0, 1dwd_ref, ( 20.19 91.54 242.38 0.00 180.01 152.15 163.66 ↩
    185.94 187.96 193.89 232.80 130.21 69.41 0.35 55.90 0.54 ↩
    0.80 ), ( 0.00 0.16 0.36 0.00 0.00 0.36 0.16 1.00 0.89 ↩
    0.36 0.89 0.63 0.63 0.05 0.05 0.01 0.17 )
0, 1dwd_ref, ( 20.19 91.54 242.38 0.00 180.01 152.15 163.66 ↩
    185.94 187.96 193.89 232.80 130.21 69.41 0.35 55.90 0.54 ↩
    0.80 ), ( 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ↩
    0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ), 0, 1
1, 1dwd_1, ( 303.59 90.00 175.44 11.34 180.00 0.00 117.77 ↩
    4.87 5.35 89.77 210.00 180.00 180.00 55.90 0.35 0.80 0.54 ↩
    ), ( -0.00 0.00 0.13 0.00 0.00 0.30 0.04 0.99 0.88 0.21 ↩
    0.11 0.18 0.39 0.02 0.02 0.00 0.00 ), 0.572473, 1
2, 1dwd_2, ( 303.59 270.00 175.44 11.34 180.00 0.00 117.77 ↩
    4.87 5.35 89.77 210.00 180.00 180.00 55.90 0.35 0.80 0.54 ↩
    ), ( -0.00 0.00 0.13 0.00 0.00 0.30 0.04 0.99 0.88 0.21 ↩
    0.11 0.18 0.39 0.02 0.02 0.00 0.00 ), 0.572475, 1
3, 1dwd_3, ( 303.58 90.00 175.44 11.34 180.00 0.00 117.77 ↩
    4.87 5.35 89.77 210.00 0.00 180.00 55.90 0.35 0.80 0.54 ),↩
     ( -0.00 0.00 0.13 0.00 0.00 0.30 0.04 0.99 0.88 0.21 0.11↩
     0.46 0.39 0.02 0.02 0.00 0.00 ), 0.621833, 1
...
```

Abbildung A.2.: Ausgabedatei von *TFDCalculator* mit den Ergebnissen der TFD-Berechnung

A.4. TorsionChecker

Mit dem Kommandozeilenprogramm `TorsionChecker` lassen sich die Torsionswinkel ein oder mehrerer Moleküle im MOL2- oder SDF-Format analysieren. Das Programm wird wie folgt aufgerufen:

```
./torsionchecker --torLib torsionLibrary.xml \
--mols 1dwd_confs.sdf --results output.csv
```

Beim Aufruf des Programms muss die Datei mit den Eingabemolekülen, sowie die Torsionsbibliothek im XML-Format und der Ausgabedateiname für die Ergebnisse im CSV-Format angegeben werden. In der Ausgabedatei stehen die Namen der analysierten Moleküle, sowie für jedes Molekül die Anzahl der als *häufig* (Green), *grenzwertig* (Orange) und *selten* (Red) klassifizierten Torsionswinkel (siehe auch Abbildung A.3).

```
Name, Green, Orange, Red
1dwd_ref, 8, 2, 1
1dwd_1, 11, 0, 0
1dwd_2, 11, 0, 0
1dwd_3, 11, 0, 0
1dwd_4, 11, 0, 0
...
1dwd_31, 7, 4, 0
1dwd_32, 9, 2, 0
1dwd_33, 8, 3, 0
1dwd_34, 7, 4, 0
...
```

Abbildung A.3.: Ausgabedatei von *TorsionChecker*

Obligatorisch:

`--torLib`
Torsionsbibliothek im XML-Format

`--mol`
Molekül(e) im MOL2- oder SDF-Format

`--results`
Ergebnisdatei im CSV-Format

Optionen:

`--help`
Verfügbare Optionen anzeigen

B Anhang B
Implementierung

Im Rahmen dieser Dissertation wurden drei Kommandozeilenprogramme (*Confect, TFDCalculator, TorsionChecker*) ein graphisches Softwarewerkzeug (*TorsionAnalyzer*) und zwei Softwarebibliotheken (*TorsionLib, Conformations*) entwickelt. Die komplette Software wurde in C und C++ implementiert und verwendet einige interne und externe Softwarebibliotheken als Abhängigkeiten (Abbildung B.1). Zum Einlesen und Verarbeiten von Moleküldaten wird die am Zentrum für Bioinformatik in der Arbeitsgruppe Algorithmisches Molekulares Design entwickelte Softwarebibliothek *NAOMI* [134] benutzt.

Die TorsionLib-Bibliothek beinhaltet Komponenten zum Arbeiten mit Torsionsbibliotheken, zum Generieren und Analysieren von Torsionshistogrammen und zur Analyse von Molekülkonformationen. Zum Einlesen und Schreiben der in XML definierten Torsionsbibliotheken wird die externe Softwarebibliothek *libXML2* (`http://www.xmlsoft.org`) benutzt. Der SMARTS-Matching-Algorithmus [99], der bei der Zuordnung von Torsionssignaturen zu rotierbaren Bindungen benutzt wird, ist in der internen Softwarebibliothek *SMARTS* implementiert. Die SMARTS-Bibliothek wurde ebenfalls am Zentrum für Bioinformatik in der Arbeitsgruppe Algorithmisches Molekulares Design entwickelt.

Die Conformations-Bibliothek beinhaltet Methoden zum Generieren von Molekülkonformationen. Sie benutzt dabei die TorsionLib- und die NAOMI-Bibliothek.

Die drei Kommandozeilenprogramme und der graphische TorsionAnalyzer wurden als eigenständige *Tools* entwickelt. Alle vier Programme benutzen die Softwarebibliotheken TorsionLib, Conformtions und NAOMI. Der

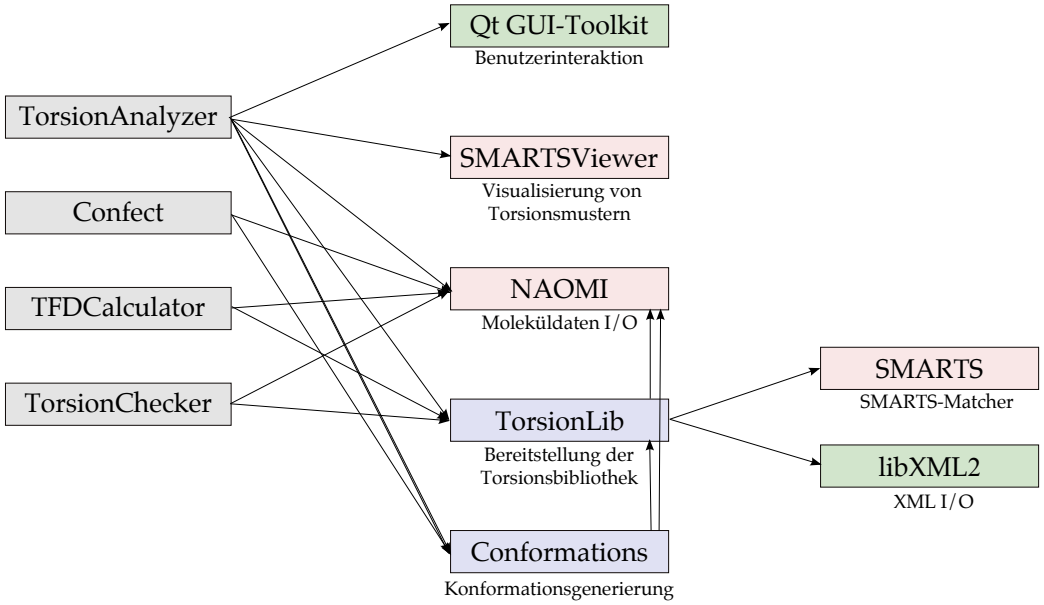


Abbildung B.1.: Übersicht der entwickelten Programme und Softwarebibliotheken und deren Abhängigkeit von internen und externen Softwarebibliotheken. Grün: externe Softwarebibliotheken, rosa: interne Softwarebibliotheken, blau: interne, vom Autor dieser Arbeit implementierte Softwarebibliotheken.

TorsionAnalyzer benötigt zusätzlich noch das externe *Qt GUI-Toolkit*, welches eine große Anzahl an GUI-Widgets und Benutzerinteraktionen zur Verfügung stellt und die interne Softwarebibliothek *SMARTSViewer* [109] zum Generieren der SMARTSViewer-Bilder im Dialogfenster zum Editieren von Torsionssignaturen. Die SMARTSViewer-Bibliothek wurde ebenfalls am Zentrum für Bioinformatik in der Arbeitsgruppe Algorithmisches Molekulares Design entwickelt.

Anhang C

C XML-Schema der Torsionsbibliothek

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- definition of attributes -->
<xs:attribute name="smarts" type="xs:string"/>
<xs:attribute name="value" type="xs:double"/>
<xs:attribute name="name" type="xs:string"/>
<xs:attribute name="id1" type="xs:string"/>
<xs:attribute name="id2" type="xs:string"/>
<xs:attribute name="count" type="xs:integer"/>
<xs:attribute name="text"  type="xs:string"/>

<!-- definition of complex elements -->
<xs:element name="bin">
  <xs:complexType>
    <xs:attribute ref="count" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="histogram">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="bin" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="histogram2">
  <xs:complexType>
```

```
30      <xs:sequence>
31        <xs:element ref="bin" maxOccurs="unbounded"/>
32      </xs:sequence>
33    </xs:complexType>
34  </xs:element>

36  <xs:element name="histogram_folded">
37    <xs:complexType>
38      <xs:sequence>
39        <xs:element ref="bin" maxOccurs="unbounded"/>
40      </xs:sequence>
41    </xs:complexType>
42  </xs:element>

44  <xs:element name="histogram2_folded">
45    <xs:complexType>
46      <xs:sequence>
47        <xs:element ref="bin" maxOccurs="unbounded"/>
48      </xs:sequence>
49    </xs:complexType>
50  </xs:element>

52  <xs:element name="histogram_shifted">
53    <xs:complexType>
54      <xs:sequence>
55        <xs:element ref="bin" maxOccurs="unbounded"/>
56      </xs:sequence>
57    </xs:complexType>
58  </xs:element>

60  <xs:element name="histogram2_shifted">
61    <xs:complexType>
62      <xs:sequence>
63        <xs:element ref="bin" maxOccurs="unbounded"/>
64      </xs:sequence>
65    </xs:complexType>
66  </xs:element>

68  <xs:element name="scattergram">
69    <xs:complexType>
70      <xs:sequence>
71        <xs:element ref="bin" maxOccurs="unbounded"/>
72      </xs:sequence>
73      <xs:attribute ref="smarts" use="required"/>
74      <xs:attribute name="orientation" type="xs:int" default="↩
          -1"/>
75      <xs:attribute name="dependency" type="xs:boolean" default↩
          ="false"/>
76    </xs:complexType>
77  </xs:element>
```

```
<xs:element name="note">
  <xs:complexType>
    <xs:attribute ref="text"/>
  </xs:complexType>
</xs:element>

<xs:element name="angle">
  <xs:complexType>
    <xs:attribute ref="value" use="required"/>
    <xs:attribute name="tolerance1" type="xs:double" default="10.0"/>
    <xs:attribute name="tolerance2" type="xs:double" default="20.0"/>
    <xs:attribute name="score" type="xs:double" default="0.0"/>
  </xs:complexType>
</xs:element>

<xs:element name="angleList">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="angle" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="reducingRule">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="angle"/>
      <xs:element ref="angleList"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="dependentTorsion">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="reducingRule" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="scattergram" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="smarts" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="torsionRule">
  <xs:complexType>
    <xs:sequence>
```

```
124        <xs:element ref="angleList"/>
125        <xs:element ref="histogram" minOccurs="0"/>
126        <xs:element ref="histogram_folded" minOccurs="0"/>
127        <xs:element ref="histogram_shifted" minOccurs="0"/>
128        <xs:element ref="histogram2" minOccurs="0"/>
129        <xs:element ref="histogram2_folded" minOccurs="0"/>
130        <xs:element ref="histogram2_shifted" minOccurs="0"/>
131        <xs:element ref="scattergram" minOccurs="0" maxOccurs="↩
              unbounded"/>
132        <xs:element ref="dependentTorsion" minOccurs="0" ↩
              maxOccurs="unbounded"/>
133        <xs:element ref="note" minOccurs="0"/>
134      </xs:sequence>
135      <xs:attribute ref="smarts" use="required"/>
136      <xs:attribute name="disabled" type="xs:boolean" default="↩
            false"/>
137    </xs:complexType>
138  </xs:element>

140  <xs:element name="hierarchySubClass">
141    <xs:complexType>
142      <xs:sequence>
143       <xs:element ref="torsionRule" minOccurs="0" maxOccurs="↩
             unbounded"/>
144        <xs:element ref="hierarchySubClass" minOccurs="0" ↩
              maxOccurs="unbounded"/>
145        <xs:element ref="note" minOccurs="0"/>
146      </xs:sequence>
147      <xs:attribute ref="name" use="optional"/>
148      <xs:attribute ref="smarts" use="required"/>
149    </xs:complexType>
150  </xs:element>

152  <xs:element name="hierarchyClass">
153    <xs:complexType>
154      <xs:sequence>
155        <xs:element ref="torsionRule" minOccurs="0" maxOccurs="↩
              unbounded"/>
156        <xs:element ref="hierarchySubClass" minOccurs="0" ↩
              maxOccurs="unbounded"/>
157        <xs:element ref="note" minOccurs="0"/>
158      </xs:sequence>
159      <xs:attribute ref="id1" use="required"/>
160      <xs:attribute ref="id2" use="required"/>
161      <xs:attribute ref="name" use="required"/>
162    </xs:complexType>
163  </xs:element>

165  <xs:element name="library">
166    <xs:complexType>
```

```
    <xs:sequence>
      <xs:element ref="hierarchyClass" minOccurs="0" ↩
         maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

D Anhang D Veröffentlichungen

D.1. Veröffentlichungen in wissenschaftlichen Zeitschriften

1. C. Schärfer, T. Schulz-Gasch, J. Hert, B. Schulz, T. Inhester, M. Stahl, and M. Rarey. Confect: Conformations from an Expert Collection of Torsion Patterns. *in Vorbereitung*

2. C. Schärfer, T. Schulz-Gasch, H.-C. Ehrlich, W. Guba, M. Rarey, and M. Stahl. Torsion Angle Preferences in Drug-like Chemical Space: A Comprehensive Guide. *J. Med. Chem.*, 56(5):2016–2028, 2013

3. T. Schulz-Gasch, C. Schärfer, W. Guba, and M. Rarey. TFD: TorsionFingerprints As a New Measure To Compare Small Molecule Conformations. *J. Chem. Inf. Model.*, 52(6):1499–1512, 2012.

D.2. Vorträge

1. C. Schärfer, T. Schulz-Gasch, M. Rarey, and W. Guba. TorsionAnalyzer: Interactive Analysis and Exploration of the Conformational Space. 244th ACS National Meeting, Philadelphia, PA.

D.3. Poster

1. C. Schärfer, T. Schulz-Gasch, and M. Rarey. Systematic Search for Pairwise Dependencies of Torsion Angles. 7th German Conference on Chemoinformatics, Goslar
2. C. Schärfer, T. Schulz-Gasch, M. Rarey, and W. Guba. Torsion Fingerprint Deviation: A Novel Measure to Compare Small Molecule Conformations. 244th ACS National Meeting, Philadelphia, PA.
3. C. Schärfer, T. Schulz-Gasch, and M. Rarey. TorsionAnalyzer: Explore the Conformational Space. 8th German Conference on Chemoinformatics, Goslar