

Entwicklung eines strukturbasierten de-novo-Design-Verfahrens und chemischer Fragmenträume für den rationalen Wirkstoffentwurf

Dissertation

zur

Erlangung des Doktorgrades
der Naturwissenschaften
(Dr. rer. nat.)

dem

Department Informatik
der Universität Hamburg
vorgelegt von

Jörg Degen
aus Reutlingen

Hamburg 2009

Vom Department Informatik der Universität Hamburg
als Dissertation angenommen am

04.02.2009

Erstgutachter:

Prof. Dr. M. Rarey

Zweitgutachter:

Prof. Dr. W. Menzel

Drittgutachter:

Prof. Dr. J. Gasteiger

Tag der mündlichen Prüfung:

04.02.2009

Meinen Eltern

Die Untersuchungen zu der vorliegenden Arbeit wurden auf Anregung von Herrn Prof. Dr. M. Rarey am Zentrum für Bioinformatik der Universität Hamburg in der Zeit von Dezember 2002 bis März 2008 durchgeführt.

An dieser Stelle gilt mein herzlicher Dank:

- Prof. Dr. Matthias Rarey für die äußerst spannende Themenstellung, die intensive Betreuung und ständige Diskussionsbereitschaft und für die Unterstützung, die Arbeit frühzeitig und regelmäßig international vorzustellen und diskutieren zu können. Besonderer Dank gilt ihm auch für die gebotene Freiheit in der Gestaltung der Arbeit sowie die nachhaltige wissenschaftliche Förderung.
- Prof. Dr. Wolfgang Menzel für die Bereitschaft zur Übernahme des Koreferats.
- Prof. Dr. Johann Gasteiger für die Bereitschaft zur externen Begutachtung der Arbeit und die wissenschaftliche Unterstützung.
- allen Partnern des NovoBench-Projektes für die gute Zusammenarbeit und die vielen interessanten Projekttreffen. Insbesondere gilt mein Dank Dr. Andrea Zaliani für den intensiven und fortwährenden wissenschaftlichen Austausch sowie den ambitionierten und langjährigen Einsatz von und für FlexNovo.
- Ingo Reulecke unter anderem für regelmäßige und hilfreiche Unterhaltungen über die mannigfaltigen Aspekte unserer Themengebiete, die langjährige Etablierung des Chemiker-Büros und unzählige Unternehmungen.
- Patrick Maaß für viele anregende Gespräche, nicht nur über Fragmente und Shredding. Seine Fähigkeit und Bereitschaft, sich im Detail mit programmiertechnischen Herausforderungen (anderer) auseinanderzusetzen ist unübertroffen.
- Juri Pärn, Robert Fischer und Tobias Lippert für die Verwendung und Weiterentwicklung der entstandenen Software-Bibliothek für Fragmenträume.
- allen Mitgliedern der Arbeitsgruppe für die regelmäßigen und spannenden Diskussionen, die stetige Hilfsbereitschaft sowie die ausgezeichnete Arbeitsatmosphäre.
- allen Mitarbeitern am Zentrum für Bioinformatik für das äußerst angenehme Arbeitsklima und die unermüdliche Unterstützung bei der Meisterung aller (verwaltungs)technischen Tücken.
- Dr. Christof Wegscheid-Gerlach für die produktive und intensive Zusammenarbeit bei der Erstellung der BRICS-Fragmenträume.
- Dr. Judith Günther für den entscheidenden Hinweis auf die bevorstehende Einrichtung des Zentrums für Bioinformatik an der Universität Hamburg, ohne den diese Arbeit höchstwahrscheinlich nicht zustande gekommen wäre.
- BioSolveIT für die Bereitstellung der Software-Bibliothek.
- meiner Familie für die ständige und bedingungslose Unterstützung und den starken Zusammenhalt in schwierigen Zeiten.
- Sabrina für die unglaubliche Motivationsleistung, Geduld, Toleranz und überhaupt alles andere was war und noch kommt.

Aus der vorliegenden Arbeit sind die folgenden Publikationen und Tagungsbeiträge hervorgegangen:

Publikationen:

- Degen, J. & Rarey, M. (2006). FlexNovo: Structure-Based Searching in Large Fragment Spaces. *ChemMedChem*, **1** (8), 854–868.
- Rarey, M., Degen, J. & Reulecke, I. (2007). Docking and Scoring for Structure-based Drug Design. In: *Bioinformatics - From Genomes to Therapies*, (Lengauer, T. Hrsg.), Bd. II: Molecular Interactions, S. 541–599, Wiley-VCH Verlag GmbH & Co. KGaA Weinheim.
- Pärn, J., Degen, J. & Rarey, M. (2007). Exploring Fragment Spaces Under Multiple Physicochemical Constraints, *J Comput Aided Mol Design*, **21** (6), 327–340.
- Degen, J., Wegscheid-Gerlach, C., Zaliani, A. & Rarey, M. (2008). On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces, *ChemMedChem*, **3** (10), 1503–1507.

Tagungsbeiträge:

- Degen, J. & Rarey, M.: Poster präsentiert beim *4th International Workshop on New Approaches in Drug Design & Discovery*, Schloss Rauischholzhausen, 21.–24. März 2005.
- Degen, J. & Rarey, M.: Vortrag präsentiert bei der *7th International Conference on Chemical Structures*, Noordwijkerhout, Niederlande, 05.–09. Juni 2005.
- Degen, J. & Rarey, M.: Vortrag präsentiert beim *231st ACS National Meeting*, Atlanta, Georgia, USA, 26.–30. März 2006.
- Degen, J. & Rarey, M.: Poster und Kurzvortrag präsentiert beim *Keystone Symposium on Structure Based Drug Discovery*, Whistler, British-Columbia, Kanada, 04.–09. April 2006.
- Degen, J. & Rarey, M.: Vortrag präsentiert beim *MPG-Koç Workshop on Protein Bioinformatics*, Koç Universität, Istanbul, Türkei, 06.–08. Sep. 2006.
- Lemmen, C., Claussen, H., Gastreich, M., Degen, J., Pärn, J. & Rarey, M.: Vortrag präsentiert beim *233rd ACS National Meeting*, Chicago, Illinois, USA, 25.–29. März 2007.
- Degen, J. & Rarey, M.: Poster präsentiert bei der *3rd German Conference on Chemoinformatics*, Goslar, 11.–13. November 2007.

- Wegscheid-Gerlach, C., Degen, J., Zaliani, A. & Rarey, M.: Poster präsentiert bei der *8th International Conference on Chemical Structures*, Noordwijkerhout, Niederlande, 01.–05. Juni 2008.
- Pärn, J., Degen, J. & Rarey, M.: Vortrag präsentiert beim *236th ACS National Meeting*, Philadelphia, Pennsylvania, USA, 17.–21. August 2008.
- Zaliani, A., Degen, J. & Rarey, M.: Vortrag präsentiert beim *236th ACS National Meeting*, Philadelphia, Pennsylvania, USA, 17.–21. August 2008.
- Degen, J., Wegscheid-Gerlach, C., Zaliani, A. & Rarey, M.: Vortrag präsentiert beim *236th ACS National Meeting*, Philadelphia, Pennsylvania, USA, 17.–21. August 2008.
- Wegscheid-Gerlach, C., Degen, J., Briem, H., Rarey, M. & Zaliani, A.: Poster präsentiert beim *237th ACS National Meeting*, Salt Lake City, Utah, USA, 22.–26. März 2009.
- Wegscheid-Gerlach, C., Degen, J., Briem, H., Rarey, M. & Zaliani, A.: Vortrag präsentiert beim *237th ACS National Meeting*, Salt Lake City, Utah, USA, 22.–26. März 2009.

Kurzfassung

Seit der Entwicklung von Medikamenten im Rahmen des rationalen Wirkstoffentwurfs stehen Wissenschaftler vor der Herausforderung, kleine Moleküle mit vielversprechenden strukturellen Motiven zu identifizieren und daraus neue pharmakologisch aktive Verbindungen zu entwickeln. Die Anzahl der prinzipiell möglichen Verbindungen, das chemische Universum, ist allerdings unendlich groß. Gleichzeitig ist die Abdeckung eines hinreichend großen Teils des chemischen Raums notwendig, um neue und bis dato unerforschte Strukturen zu identifizieren. Durch die Verwendung von Fragmenträumen lässt sich ein großer Teil des chemischen Raums in einer kombinatorischen Weise repräsentieren. Durch die Kombination mit computergestützten Verfahren für das de-novo-Design lassen sich gleichzeitig effiziente Suchalgorithmen realisieren.

Die Ableitung von physikochemischen und geometrischen Randbedingungen aus der Bindetasche eines Proteins erlaubt dabei eine umfangreichere Behandlung spezifischer Aspekte der Leitstruktursuche und -optimierung, denen mit typischen ligandbasierten Methoden nur unzureichend Rechnung getragen werden kann. Die meisten Programme für strukturbasiertes de-novo-Design verwenden allerdings abstrahierte Bausteine zur Erzeugung von Molekülen und berücksichtigen deshalb essenzielle Gesichtspunkte der synthetischen Zugänglichkeit sowie pharmakologische Kriterien nur unzureichend.

Das im Rahmen dieser Arbeit entwickelte Software-Werkzeug FlexNovo arbeitet hingegen direkt mit Fragmenten und adaptiert die chemischen Modelle für molekulare Wechselwirkungen und strukturelle Flexibilität sowie den inkrementellen Aufbaualgorithmus des Docking-Programms FlexX. Dabei erzeugt FlexNovo Moleküle in der Bindetasche eines Proteins unter Berücksichtigung multipler Randbedingungen. Dazu zählen Pharmakophore, physikochemische Eigenschaften, Kriterien für die Geometrie der erzeugten Bindungsmodi sowie Parameter zur Steuerung der strukturellen Diversität.

Mit FlexNovo werden vielseitige pharmazeutisch relevante Anwendungsstudien durchgeführt. In allen Fällen kann FlexNovo Moleküle erzeugen, die eine hohe Komplementarität zur jeweiligen Proteinbindetasche aufweisen und nahezu alle wichtigen Pharmakophor-Bedingungen erfüllen. Vergleiche zu bekannten Inhibitoren zeigen, dass zentrale chemische Motive reproduziert und sinnvolle strukturelle Ersetzungen vorgeschlagen werden. Die Analysen der physikochemischen Eigenschaften und Werte für die synthetische Zugänglichkeit ergeben, dass die Ergebnisse wichtige pharmakologische Eigenschaften erfüllen und synthetisch leicht herstellbar sind.

Auf Basis der gewonnenen Erkenntnisse wird ein verbessertes Regelwerk für die Modellierung chemischer Fragmenträume entworfen. Der neu entwickelte BRICS-Ansatz erweitert und präzisiert bestehende Konzepte in vielerlei Hinsicht und wird zur Erstellung mehrerer Fragmenträume verwendet. Diese zeichnen sich durch eine breite Anwendbarkeit aus und die zugrunde liegenden Fragmente repräsentieren sowohl bekannte Inhibitoren als auch kommerziell sowie öffentlich zugängliche Moleküle wesentlich besser, als dies mit bisherigen Ansätzen möglich ist.

Das im Rahmen dieser Arbeit entwickelte Software-Paket FlexNovo und die BRICS-Fragmenträume werden bereits erfolgreich von mehreren akademischen Partnern sowie in der pharmazeutischen Industrie eingesetzt.

Abstract

Ever since the first rational approaches for the discovery of lead structures were applied, it has been a challenge for scientists to assess, generate and combine promising structural motifs to form new and potent chemical entities. In this context, researchers are confronted with the tremendous size of the chemical universe. At the same time, the coverage of a reasonable amount of chemical space is crucial for the identification of novel and so-far unknown structures. Fragment spaces allow for modeling of the chemical space in a combinatorial way. In combination with computational methods for de novo design, this enables for using efficient search strategies.

In the context of lead identification and optimization, the deduction of physicochemical and geometrical constraints from a protein's active site allows for a much more elaborate consideration of all the different aspects in the context of structure generation than this would be possible with ligand-based approaches. However, most of the programs for structure-based de novo design employ only very general rules for the generation of compounds. Thus, they cannot appropriately consider the aspects of synthetic accessibility and pharmacologically relevant information.

By contrast to earlier approaches for de novo design, the FlexNovo approach presented here treats fragments in a more comprehensive way. It works directly on fragments themselves, rather than using abstract molecular structures. FlexNovo is based on the molecular-docking software FlexX, employs its incremental construction algorithm, and uses the same chemical models and scoring functions. Additionally, several filters based on pharmacophore and geometric information as well as physicochemical property and diversity filter criteria are integrated directly in the build-up process.

FlexNovo has been used to design potential inhibitors for several targets of pharmaceutical interest. The results obtained show that FlexNovo is able to generate diverse sets of molecules that are highly complementary to the individual target proteins. These molecules exhibit drug-like properties, are synthetically well accessible and have reasonable predicted binding orientations. The results, including those for an automated similarity analysis, indicate that FlexNovo reproduces structural motifs and offers reasonable structural alternatives. At the same time, the predicted binding modes mimic the corresponding ones from known active structures.

Based on these findings, a new and improved set of rules has been developed for the generation of fragments and the specification of the corresponding fragment spaces. The newly developed approach BRICS incorporates the underlying medicinal chemistry concepts more comprehensively with respect to multiple aspects. BRICS has been used to generate several fragment spaces that can serve as basis for different molecular design objectives. The BRICS fragment spaces significantly raise the quality of the results with respect to the similarity of the generated molecules compared to existing inhibitors or compounds from vendor catalogues.

The FlexNovo software package and the BRICS fragment spaces have been successfully used and are readily integrated in the research process of several academic groups as well as partners from pharmaceutical industry.

Inhaltsverzeichnis

1	Einleitung und Problemstellung	1
2	Grundlegende Aspekte des Wirkstoffentwurfs	7
2.1	Der Protein-Ligand-Komplex	7
2.1.1	Strukturaufklärung	8
2.1.2	Chemisches Gleichgewicht	8
2.1.3	Molekulare Wechselwirkungen	10
2.1.4	Energetische Betrachtungen	11
2.2	Suche und Optimierung von Leitstrukturen	12
2.2.1	Das Konzept der Bioisosterie	13
2.2.2	Aktivität und Selektivität	14
2.2.3	Pharmakologische Eigenschaften	15
2.3	Rationaler Wirkstoffentwurf	15
3	Ansätze für das de-novo-Design aus der Literatur	17
3.1	Ansätze ohne Verwendung der Rezeptorstruktur	18
3.2	Ansätze mit Verwendung der Rezeptorstruktur	22
3.2.1	Gitterbasierte Ansätze	22
3.2.2	Verfahren zum Platzieren und Kombinieren	23
3.2.3	Algorithmen zum sequenziellen Aufbau	24
3.2.4	Simulationsmethoden und randomisierte Verfahren	25
3.2.5	Genetische Algorithmen und evolutionäre Programmierung	27
3.3	Vergleich der Verfahren für das de-novo-Design	28
3.4	Ansätze zur Identifikation und Auswahl struktureller Bausteine	30
4	Modellierung	33
4.1	Das molekulare Docking-Programm FlexX	33
4.1.1	Repräsentation von organischen Molekülen	33
4.1.2	Die Behandlung von flexiblen Molekülen	34
4.1.3	Das Wechselwirkungsmodell	34
4.1.4	Die Bewertungsfunktion	37
4.2	Chemische Fragmenträume	38
4.2.1	Retrosynthetische Zerlegung von Molekülen	38
4.2.2	Shredding-Strategie	38
4.2.3	Von einzelnen Fragmenten zum Fragmentraum	42
4.2.4	Modellierung chemischer Reaktionen	43
4.2.5	Berücksichtigung der synthetischen Zugänglichkeit	43

4.3	Der FlexNovo-Ansatz	44
4.3.1	Entwicklungskriterien	45
4.3.2	Initialisierung von Fragmenten	45
4.3.3	Physikochemische und geometrische Eigenschaften	47
4.3.4	Erzeugung neuer Fragmente und Moleküle	48
4.3.5	Pharmakophore	49
5	Algorithmische Konzepte	51
5.1	Globale Vorgehensweisen beim de-novo-Design	51
5.1.1	Strukturelle Variation durch Verwendung von Molekül-Bausteinen	51
5.1.2	Alternative Verfahren zur Realisierung struktureller Variabilität .	54
5.1.3	Die verwendeten molekularen Bausteine	56
5.2	Platzierung von Fragmenten in der Bindetasche	56
5.2.1	Flexibles Docking	56
5.2.2	Alternative Verfahren zum Docking flexibler Fragmente	57
5.2.3	Der verwendete Platzierungsalgorithmus	58
5.3	Inkrementeller Aufbau von Molekülen	59
5.3.1	Die Greedy-Methode	60
5.3.2	Globale Randbedingungen	64
5.3.3	Erweiterung einer Menge von Teillösungen	64
5.3.4	Schritte zum Anbau eines Fragments	65
5.3.5	Die finale Lösungsliste	72
5.3.6	Algorithmische Aspekte	73
6	Erzeugung und Charakterisierung von Fragmenträumen	75
6.1	Erzeugung von Fragmenträumen	75
6.1.1	Die verwendete Datengrundlage	75
6.1.2	Shredding-Prozedur	76
6.1.3	Die verwendeten Shredding-Regelwerke	77
6.1.4	Target-spezifische Fragmente	80
6.1.5	Fragmente aus Wirkstoff- und Molekül-Katalogen	80
6.1.6	Die BRICS-Fragment-Datensätze	85
6.2	Charakterisierung von Fragmenträumen	86
6.2.1	Grundlegende Eigenschaften von Fragmenträumen	86
6.2.2	Target-spezifische Fragmenträume	86
6.2.3	Performance der BRICS-Fragmenträume	94
7	Strukturbasierte Suche in Fragmenträumen	101
7.1	Parametrisierung von FlexNovo	101
7.1.1	Abhängigkeiten von FlexX	101
7.1.2	Algorithmische Parameter	102
7.2	Fallstudien	102
7.2.1	Überblick über die Testsysteme	103
7.2.2	Aufbereitung und Prozessierung der Eingabedaten	104

7.2.3	Durchführung der Rechnungen	104
7.2.4	Suche nach alternativen Leitstrukturen	105
7.2.5	Berücksichtigung einer komplexen Pharmakophor-Definition	125
7.2.6	Serendipity	129
7.3	Laufzeitverhalten	131
8	Zusammenfassung und Ausblick	137
	Anhang	147
A	Die Benutzungsschnittstelle von FlexNovo	149
A.1	Einführung in FlexNovo	149
A.2	FlexNovo-Einstellungen	167
A.2.1	Konfiguration	167
A.2.2	Algorithmische Parameter	168
A.2.3	Grafik-Einstellungen	172
A.3	FlexNovo-Kommandos	176
A.3.1	Arbeiten mit Fragmenträumen (FRAGSPACE Untermenü)	176
A.3.2	Arbeiten mit Filter-Spezifikationen (FILTER Untermenü)	184
A.3.3	Platzierung von Fragmenten (PLACING Untermenü)	185
A.3.4	Strukturbasierte Suche (BUILDUP Untermenü)	187
A.3.5	Zeichnen und Auflisten wichtiger Platzierungsinformationen	191
A.4	FlexNovo-Beispieldateien	200
A.4.1	Konfigurationsdatei	200
A.4.2	Skripte für typische Berechnungen	209
B	Implementierung	213
B.1	Die Programmbibliothek	213
B.1.1	Das Fragspace-Modul	213
B.1.2	Das Filter-Modul	215
B.1.3	Das Docking-Modul	215
B.1.4	Das Buildup-Modul	216
B.2	FlexNovo-Versionshistorie	217
C	Daten	227
C.1	Shredding-Regelwerke	227
C.1.1	Das BRICS-Modell	227
C.1.2	Die RECAP-Reimplementierung	234
C.2	FlexNovo-Berechnungen	239
C.2.1	Filter-SMARTS für reaktive chemische Gruppen	239
C.2.2	Pharmakophor-Spezifikationen	240
C.3	Enumeration der target-spezifischen Fragmenträume	242
	Literaturverzeichnis	243

Inhaltsverzeichnis

Abkürzungsverzeichnis

1D	Eine Dimension, eindimensional
2D	Zwei Dimensionen, zweidimensional
3D	Drei Dimensionen, dreidimensional
Abb.	Abbildung
ADMET	Adsorption, Distribution, Metabolisierung, Exkretion, Toxizität
ANSI	<i>American National Standards Institute</i>
ASCII	<i>American Standard Code for Information Interchange</i>
ATP	Adenosin Triphosphat
CPU	<i>Central processing unit</i> (Hauptprozessor)
DUD	<i>Directory of useful decoys</i> (Datenbank nützlicher „Testverbindungen“) Huang <i>et al.</i> [2006]
HTS	<i>High-throughput screening</i> (Hochdurchsatz-Screening)
Kap.	Kapitel
MCS	<i>Maximum common subgraph</i> (Maximal gemeinsamer Teilgraph)
MD	Molekulardynamik
NMR	<i>Nuclear magnetic resonance</i> (Kernmagnetische Resonanz)
PDB	<i>Protein Data Bank</i> (Proteindatenbank) [Berman <i>et al.</i> , 2000; Bernstein <i>et al.</i> , 1977]
PSAS	<i>Polar solvent accessible surface area</i> (Polare lösungsmittelzugängliche Oberfläche)
QSAR	<i>Quantitative structure-activity relationship</i> (Quant. Struktur-Wirkungs-Beziehung)
QSPR	<i>Quantitative structure-property relationship</i> (Quant. Struktureigenschafts-Beziehung)
SMARTS	<i>Smiles arbitrary target specification</i> (Sprache zur Spezifikation chemischer Substrukturen) [Daylight, 2007]
SMILES	<i>Simplified molecular input line entry system</i> (Vereinfachte Darstellung einer chemischen Struktur als ASCII-Zeichenkette) [Weininger, 1988]
Tab.	Tabelle
WDI	<i>World Drug Index</i> (Datenbank entwickelter und vermarkteter Arzneistoffe)
WW	Wechselwirkung(en)

Abkürzungen für die verwendeten Target-Klassen

ACE	Angiotensin-konvertierendes Enzym
AChE	Acetylcholinesterase
ADA	Adenosine Deaminase
ALR2	Aldose Reduktase
AmpC	AmpC Betalaktamase
AR	Androgen Rezeptor
CDK2	Cyclin-abhängige Kinase 2
COMT	Catechol <i>O</i> -methyltransferase
COX-1	Cyclooxygenase-1
COX-2	Cyclooxygenase-2
DHFR	Dihydrofolat-Reduktase
ER	Estrogenrezeptor
FGFr1	Fibroblast Wachstumsfaktor Rezeptor Kinase
FXa	Faktor Xa
GART	Glycinamid-ribonucleotid Transformylase
GPB	Glycogen-Phosphorylase b
GR	Glucocorticoid Rezeptor
HIVPR	HIV-Protease
HIVRT	HIV reverse Transkriptase
HMGR	Hydroxymethylglutaryl-CoA Reduktase
HSP90	Humanes Heat Shock Protein 90
InhA	Enoyl-ACP-Reduktase
MR	Mineralocorticoid receptor
NA	Neuraminidase
P38	p38 Mitogen-aktiviertes Protein
PARP	Poly(ADP-Ribose) Polymerase
PDE5	Phosphodiesterase-5
PDGFRb	<i>Platelet derived growth factor receptor kinase</i>
PNP	Purin-Nucleosid-Phosphorylase
PPARg	Peroxisomen Proliferator-aktivierte Rezeptor c
PR	Progesteronrezeptor
RXRa	Retinoid X Rezeptor alpha
SAHH	S-adenosylhomocystein Hydrolase
SRC	Tyrosinkinase SRC
TK	Thymidinkinase
VEGFR2	<i>Vascular endothelial growth factor receptor</i>

Abbildungsverzeichnis

2.1	Einige Beispiele für isostere Ersetzungen	14
4.1	Zerlegung des Inhibitors Gefitinib in Komponenten	35
4.2	Wechselwirkungsgeometrien in FlexX	36
4.3	Geometrisches Modell einer Wasserstoffbrücken-Wechselwirkung	36
4.4	Zerlegung des Inhibitors Sorafenib in Fragmente	39
4.5	Retrosynthetische Zerlegung des Inhibitors Celecoxib	41
4.6	Die FTree-Fragment-Prototypen	42
4.7	Operationen auf Fragmenten und Link-Atomen	44
4.8	Pharmakophor für Dihydrofolat-Reduktase.	49
5.1	Beispiele alternativer Verfahren zur Strukturgenerierung	54
5.2	Suchbaum während des inkrementellen Aufbaus von Molekülen	60
5.3	Geometrieangepassung bei der Fragmentverknüpfung	67
5.4	Übertragung der Platzierungs- und Wechselwirkungsinformation	69
6.1	Die BRICS-Shredding-Subgraphen in vereinfachter Darstellung	78
6.2	Die RECAP-Fragmentierungsregeln	79
6.3	Die BRICS-Fragment-Prototypen	81
6.4	Verteilung des Molekulargewichts für die Fragmente des WDI und ZINC	84
6.5	Verteilung der Link-Typen für BRICS und RECAP	85
6.6	Verteilungen der FTree-Ähnlichkeitswerte für die enumerierten Moleküle	91
6.7	Beispiele ähnlicher Molekül-Inhibitor-Paare	93
6.8	Performance der BRICS_4k und RECAP_4k Fragmenträume	96
6.9	Performance der BRICS_4k, BRICS_9k und BRICS_20k Fragmenträume	97
6.10	Rekonstruktion von Sorafenib aus den erstellten Fragmenträumen	99
7.1	Schematische Darstellung der Bindetasche von DHFR	107
7.2	Repräsentative Ergebnisse für DHFR	107
7.3	Schematische Darstellung der ATP-Bindetasche von CDK2	109
7.4	Repräsentative Ergebnisse für CDK2	109
7.5	Bindungsmodus-Modell von Traxler für Kinase-Inhibitoren	110
7.6	Schematische Darstellung der Bindetasche von COX-2	112
7.7	Ausgewähltes Ergebnis für COX-2	112
7.8	Schematische Darstellung der Bindetasche von ER	113
7.9	Repräsentative Ergebnisse für ER	114
7.10	Repräsentative Ergebnisse der FlexNovo-Rechnungen	115

Abbildungsverzeichnis

7.11	FTree-Ähnlichkeitsanalysen der FlexNovo-Ergebnisse	117
7.12	Verteilungen der physikochemischen Eigenschaften der Ergebnisse	119
7.13	Verteilungen der Werte der synthetischen Zugänglichkeit	122
7.14	Schematische Darstellung der Bindetasche von Thrombin	126
7.15	Repräsentative Ergebnisse der Rechnungen für Thrombin	127
7.16	<i>Serendipity</i> im strukturbasierten molekularen Design	130
7.17	Darstellung der Bindetasche und des Bindungsmodus von Staurosporin	130
7.18	Laufzeitanteile der Teilberechnungen des Aufbaualgorithmus	133
7.19	Fragment-Statistik des Aufbaualgorithmus	134
7.20	Platzierungsstatistik des Aufbaualgorithmus	135
7.21	Speicherbedarf	135

Tabellenverzeichnis

4.1	Wechselwirkungstypen in FlexX	35
4.2	Phasenmodell der Initialisierung in FlexNovo	46
6.1	Aufbereitete Datenquellen zur Erzeugung der Fragmenträume	76
6.2	Shredding-Ergebnisse des World Drug Index (WDI)	82
6.3	Shredding-Ergebnisse des ZINC 'drug-like' Datensatzes	82
6.4	Shredding-Ergebnisse und Eigenschaften der Fragmenträume	88
6.5	Ergebnisse der Enumeration	89
6.6	Grundlegende Eigenschaften der BRICS-Fragmenträume	95
7.1	Daten und Referenzen zu den vier Targets	105
7.2	Diversitätsanalyse der Ergebnisse	121
7.3	Statistische Informationen zu den Rechnungen	124
7.4	Daten und Referenzen zu Thrombin	125
B.1	Bestandteile des Fragspace-Moduls	213
B.2	Bestandteile des Filter-Moduls	215
B.3	Bestandteile des Docking-Moduls	215
B.4	Bestandteile des Buildup-Moduls	216

Tabellenverzeichnis

1 Einleitung und Problemstellung

Die Identifikation pharmakologisch wirksamer Substanzen ist seit jeher von großer Bedeutung und hat mitunter nachhaltige Auswirkungen auf die Gesellschaft [Nicolaou & Montagnon, 2008]. Neben der Isolierung und Charakterisierung von Stoffen aus der Natur ist deren synthetische Herstellung, mit ihren mannigfaltigen Aspekten, zu einem großen Teil dem Enthusiasmus synthetischer Chemiker zu verdanken [Nicolaou, 2003]. Neben der Faszination für chemische Strukturen ist insbesondere die gezielte Entwicklung von neuen Molekülen und deren Einfluss auf die Funktion von Makromolekülen im Organismus für den späteren Einsatz als Medikament das Ziel umfassender akademischer und industrieller Forschung [Hanessian, 2006]. Solch ein Vorhaben stellt allerdings eine immense interdisziplinäre Herausforderung dar [Corey *et al.*, 2007], die das folgende Zitat sehr treffend beschreibt: „Toward this end, mere chemical knowledge or technology is often insufficient and basic research through interdisciplinary collaboration with scientists in other fields is needed“ (Ryoji Noyori, Nobel-Vortrag, 8. Dezember 2001).

Zum Verständnis von Wechselwirkungen zwischen Makromolekülen und niedermolekularen Verbindungen haben die Entwicklung und stetige Verbesserung der Methoden zur experimentellen Strukturaufklärung einen entscheidenden Beitrag geleistet. Mit diesen Verfahren lässt sich nahezu atomare Auflösung erreichen, wodurch sich detaillierte Einblicke auf molekulare Wechselwirkungen und die grundlegenden Phänomene chemischer und struktureller Komplementarität eröffnen. Die Bestimmung von inzwischen mehr als 50000 Proteinstrukturen unterschiedlichster Art und Funktion [Berman, 2008] – hauptsächlich durch Röntgenstrukturanalyse [Ladd & Palmer, 2003] aber zunehmend auch durch NMR-Spektroskopie [Cavanagh *et al.*, 1996] und mithilfe von Kryoelektronenmikroskopie-Verfahren [Kühlbrandt & Williams, 1999] – ist auch auf die Tatsache zurückzuführen, dass viele dieser Strukturen wichtige Funktionen im Organismus erfüllen und in Wechselwirkung mit pharmakologisch aktiven Verbindungen treten. Die durch Röntgenstrukturanalyse bestimmte Anordnung des Proteins im Kristall entspricht dabei in guter Näherung der des aktiven Proteins in Lösung [Knegtel *et al.*, 1991; Doscher & Richards, 1963]. Diese Information ist von großer Bedeutung für das strukturelle Verständnis von molekularer Erkennung und für die Charakterisierung pharmakologisch relevanter Makromoleküle.

Die Verwendung der dreidimensionalen Struktur eines Proteins ist im Rahmen des gezielten Entwurfs eines neuen Wirkstoffes zum festen Bestandteil geworden [Böhm & Klebe, 1996]. Aus der Proteinstruktur werden geometrische und physikochemische Informationen abgeleitet, die für die Identifikation neuer niedermolekularer Verbindungen genutzt werden können. Dabei macht man sich die Erkenntnis zunutze, die bereits vor über 100 Jahren von Emil Fischer formuliert wurde [Fischer, 1894]: Dem-

nach lässt sich die Beziehung von Inhibitor und Protein analog eines „Schlüssel-Schloss-Prinzips“ beschreiben, wonach ein Ligand (Schlüssel) an ein Protein (Schloss) bindet, sofern die erforderliche strukturelle (und physikochemische) Komplementarität vorhanden ist. Neben der dreidimensionalen Struktur eines Proteins können allerdings auch bekannte Inhibitoren als Basis für die Identifikation neuer Wirkstoffkandidaten dienen. In diesem Fall wird von einer analogen Hypothese ausgegangen, nach der sich aus den Strukturen der Liganden Informationen ableiten lassen, aus denen Rückschlüsse auf die biologische Aktivität strukturell verwandter Moleküle gezogen werden können [Klebe, 1998].

Trotz der Verwendung vielseitiger komplementärer Informationen im Rahmen des *rationalen Wirkstoffentwurfs* ist die erfolgreiche Entwicklung eines neuen Medikaments ein zeitaufwändiger und kostenintensiver Prozess, der sich leicht über ein Jahrzehnt erstrecken und Investitionen von mehreren Hundert Millionen Euro erfordern kann [Petsko, 1996]. Der Entwicklungsprozess gliedert sich in mehrere Phasen [Pritchard *et al.*, 2003]. Zu Beginn steht dabei die Identifikation einer niedermolekularen Verbindung, der *Leitstruktur*, die bereits eine biologische Wirkung zeigt und sich in anschließenden Phasen hinsichtlich der Affinität und Selektivität sowie verschiedener weiterer Eigenschaften optimieren lässt [Böhm *et al.*, 2002]. Im Zusammenhang pharmazeutisch industrieller Forschung kommt dabei der Entwicklung neuer (patentierbarer) chemischer Verbindungen (englisch: *new chemical entities*) eine zentrale Bedeutung zu [Baxendale *et al.*, 2007]. Die schnelle und aussagekräftige Identifikation potenzieller Leitstrukturen ist deshalb von immenser Wichtigkeit, um bereits zu Beginn der Entwicklung eines Medikaments neue und viel versprechende Strukturvorschläge berücksichtigen zu können [Kubinyi, 2003]. Dabei werden an vielen Stellen des Entwicklungsprozesses, und insbesondere bei der Identifikation und Optimierung von Leitstrukturen, computergestützte Verfahren eingesetzt [Jorgensen, 2004; Martin, 2006].

Die Identifikation neuer und insbesondere aussichtsreicher Leitstrukturen gestaltet sich äußerst schwierig, da das zugrunde liegende „chemische Universum“ praktisch unendlich groß ist. Dabei sind die für Leitstrukturen in Frage kommenden niedermolekularen Verbindungen nicht gleichmäßig innerhalb dieses Universums verteilt, sondern lassen sich in abgeschlossenen aber nicht zwangsläufig miteinander verbundenen Bereichen lokalisieren [Dobson, 2004; Lipinski & Hopkins, 2004]. Allerdings liegt die geschätzte Zahl kleiner organischer Moleküle – bestehend aus bis zu 30 Schweratomen unter Berücksichtigung der Elemente Kohlenstoff, Stickstoff, Sauerstoff und Schwefel – in der Größenordnung von etwa 10^{60} [Bohacek *et al.*, 1996]. Andere Schätzungen gehen von etwa 10^{100} potenzielle Moleküle aus [Walters *et al.*, 1998]. Bei Betrachtung dieser Zahlen wird deutlich, dass keine jetzige oder absehbare Technologie eine derart große Menge an Molekülen in expliziter Form handhaben kann.

Aus diesem Grund sind *Screening-Verfahren*, die existierende Datenbanken oder Bibliotheken von Molekülen sequenziell bewerten, in diesem Zusammenhang nur begrenzt einsetzbar [Gribbon & Sewing, 2005]. Screening-Technologien dienen zunächst der Identifikation einer begrenzten Zahl an Molekülen (*Hits*) aus einer Menge von typischerweise mehreren Millionen Verbindungen. Diese Hits werden gegebenenfalls in einem anschließenden mehrstufigen Prozess zu potenziellen Leitstrukturen weiterentwickelt. Abgesehen

von den technischen und qualitativen Grenzen solcher Verfahren ist die Identifikation neuer Moleküle damit prinzipiell nicht möglich.

Anstatt der sequenziellen Durchmusterung verhältnismäßig kleiner Bibliotheken bietet sich alternativ die kombinatorische Modellierung des entsprechenden chemischen Raums an. Dabei hat sich die Verwendung molekularer Bausteine (*Fragmente*) im Laufe des vergangenen Jahrzehnts als komplementärer Ansatz etabliert [Hubbard & Wilsbacher, 2007; Hajduk & Greer, 2007]. Auf der Basis dieser Fragmente lassen sich virtuelle Bibliotheken (*Fragmenträume*) formulieren, aus denen anhand eines Regelwerks kombinatorisch neue Moleküle erzeugt werden können [Leach & Hann, 2000; Mauser & Stahl, 2007]. Solche Fragmenträume sind um viele Größenordnungen größer als „klassische“ Bibliotheken und erlauben dadurch eine wesentlich bessere Abdeckung des chemischen Raums [Rarey, 2001].

Aufgrund ihrer Struktur erlauben Fragmenträume die Umsetzung effizienter Strategien zur Erzeugung neuer Moleküle (*de-novo-Design*). Dabei haben die zugrunde liegenden Fragmente und das Regelwerk zu deren Verknüpfung einen wesentlichen Einfluss auf die Qualität der Ergebnisse. Die kombinatorische Struktur dieser Fragmenträume wird bisher ausschließlich von Verfahren zum ligandbasierten de-novo-Design ausgenutzt, d.h. ohne dabei die dreidimensionale Struktur der Bindetasche zu berücksichtigen [Schneider *et al.*, 2000; Rarey, 2001; Fechner & Schneider, 2006; Pärn *et al.*, 2007; Hartenfeller *et al.*, 2008; Dey & Caffisch, 2008; Schüller *et al.*, 2008]. Im Gegensatz dazu erlaubt die Ableitung von physikochemischen und geometrischen Randbedingungen aus der Bindetasche eines Proteins die wesentlich umfangreichere Behandlung von spezifischen Aspekten im Rahmen der Leitstruktursuche und -optimierung, denen mit typischen ligandbasierten Methoden nur unzureichend Rechnung getragen werden kann.

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung eines Software-Werkzeugs für strukturbasiertes de-novo-Design auf der Basis chemischer Fragmenträume und dessen Anwendung in unterschiedlichen Szenarien sowie mit der Entwicklung eines Ansatzes zur Ableitung und Charakterisierung molekularer Bausteine für den rationellen Wirkstoffentwurf. Sie liefert damit einen Beitrag zur Beurteilung der Anwendbarkeit automatisierter de-novo-Design-Verfahren und für die Modellierung und Evaluierung der zugrunde liegenden Fragmente und Regelwerke.

Für die Entwicklung eines neuen computergestützten Verfahrens für den strukturbasierten Wirkstoffentwurf lassen sich in diesem Zusammenhang vielfältige Anforderungen formulieren. Diese können zunächst in voneinander unabhängige, aber gleich wichtige Kategorien unterteilt werden. Zum einen betrifft dies die Anforderungen an die verwendeten chemischen Modelle und den zugrunde liegenden Fragmentraum, von denen die Vorhersagekraft des Verfahrens maßgeblich abhängt. Zum anderen spielt auch die programmiertechnische Umsetzung und insbesondere die Gestaltung der Benutzungsschnittstelle des Programms eine Rolle. Die letztendliche Akzeptanz und Verwendbarkeit der resultierenden Methode hängt in entscheidender Weise von der Qualität der erzeugten Ergebnisse ab.

1 Einleitung und Problemstellung

Die folgenden Aspekte werden dabei in besonderer Weise berücksichtigt:

- Gewährleistung großer Robustheit bezüglich der Verwendung unterschiedlicher Kombinationen von Rezeptoren und Fragmenträumen durch vielfältige, breite, modulare und einfache Anwendbarkeit der Software.
- Erweiterung und Anpassung des bestehenden Dateiformats für Fragmenträume sowie Bereitstellung von Schnittstellen zu weiteren chemoinformatischen Software-Werkzeugen (Corina [Sadowski *et al.*, 1994] und Sylvia [Boda *et al.*, 2007]).
- Adaption umfangreicher und präziser physikochemischer und geometrischer Modelle für die Behandlung der spezifischen Aspekte molekularer Wechselwirkungen zwischen Fragment bzw. Molekül und Protein sowie für die Beschreibung von struktureller Flexibilität.
- Verwendung qualitativ hochwertiger Fragmente durch Formulierung eines spezifischen Regelwerks zu deren Ableitung aus Moleküldatenbanken mit bekannten Inhibitoren sowie kommerziell erhältlichen Verbindungen unter besonderer Berücksichtigung und Konservierung der dreidimensionalen Molekülgeometrie.
- Ableitung fokussierter und leistungsfähiger Fragment-Zusammenstellungen für spezifischen Einsatz mit de-novo-Design-Verfahren unter Berücksichtigung der dreidimensionalen Struktur der Bindetasche.
- Realisierung einer hohen Vorhersagekraft bezüglich der Qualität der erzeugten Moleküle durch Verwendung eines präzisen und speziell auf das strukturbasierte de-novo-Design angepassten Regelwerks zur Verknüpfung von Fragmenten.
- Integration ergänzender und komplementärer Randbedingungen zur Steuerung der physikochemischen Eigenschaften der Moleküle sowie für die Berücksichtigung umfassender zusätzlicher geometrischer Kriterien beim Aufbau von Molekülen in der Proteinbindetasche.

Kapitel 2 enthält eine Einführung in die Grundlagen und zentralen Konzepte im Zusammenhang mit dem rationalen Wirkstoffentwurf. In Kapitel 3 werden in einem Literaturüberblick die existierenden Ansätze für das de-novo-Design im Hinblick auf die verwendeten algorithmischen und physikochemischen Modelle sowie die zugrunde liegenden molekularen Bausteine betrachtet. Kapitel 4 beschreibt die Modellierung des im Rahmen dieser Arbeit entwickelten Software-Werkzeugs sowie die verwendete Strategie zur Ableitung molekularer Bausteine. In Kapitel 5 werden die Algorithmen für das entwickelte de-novo-Design-Verfahren im Detail beschrieben. Die Ergebnisse dieser Arbeit lassen sich in zwei Bereiche unterteilen. Kapitel 6 befasst sich mit der Charakterisierung von fokussierten Fragment-Bibliotheken und daraus abgeleiteten target-spezifischen Fragmenträumen und beschreibt im weiteren Verlauf das im Rahmen dieser Arbeit neu erstellte, verbesserte Modell zur Erzeugung und Modellierung generischer chemischer Fragmenträume. In Kapitel 7 werden die mit dem entwickelten Software-Werkzeug FlexNovo erzielten Ergebnisse für mehrere Fallbeispiele dargestellt und diskutiert sowie das Laufzeitverhalten analysiert. Eine Zusammenfassung der Arbeit ist in Kapitel 8 zu finden. Darin wird auch auf die Grenzen des de-novo-Design-Verfahrens eingegangen und Schlussfolgerungen für die praktische Anwendbarkeit des entwickelten Software-Werkzeugs gezogen. Schließlich werden Erweiterungsmöglichkeiten des entwickelten Ansatzes vorgestellt. In Anhang A wird die Verwendung des entstandenen Software-Werkzeugs anhand eines Tutoriums erläutert und die Benutzungsschnittstelle umfassend vorgestellt und dokumentiert. Die Details der Implementierung sind in Anhang B beschrieben. Anhang C enthält die verwendeten Daten und Eingabedateien für die entwickelten Regelwerke zur Modellierung chemischer Fragmenträume.

1 Einleitung und Problemstellung

2 Grundlegende Aspekte des Wirkstoffentwurfs

So unterschiedlich die Anwendungsszenarien von Arzneimitteln sind, so verschieden sind auch die zugrunde liegenden Mechanismen. Viele Medikamente wirken als *Inhibitor* eines *Enzyms*, d.h., sie unterbinden dessen biokatalytische Funktion zur Umsetzung eines oder mehrerer Substrate zu einem oder mehreren Produkten. Dabei kann der Inhibitor seine Wirkung auf der einen Seite *kompetitiv* entfalten, d.h., in Konkurrenz zum Substrat an das Enzym binden und die katalytische Umsetzung dadurch verhindern. Andererseits kann er *allosterisch* an einer anderen Stelle an das Enzym binden und dadurch eine Veränderung in der Struktur hervorrufen, durch die eine Anlagerung des Substrats verhindert wird.

Eine andere große Klasse von Medikamenten wirken als *Agonisten* eines *Rezeptors*, d.h., sie besetzen eine Bindestelle und bewirken dadurch eine Änderung in der Konformation des Rezeptors oder stabilisieren die „aktive“ Konformation, wodurch in einem anschließenden Prozess eine Reaktion ausgelöst wird. *Antagonisten* verhindern die Anlagerung eines Agonisten oder körpereigenen Liganden an die Bindestelle eines Rezeptors. Ein *inverser Agonist* stabilisiert die inaktive Konformation eines Rezeptors.

Generell kann die Wirkung eines Medikaments durch Ausbildung *reversibler* Wechselwirkungen oder *irreversibler* (stabiler) kovalenter Bindungen erfolgen. Auf letztere soll jedoch nicht weiter eingegangen werden da sie für den Entwurf neuer Wirkstoffe nur eine untergeordnete Rolle spielen. Auch die zahlreichen weiteren Wirkmechanismen werden an dieser Stelle nicht behandelt. Eine umfassende Übersicht ist bei Mutschler zu finden [Mutschler *et al.*, 2008].

2.1 Der Protein-Ligand-Komplex

Das Verständnis molekularer Erkennung wurde maßgeblich durch zwei Forscher geprägt. Emil Fischer formulierte im Jahr 1894 einen berühmten Vergleich, der bildlich besagt, „dass Enzym und Glucosid wie Schloss und Schlüssel zueinander passen müssen, um eine chemische Wirkung aufeinander auszuüben.“ Im Jahr 1913 prägte Paul Ehrlich den Satz: „Corpora non agunt nisi fixata.“¹

Demnach müssen Protein und Ligand geometrisch zueinander passen und der Ligand kann nur dann Einfluss auf die Funktion des Proteins ausüben, wenn beide Strukturen aneinander gebunden sind. Im weiteren Sinne schließt dies auch spätere Theorien von Koshland mit ein, wonach der Ligand Änderungen in der dreidimensionalen Geometrie

¹ „Die Körper wirken nicht, wenn sie nicht verbunden sind.“

des Proteins induziert und erst dadurch eine tatsächliche Bindung zustande kommt. Dieses Phänomen wird als *induced fit* bezeichnet [Koshland & Neet, 1968].

Als Resultat aus der Anlagerung eines Liganden an ein Makromolekül entsteht ein *Komplex*. Zum Verständnis der Bildung und der Eigenschaften von Komplexen haben in entscheidendem Maße die experimentellen Methoden zur Strukturaufklärung beigetragen, ohne die es weder strukturbasierte Verfahren für den Wirkstoffentwurf noch viele der im Folgenden beschriebenen Erkenntnisse geben würde.

2.1.1 Strukturaufklärung

Die stetig wachsende Zahl aufgeklärter Proteinstrukturen geht neben Verfahren zur Kryo-Elektronenmikroskopie [Kühlbrandt & Williams, 1999] ($\approx 1\%$) und zur hochauflösenden NMR-Spektroskopie [Cavanagh *et al.*, 1996] ($\approx 14\%$) hauptsächlich auf die breite Anwendung der Röntgenstrukturanalyse [Ladd & Palmer, 2003] ($\approx 85\%$) zurück (Anteile bezogen auf die Einträge in der PDB [Berman *et al.*, 2000] nach heutigem Stand).

Für die Durchführung der Röntgenstrukturanalyse werden (Protein-)Kristalle verwendet, die sich durch die dreidimensionale periodische Anordnung einer Struktureinheit (*Elementarzelle*) in Form eines Kristallgitters auszeichnen und makroskopisch in Form eines anisotropen, homogenen Körpers (*Kristall*) vorliegen. Beim Auftreffen von Röntgenstrahlung auf solch einen Kristall treten Beugungseffekte auf, da die Ausdehnung des Kristallgitters und die Wellenlänge der Röntgenstrahlung von der gleichen Größenordnung sind. Dabei wird die Röntgenstrahlung an den Elektronenwolken der Atome gebeugt, im Gegensatz zur Neutronenstrahlung, wo die Beugung an den Atomkernen auftritt. Das resultierende Beugungsmuster stellt dabei den Mittelwert aller im Kristall enthaltenen Proteine dar und repräsentiert ebenso inhärente Fehler in der Kristallstruktur, die beispielsweise durch Packungsdefekte zustande kommen.

Anhand des Beugungsbildes kann die Struktur einer Elementarzelle des Kristalls bestimmt werden. Aus der Fouriertransformation erhält man eine Elektronendichteverteilung der Elementarzelle, aus der sich die Anordnung der Atome bestimmen lässt. Aus den Anordnungen wird zunächst ein Molekülmodell generiert, das in mehreren Schritten mithilfe unterschiedlicher Optimierungsverfahren verfeinert wird. Die Position von Wasserstoffatomen kann in der Regel nicht direkt durch die Röntgenstrukturanalyse bestimmt werden. Dafür werden die Informationen des generierten Strukturmodells herangezogen. Die letztendliche Auflösung einer durch Röntgenstrukturanalyse bestimmten Proteinstruktur liegt typischerweise zwischen $1,5\text{\AA}$ und 3\AA . Die Ungenauigkeit liegt demzufolge im Bereich von ein bis zwei Bindungslängen. Bei der Verwendung von Kristallstrukturen für den Entwurf neuer Wirkstoffe muss dies stets bedacht werden.

2.1.2 Chemisches Gleichgewicht

Die Bildung nicht-kovalenter, reversibler Komplexe findet im Allgemeinen in wässriger Lösung statt und lässt sich mit dem *Massenwirkungsgesetz* beschreiben, nach dem der Protein-Ligand-Komplex ($P'L'$) sowie der „freie“ Ligand (L) und das „freie“ Protein (P) im Gleichgewicht vorliegen:



Daraus leitet sich die *Assoziationskonstante* (K_A) bzw. die dazu inverse Dissoziationskonstante (K_D) ab, die sich aus den Verhältnissen der Konzentrationen (durch eckige Klammern symbolisiert) der jeweiligen Spezies ergibt. Dadurch hat K_A bzw. K_D ebenfalls die Einheit einer Konzentration, d.h. Stoffmenge pro Volumen (mol/l bzw. M).

$$K_A = K_D^{-1} = \frac{[P'L']}{[P] \cdot [L]} \quad (2.2)$$

Die Bindung eines Liganden an ein Protein ist umso stärker, je kleiner der Wert für K_D ist. Unter der Annahme, dass die Spezies im thermodynamischen Gleichgewicht vorliegen, kann aus der Gleichgewichtskonstante die *freie Bindungsenthalpie* ΔG gemäß folgender Gleichung bestimmt werden:

$$\Delta G = -RT \ln K_A \quad (2.3)$$

ΔG hängt demnach von der absoluten Temperatur (T in Kelvin) und der Gaskonstante ($R \approx 8,3JK^{-1}mol^{-1}$) ab. Die durch Experimente bestimmten Gleichgewichtskonstanten von Protein-Ligand-Komplexen liegen in den meisten Fällen im Bereich von 10^{-3} bis 10^{-12} M und entsprechen damit freien Bindungsenthalpien von -10 bis -70 kJ/mol bei $T = 298K$. Die Änderung der Bindungskonstante um eine Größenordnung bedeutet eine Änderung in der freien Bindungsenthalpie von $5,7$ kJ/mol [Böhm & Klebe, 1996].

Die Stabilität eines Protein-Ligand-Komplexes hängt in entscheidendem Maße sowohl von den bereits genannten geometrischen Kriterien als auch von physikochemischen und energetischen Aspekten ab. Letztere spiegeln sich in den ausgebildeten Wechselwirkungen (enthalpische Beiträge) zwischen Protein und Ligand sowie in den jeweiligen durch die Komplexbildung bedingten Veränderungen von Protein ($P \rightarrow P'$) und Ligand ($L \rightarrow L'$) (entropische Beiträge) wider.

Die freie Bindungsenthalpie setzt sich aus einem enthalpischen (ΔH) und einem entropischen Anteil ($T\Delta S$) zusammen. Darauf wird in den folgenden Abschnitten noch näher eingegangen.

$$\Delta G = \Delta H - T\Delta S \quad (2.4)$$

Anstatt der Assoziationskonstante bzw. der freien Bindungsenthalpie wird oftmals der IC_{50} -Wert verwendet, der angibt ab welcher Konzentration des Liganden die Aktivität des Proteins um die Hälfte reduziert wird. Der IC_{50} hängt dabei unter anderem von der Konzentration des Proteins und vom Wirkmechanismus ab. Beide Werte stehen miteinander in Zusammenhang, sodass der experimentell relativ einfach bestimmbare IC_{50} gut zum Vergleich der Wirksamkeit zweier Liganden herangezogen werden kann [Cheng & Prusoff, 1973].

2.1.3 Molekulare Wechselwirkungen

Bei der Bildung eines nicht-kovalenten Protein-Ligand-Komplexes treten vor allem so genannte *schwache* intermolekulare Wechselwirkungen auf, wovon die wichtigsten im Folgenden kurz vorgestellt werden.

Eine besondere Rolle spielt in diesem Zusammenhang die Ausbildung von *Wasserstoffbrücken*. Dabei handelt es sich um elektrostatische Wechselwirkungen, die zwischen zwei elektronegativen Atomen auftreten, wovon eines ein kovalent gebundenes Wasserstoffatom (*Wasserstoffbrücken-Donor*) und das andere eine negative Partiaalladung hat (*Wasserstoffbrücken-Akzeptor*). Durch die hohe Elektronegativität des Donors ist die Elektronendichte des Wasserstoffatoms zum Bindungspartner verschoben, wodurch der Akzeptor sich dem Proton stärker nähern kann, als das von der Summe der Van-der-Waals-Radien (s.u.) eigentlich zu erwarten wäre. Dadurch kann das Proton keinem der beiden Schweratome mehr eindeutig zugeordnet werden, es nimmt also eine überbrückende Position ein. In den meisten Fällen treten als Schweratome Stickstoff, Sauerstoff und Schwefel auf, aber auch CH-Gruppen und Halogenatome sind unter speziellen Umständen beteiligt. Wasserstoffbrücken sind kurzreichweitige Wechselwirkungen mit sehr gut charakterisierten Geometrien, die nur geringe Abweichungen zulassen. Aus diesem Grund haben sie eine wesentliche Funktion für die Positionierung eines Liganden in der Bindetasche. Die Stärken von Wasserstoffbrücken können sehr unterschiedlich sein und hängen vor allem davon ab, ob sie ladungsunterstützt ausgebildet werden. Letzteres wird auch als *Salzbrücke* bezeichnet. Diese kann einen sehr hohen Beitrag zur Wechselwirkungsenergie leisten. Bei vielen Protein-Ligand-Komplexen wird die Assoziation zu einem großen Teil durch Salzbrücken bestimmt.

Eine weitere wichtige Kategorie bilden *hydrophobe Wechselwirkungen*, die durch Nachbarschaft unpolarer Aminosäure-Seitenketten des Proteins und lipophiler Gruppen des Liganden zustande kommen. Zu letzteren zählen beispielsweise aliphatische und aromatische Kohlenwasserstoffe, die auch mit Halogenen substituiert sein können sowie viele aromatische, heterozyklische Verbindungen, die keine polaren Wechselwirkungen ausbilden können. Im Gegensatz zu Wasserstoffbrücken haben hydrophobe Wechselwirkungen keine ausgezeichnete Direktionalität. Allerdings gibt es im Fall der Wechselwirkung mit aromatischen Systemen bevorzugte relative Anordnungen der jeweiligen Gruppen. Für große, lipophile Liganden sind hydrophobe Wechselwirkungen oftmals die treibende Kraft bei der Komplexbildung, wobei der Großteil des energetisch günstigen Beitrags tatsächlich auf die Freisetzung von Wassermolekülen zurückzuführen ist. Darauf wird im nächsten Abschnitt noch genauer eingegangen.

Zwischen jedem Paar von Atomen treten unabhängig von dessen Typ *Van-der-Waals-Wechselwirkungen* auf. Sie bestehen aus einem assoziativen und einem repulsiven Anteil und beruhen auf elektromagnetischen Phänomenen. Bei Unterschreitung eines gewissen Abstands (Van-der-Waals-Abstand) überwiegt der repulsive energetische Anteil, der im Wesentlichen aus der Abstoßung der Kerne und der Elektronenhüllen besteht. Energetisch ist der anziehende Anteil der Van-der-Waals-Wechselwirkungen verhältnismäßig schwach, der repulsive Anteil steigt bei fallendem Abstand jedoch sehr schnell an.

2.1.4 Energetische Betrachtungen

Wasser spielt bei der Bildung von Protein-Ligand-Komplexen eine entscheidende Rolle. Wie bereits erwähnt, findet die Komplexbildung im Allgemeinen in wässriger Lösung statt, Protein und Ligand sind dort mit einer Hülle von Lösungsmittel-Molekülen (*Solvathülle*) umgeben. Die Wassermoleküle formen untereinander ein Netzwerk von Wasserstoffbrücken. Einige Wassermoleküle bilden gleichzeitig Wasserstoffbrücken mit dem Protein oder dem Liganden aus. Nicht alle Wassermoleküle sind jedoch vollständig in das Wasserstoffbrücken-Netzwerk integriert, z.B. solche, die hauptsächlich mit lipophilen Bereichen der Proteinoberfläche in Kontakt sind.

Bevor der Komplex gebildet werden kann, muss der Ligand sowohl seine eigene Solvathülle abstreifen als auch Wassermoleküle aus der Bindetasche des Proteins verdrängen (*Desolvatation*). Während dieses Prozesses werden Wasserstoffbrücken gebrochen, was für die Komplexbildung zunächst energetisch ungünstig ist. Durch die Ausbildung von Wechselwirkungen zwischen Protein und Ligand entstehen wiederum energetisch günstige Beiträge. Auch die Verdrängung von nicht optimal positionierten Wassermolekülen aus der Bindetasche hat energetisch einen günstigen Effekt.

Eine Aufrechnung der einzelnen Energiebeiträge ist dabei nur schwer möglich, da es nicht nur auf die Anzahl der Wechselwirkungen sondern auch auf deren Stärke ankommt. Für die letztendliche Gesamt-Energiebilanz der Komplexbildung sind jedoch nicht nur die Desolvatation und die Ausbildung von Wechselwirkungen zwischen Protein und Ligand entscheidend. Neben diesen enthalpischen Beiträgen spielen entropische Beiträge eine große Rolle, die ein Maß für die Unordnung des Systems sind und im weitesten Sinne durch den Verlust von *Freiheitsgraden* zustande kommen. Der Ligand, das Protein und die Wassermoleküle haben im ungebundenen Zustand drei Translations- und eine Vielzahl von Rotations-Freiheitsgraden. Durch die Fixierung des Liganden in der Bindetasche des Proteins gehen die jeweils unabhängigen Translations- und Rotations-Freiheitsgrade verloren. Gleiches gilt für während der Komplexbildung fixierte Wassermoleküle, wohingegen frei gesetzte Wassermoleküle entropisch begünstigt sind.

Letztendlich können sowohl enthalpische als auch entropische Effekte den größeren „negativen“ (günstigen) energetischen Beitrag zu ΔG liefern. Bei der *enthalpiegetriebenen* Bildung von Komplexen resultieren diese hauptsächlich aus energetisch günstigen Wechselwirkungen, wohingegen sie bei der *entropiegetriebenen* Komplexbildung insbesondere durch die Freisetzung von Wassermolekülen, oft in Verbindung mit der Komplexbildung von großen, lipophilen Liganden, zustande kommen.

Trotz der hohen Komplexität der Zusammenhänge lassen sich für den Entwurf neuer Verbindungen einige generelle Regeln ableiten. Demnach binden starre Liganden stärker an ein Protein als flexible, da der Entropieverlust bei der Komplexbildung geringer ausfällt. Auch die Erhöhung der lipophilen Kontaktfläche zwischen Protein und Ligand führt in der Regel zu einer höheren Bindungsaffinität. Die Ausbildung zusätzlicher Wasserstoffbrücken wirkt sich hingegen nicht zwangsläufig günstig auf die Bindungsaffinität eines Liganden aus. Hingegen führt das *vergraben* polarer Gruppen, d.h., die Desolvatation ohne anschließende Ausbildung von Wasserstoffbrücken, immer zu einem Energieverlust und sollte deshalb beim Entwurf neuer Liganden stets berücksichtigt werden.

2.2 Suche und Optimierung von Leitstrukturen

Am Anfang der strukturellen Entwicklung eines neuen Medikaments steht die Identifikation einer oder mehrerer *Leitstrukturen*. Diese zeigen bereits eine gewisse biologische Aktivität, ihnen fehlen aber noch wichtige pharmakologische Eigenschaften für einen therapeutischen Einsatz. Aus diesem Grund sollte die Leitstruktur vielseitige Möglichkeiten zur chemischen Modifikation bieten, damit die Eigenschaften der Verbindung in einem zyklischen Prozess verbessert werden können.

Die *Leitstruktursuche* befasst sich mit der Identifikation potenziell in Frage kommender Moleküle mit bereits vorhandener biologischer Aktivität. Dafür können eine Vielzahl sehr unterschiedlicher Quellen herangezogen werden. So kommt der Verwendung wirksamer Inhaltsstoffe aus Pflanzen und Wirkstoffen von Tieren oder Mikroorganismen historisch und auch heutzutage noch eine große Bedeutung zu. Solche *Naturstoffe* werden jedoch selten selbst als Medikament eingesetzt. Auch die präparative Chemie hat einen großen Anteil an der (oft erst nachträglichen) Identifikation biologisch aktiver Verbindungen [Sneider, 1996].

Neben der gezielten Ableitung bzw. Herstellung von Leitstrukturen werden *Screening-Verfahren* im großen Stil eingesetzt. Im Hochdurchsatz-Screening (HTS) wird eine große Anzahl verfügbarer Moleküle in einem automatisierten Verfahren experimentell auf biologische Aktivität getestet bzw. mithilfe computergestützter Verfahren bewertet. Typische Molekülbibliotheken und Kataloge von kommerziell verfügbaren Verbindungen enthalten mehrere Millionen Moleküle mit steigender Tendenz. Abgesehen von den zahlreichen Erfolgen haben weder HTS noch Verfahren zur Erzeugung großer Mengen an Molekülen mithilfe der kombinatorischen Chemie zu dem ursprünglich antizipierten Durchbruch bei der Wirkstoffentwicklung geführt, was zum Großteil auf die hohen Fehlerraten zurückzuführen ist [Gribbon *et al.*, 2005; Gribbon & Sewing, 2005].

Ein zentraler, offensichtlicher und viel versprechender Ansatzpunkt für die Identifikation einer Leitstruktur ist die Ableitung aus bereits bekannten Inhibitoren oder natürlichen (endogenen) Liganden, d.h., aus körpereigenen Stoffen wie z.B. Hormonen und Neurotransmittern. Die Philosophie spiegelt sich in folgendem berühmten Zitat wider: "The most fruitful basis for the discovery of a new drug is to start with an old drug." (Sir James Black [Raju, 2000])

Im Zuge der *Leitstruktur-Optimierung* werden die Leitstrukturen hinsichtlich ihrer Eigenschaften iterativ verbessert, wobei die verschiedenen strukturellen Motive eines Inhibitors nach ihrer Funktion unterschieden werden müssen: Im Vordergrund steht das eigentliche *Pharmakophor* [Willett, 1995], d.h., die chemischen Gruppen sowie deren räumliche Abstände zueinander, die für die spezifische Bindung des Moleküls an einen Rezeptor hauptsächlich verantwortlich ist. Daneben spielen zusätzliche Gruppen eine zentrale Rolle, die für die Erhöhung der Affinität und der Selektivität der Verbindung wichtig sind. Andere strukturelle Merkmale sind hauptsächlich für pharmakologische und pharmakokinetische Aspekte (Aufnahme, Transport und Verteilung im Organismus)

mus) verantwortlich. Schließlich können Motive enthalten sein, die erst nach chemischer Modifizierung oder Abspaltung einen günstigen Effekt auf die Wirksamkeit haben.

Im Zusammenhang mit der Optimierung der Struktur existieren mannigfaltige Möglichkeiten für chemische Modifikationen, die sich auf die geometrische Form und die physikochemischen Eigenschaften einer Verbindung auswirken. Einzelne Schritte beinhalten beispielsweise den Austausch, die Einführung oder Entfernung von Substituenten an Ringen, den Austausch ganzer chemischer Gruppen, die Einführung oder Entfernung von Heteroatomen in Ringen oder azyklischen Ketten, die Variation von Substituenten an Heteroatomen, die Veränderung der Topologie durch Einführung oder Entfernung von Verzweigungen, die Öffnung oder Schließung von Ringen, die Veränderung der Ringgröße, die Veränderung der Flexibilität von Ringen oder verbrückenden Gruppen durch Einführung flexibler Teilstrukturen, die Variation der Länge aliphatischer Ketten und die Variation bzw. Einführung chiraler Zentren.

Jede Veränderung in der Struktur eines Moleküls hat Auswirkungen auf dessen Geometrie, die dreidimensionale Form sowie die physikochemischen und elektronischen Eigenschaften. Die Optimierung eines Moleküls in einzelnen, voneinander unabhängigen Schritten führt dabei nicht immer zum gewünschten Ziel, da die gegenseitigen Einflüsse der strukturellen Modifikationen dabei nur unzureichend berücksichtigt werden können. Aus diesem Grund zielen Verfahren auf die gleichzeitige Optimierung hinsichtlich mehrerer Kriterien ab. Auf einige grundlegende und zentrale Aspekte im Rahmen struktureller Modifikationen wird in den folgenden Unterkapiteln näher eingegangen.

2.2.1 Das Konzept der Bioisosterie

Eines der zentralen strategischen Konzepte in der Arzneimittelforschung ist die bioisostere Ersetzung von chemischen Gruppen. Streng genommen setzt sich das Konzept aus zwei Anteilen zusammen. Dabei beinhaltet der *isostere* Ersatz die Substitution einer bestimmten Gruppe durch eine verwandte Gruppe mit ähnlichen sterischen und elektronischen Eigenschaften. Beim *bioisosteren* Ersatz bleibt zusätzlich die biologische Wirkung des Moleküls im Wesentlichen erhalten.

Einfache bioisostere Ersetzungen bestehen aus der Substitution einzelner Atome, so z.B. der Austausch von Chlor durch Brom, die beide (bis auf die Größe) praktisch identische Eigenschaften haben oder der Austausch einer Hydroxylgruppe (-OH) durch eine Thiolgruppe (-SH). Eine Zusammenstellung weiterer, auch komplizierterer bzw. größerer isostere Ersetzungen ist in Abbildung 2.1 zu finden.

Gelegentlich treten beim vermeintlichen bioisosteren Ersatz unerwartete Effekte auf, die zum Verlust oder der Umkehrung der Wirkung führen können. Einige Phänomene lassen sich nur durch Aufklärung der 3D-Struktur des Komplexes nachvollziehen. So können zum Beispiel auch Änderungen des Bindungsmodus oder drastische Änderungen bei der Aktivität und/oder Selektivität auftreten. Teilweise heben nachfolgende bioisostere Ersetzungen vorher erreichte Steigerungen der Aktivität oder Selektivität wieder auf. Die resultierenden Effekte beruhen auf komplexen Mechanismen molekularer Wechselwirkungen und lassen sich oftmals nur durch umfangreiche experimentelle Untersuchungen ermitteln.

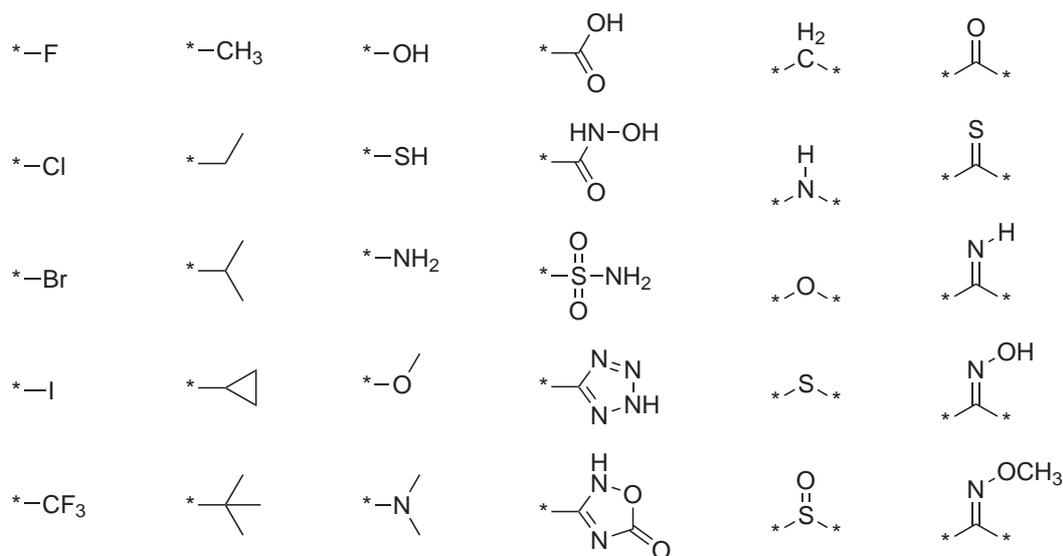


Abbildung 2.1: Einige Beispiele für isostere Ersetzungen von Atomen bzw. chemischen Gruppen. Die Ersetzungen sind jeweils spaltenweise nach fallender Ähnlichkeit angeordnet und zeigen sowohl terminale Substituenten als auch verbrückende Motive. (Adaptiert aus: [Böhm *et al.*, 2002]).

Bioisostere Ersetzungen sind ein beliebtes und weit verbreitetes Mittel in der pharmazeutisch-industriellen Forschung, um bekannte Inhibitoren so abzuwandeln, dass dadurch neue, strukturell hinreichend verschiedene und bezüglich der Wirksamkeit, der Verträglichkeit oder der Selektivität verbesserte Verbindungen erzeugt werden, die gegebenenfalls von bestehenden Patenten nicht erfasst werden und somit als *intellectual property* (IP) zum Patent angemeldet werden können. Die Existenz mehrerer Generationen von Medikamenten für spezielle Indikationsgebiete resultieren oftmals aus solchen Vorgehensweisen.

2.2.2 Aktivität und Selektivität

Durch systematische Variation von Substituenten und strukturellen Motiven soll in der Regel die Aktivität einer Verbindung erhöht werden. Gleichzeitig spielt aber auch die selektive Modulation eines Rezeptors eine entscheidende Rolle, insbesondere für die Stärke eventueller Nebenwirkungen. Dabei muss allerdings zwischen positiven, gewünschten Effekten und negativen, unerwünschten Wirkungen unterschieden werden. Viele Erkenntnisse in diesem Bereich können erst in Zusammenhang mit klinischen Studien gewonnen werden.

Generell lässt sich sagen, dass kleinere und flexiblere Moleküle mit keinen oder wenigen Stereozentren eher eine unspezifischere Wirkung haben. Dahingegen führt die Vergrößerung einer Verbindung, die Einschränkung der Flexibilität und die Einführung von

Stereozentren eher zu einer selektiveren Wirkung, sofern dabei die Aktivität weitgehend beibehalten wird.

Durch den Austausch von polaren mit unpolaren Substituenten oder die Einführung großer, sterisch anspruchsvoller Gruppen lässt sich zum Teil auch die Wirkung verändern, beispielsweise von einem Agonisten zu einem Antagonisten.

2.2.3 Pharmakologische Eigenschaften

Neben Faktoren wie Aktivität und Selektivität, die man als *Pharmakodynamik* bezeichnet, spielen für die tatsächliche Verwendung eines Inhibitors als Medikament weitere Eigenschaften eine entscheidende Rolle. Dabei handelt es sich um Aspekte der Resorption, Verteilung, Metabolisierung und Ausscheidung (*ADME-Parameter*), die unter dem Begriff *Pharmakokinetik* zusammengefasst werden. Im weiteren Sinne zählen dazu auch Aspekte der Toxizität einer Substanz.

Um seine Wirkung entfalten zu können, muss das Medikament vom Ort der Aufnahme oder der Applikation zum eigentlichen Wirkort gelangen. Dies geschieht auf komplizierten Wegen und mithilfe unterschiedlicher Transportmechanismen im Organismus. Eine zentrale und wichtige Eigenschaft ist in diesem Zusammenhang die Lipophilie des Moleküls, von der sowohl die Aufnahme als auch die Verteilung, Metabolisierung und Ausscheidung entscheidend beeinflusst wird. Prinzipiell ist die Aufnahme umso besser, je höher die Lipophilie einer Verbindung ist.

Eine gute Aufnahme der Verbindung alleine ist jedoch nicht ausreichend. So werden große, lipophile Verbindungen zwar hervorragend resorbiert, aber relativ schnell wieder ausgeschieden, d.h. die *Bioverfügbarkeit* einer solchen Verbindung ist entsprechend niedrig. Aus diesem Grund spielt eine gewisse Löslichkeit in wässrigen Phasen gleichermaßen eine wichtige Rolle. Für eine ausreichende *Wirkdauer* ist unter anderem die Stabilität der Verbindung bezüglich metabolischer Abbaureaktionen entscheidend. Diese kann durch die Einführung entsprechender Substituenten vergrößert werden, allerdings müssen dabei auch die Einflüsse auf andere Eigenschaften berücksichtigt werden.

Je nach Anwendungsgebiet und Wirkort einer Verbindung unterscheiden sich die Anforderungen an die physikochemischen und strukturellen Eigenschaften drastisch. Experimentell ist die Bestimmung pharmakokinetischer Eigenschaften mit großem Aufwand verbunden. Für die erfolgreiche Entwicklung eines neuen Medikaments sollten sie deshalb möglichst früh beim Entwurf mit einbezogen werden.

2.3 Rationaler Wirkstoffentwurf

Bei der Entwicklung eines neuen Medikaments müssen eine Vielzahl von Aspekten berücksichtigt werden. Beim rationalen Entwurf neuer Verbindungen werden Informationen von existierenden aktiven wie inaktiven Verbindungen herangezogen, um daraus Modelle und Hypothesen über den Einfluss struktureller, geometrischer und physikochemischer Parameter auf die Wirksamkeit von Molekülen abzuleiten.

Durch dreidimensionale Überlagerung von Liganden lassen sich gegebenenfalls wechselwirkende Gruppen identifizieren, die für die Bindung an ein Protein von besonderer

Wichtigkeit sind. Die Typen und Abstände dieser Gruppen lassen sich mithilfe eines *Pharmakophors* beschreiben. Daneben existieren eine Vielzahl weiterer Abstraktionsmodelle. Die Ableitung quantitativer Modelle ist dabei von besonderer Bedeutung, da durch eine entsprechende Parametrisierung für unterschiedliche chemische Gruppen unter Umständen die Vorhersage der Eigenschaften abgewandelter Verbindungen ermöglicht wird. Quantitative Modelle zur Vorhersage der Aktivität (QSAR) bzw. der Eigenschaften (QSPR) von Molekülen spielen eine große Rolle beim rationalen Wirkstoffentwurf.

Im Laufe des letzten Jahrzehnts hat die Verwendung computergestützter Methoden immens an Bedeutung gewonnen. Obwohl sie die experimentelle Verifikation und Validierung nicht ersetzen können, werden sie insbesondere aufgrund ihrer deutlich besseren Zeit- und Kosteneffizienz eingesetzt. Die Qualität der Vorhersagen computergestützter Verfahren hängen in entscheidendem Maße von der Genauigkeit der verwendeten chemischen Modelle, die benötigte Rechendauer hingegen hauptsächlich von den verwendeten Algorithmen und Optimierungsstrategien ab.

Die Identifikation neuer Leitstrukturen und deren Optimierung sind Schlüsselschritte bei der Entwicklung neuer Medikamente [Bleicher *et al.*, 2003]. Die Erzeugung innovativer struktureller Ideen durch Abwandlung bestehender Substitutionsmuster und Ersetzung zentraler Strukturelemente ist dabei von immenser Wichtigkeit. Der *Zufall*, der insbesondere im Rahmen der „frühen“ Wirkstoffentwicklung oft von großer Bedeutung war, ist in diesem Zusammenhang explizit gewünscht.

3 Ansätze für das de-novo-Design aus der Literatur

Die Verfahren für de-novo-Design haben die Erzeugung neuer Moleküle zum Ziel. Die Grenzen zu anderen Methoden im Rahmen der Leitstruktursuche und -optimierung sind allerdings fließend. So ist ein häufiges Anwendungsszenario das Ersetzen von zentralen Strukturelementen bereits existierender Moleküle (*Scaffold hopping*) [Böhm *et al.*, 2004]. Umgekehrt stellt die Dekoration eines oder mehrerer zentraler Strukturelemente (*Cores*) mit passenden Substituenten (*R-Gruppen*) bei der Verwendung *Kombinatorischer Bibliotheken* ein vergleichbar häufig eingesetztes Verfahren dar [Leach *et al.*, 2000].

Das folgende Kapitel befasst sich deshalb neben den Verfahren zum de-novo-Design auch mit Methoden, deren Hauptaufgabe im weiteren Sinne Scaffold Hopping oder die Optimierung kombinatorischer Bibliotheken ist. Auf Programme, die ausschließlich die Bewertung existierender Moleküle zum Ziel haben, oder die Optimierung der Moleküle auf den eigentlichen *Produkten* und nicht auf den molekularen Bausteinen durchführen [Gillet & Nicolotti, 2000], wird an dieser Stelle nicht eingegangen. Stattdessen werden zu Beginn der jeweiligen Abschnitte ausgewählte Übersichtsartikel genannt. Die Verfahren für das de-novo-Design sind ebenfalls mehrfach und aus unterschiedlichen Perspektiven in Übersichtsartikeln erläutert worden [Böhm & Klebe, 1996; Honma, 2003; Lloyd *et al.*, 2004; Dean *et al.*, 2004; Schneider & Fechner, 2005; Dean, 2007; van Drie, 2007; Mauser & Guba, 2008].

Ein entscheidender Teilaspekt, der sowohl thematisch als auch chronologisch vor der eigentlichen Durchführung eines de-novo-Design-Experiments steht, betrifft die Ableitung und Definition der *Randbedingungen*, auf deren Basis die jeweilige Methode neue Moleküle erzeugt. In diesem Zusammenhang lassen sich die Ansätze für das de-novo-Design in zwei Hauptgruppen einteilen:

- Ligandbasiertes de-novo-Design: Ist auf der einen Seite die dreidimensionale Struktur einer Bindetasche nicht bekannt oder soll sie aus bestimmten Gründen nicht verwendet werden, so können die Randbedingungen für die jeweiligen Verfahren auf der Basis bekannter Inhibitoren definiert werden [Johnson & Maggiora, 1990; Dean, 1995; Klebe, 1998].
- Strukturbasiertes de-novo-Design: Auf der anderen Seite ergeben sich die Randbedingungen durch die Verwendung der dreidimensionalen Struktur eines Rezeptors aus den geometrischen, chemischen und energetischen Eigenschaften der jeweiligen Bindetasche [Böhm, 1996a; Kubinyi, 1998a,b].

Einige der vorgestellten Programme arbeiten mit unterschiedlichen Randbedingungen und gehören somit prinzipiell zu beiden Kategorien. In diesem Fall wurden die jeweiligen Verfahren den strukturbasierten Ansätzen zugeordnet. Im Folgenden liegt der Schwerpunkt auf den Verfahren, die die 3D-Struktur der Bindetasche verwenden. Bei der weiteren Kategorisierung wurden die strukturbasierten Ansätze bezüglich der hauptsächlich verwendeten algorithmischen Methode eingeteilt. Bei den ligandbasierten Verfahren hat sich eine Unterteilung nach den verwendeten Randbedingungen als zweckmäßig erwiesen.

Neben den jeweiligen Randbedingungen unterscheiden sich die de-novo-Design-Verfahren insbesondere in Bezug auf verwendeten molekularen Bausteine, die gleichermaßen von zentraler Bedeutung sind [Stahl *et al.*, 2002; Cross *et al.*, 2003]. Nach der Beschreibung existierender Methoden und deren Analyse hinsichtlich Anwendbarkeit und Qualität der erzeugten Ergebnisse folgt eine Zusammenfassung der Ansätze zur Identifikation und Auswahl struktureller Bausteine im Rahmen des computergestützten Wirkstoffentwurfs.

3.1 Ansätze ohne Verwendung der Rezeptorstruktur

Bei Vernachlässigung der Rezeptorstruktur werden die Randbedingungen für die Berechnungen aus bereits existierenden Molekülen abgeleitet (*ligandbasiertes* de-novo-Design). Es wird dabei von der Hypothese ausgegangen, dass sich aus den jeweiligen Strukturren Informationen ableiten lassen, die für die Erzeugung und Bewertung der Ergebnisse hinreichend genau und aussagekräftig sind [Klebe, 1998; Jain, 2004; Lloyd *et al.*, 2004; Hessler *et al.*, 2005; Willett, 2006].

Die Art der aus den Molekülen extrahierten Informationen kann dabei sehr unterschiedlich sein. Typischerweise werden aus einzelnen oder mehreren Inhibitoren ein oder mehrere *Deskriptoren* abgeleitet [Johnson & Maggiora, 1990; Dean, 1995; Rouvray, 1995]. Die An- bzw. Abwesenheit bestimmter struktureller Motive kann dabei durch eindimensionale Bitvektoren (*Fingerprints*) repräsentiert werden [Brown & Martin, 1996]. Relative oder räumliche Anordnungen funktioneller Gruppen werden mithilfe höherdimensionaler Deskriptoren abgebildet. So können Abstände und Typen wichtiger Wechselwirkungen unter Verwendung eines *Pharmakophors* [Willett, 1995] kodiert werden. Gegebenenfalls lassen sich aussagekräftige *Modelle* aus der Überlagerung von Molekülen [Bures, 1997; Klebe, 1993] für die quantitative Beschreibung von Struktur-Aktivitätsbeziehungen (QSAR) bzw. Struktur-Eigenschaftsbeziehungen (QSPR) ableiten [Kubinyi, 1993; Kubinyi *et al.*, 1997]. Die Wahl eines passenden Deskriptors sowie die Ableitung aussagekräftiger Modelle ist eine eigenständige Disziplin und kann hier nicht in dem erforderlichen Umfang behandelt werden, weshalb auf einige Übersichtsartikel und weiterführende Bücher verwiesen werden soll [van de Waterbeemd, 1995; Sanz *et al.*, 1995; Brown & Martin, 1997; Flower, 1998; Livingstone, 2000; Raymond & Willet, 2002a; Glen & Adams, 2006].

Auf „klassische“ Verfahren, die die gegenseitige Bewertung existierender Moleküle zum Hauptgegenstand haben, wird an dieser Stelle nicht eingegangen, d.h. Methoden zur „klassischen“ Ähnlichkeitssuche [Turner *et al.*, 1995; Downs & Willet, 1996; Willett,

1996; Gillet *et al.*, 2003; Raymond & Willet, 2003; Wang *et al.*, 2005; Hert *et al.*, 2006] werden genauso wenig betrachtet wie entsprechende QSAR- bzw. QSPR-Ansätze [Lewis, 2005; Polanski *et al.*, 2006; Gedeck *et al.*, 2006] oder Pharmakophor-Suchmethoden [Clark *et al.*, 1994; Willett, 1995; Renner *et al.*, 2007; Sperandio *et al.*, 2007]. Stattdessen werden Programme vorgestellt, die auf Basis unterschiedlicher Randbedingungen neue Moleküle erzeugen.

- Einer der ersten Ansätze zur Identifikation alternativer struktureller Motive verwendete lediglich geometrische Informationen. Das Programm CAVEAT [Bartlett *et al.*, 1989; Lauri & Bartlett, 1994] sucht für einen Teil eines gegebenen Moleküls in einer Datenbank nach passenden substituierenden zentralen Strukturelementen (*Scaffolds*). Dafür werden vom Benutzer zwei Bindungen explizit spezifiziert (*exit vectors*), anhand derer eine Datenbank mit experimentell bestimmten Molekülgeometrien nach passenden Fragmenten durchsucht wird. SHOP [Bergmann *et al.*, 2007] verwendet ebenfalls ein Anfragemolekül, berücksichtigt aber zusätzlich noch die dreidimensionale Form sowie die Wechselwirkungsmuster für die Suchanfrage an eine Datenbank. Recore [Maaß *et al.*, 2007] benutzt neben mehreren Exit-Vektoren weitere geometrische und chemische Kriterien. Damit können interaktiv passende Ersetzungsmöglichkeiten aus einer Datenbank extrahiert werden, die durch Fragmentierung von Molekülen aus der CSD [Allen, 2002; Taylor *et al.*, 2002] unter Beibehaltung der jeweiligen Konformationen erzeugt worden ist.
- Anstatt nach strukturellen Ersetzungen für einzelne Moleküle zu suchen, kombinieren einige Programme strukturelle Motive aus verschiedenen Molekülen anhand geometrischer Kriterien direkt miteinander (*Hybridisierung*). Leadscope [Cross *et al.*, 2003] fragmentiert dafür eine gegebene Menge an Strukturen und setzt die erhaltenen Bausteine anschließend wieder zusammen (*Macrostructure assembly*), wobei unterschiedliche Kriterien als Zielfunktion verwendet werden können. BREED [Pierce *et al.*, 2004] benutzt keine Bewertungsfunktion, sondern führt eine vollständige Rekombination aller Bausteine miteinander durch. Das *Drug Evolution*-Verfahren von Lazar *et al.* basiert auf einer sehr ähnlichen Vorgehensweise [Lazar *et al.*, 2004]. Globus *et al.* verwenden einen genetischen Algorithmus zur Erzeugung von ähnlichen Verbindungen zu einem gegebenen Anfragemolekül [Globus *et al.*, 1999], wobei zur Bewertung der Ähnlichkeit ein einfacher graphbasierter Atompaar-Deskriptor [Carhart *et al.*, 1985] in Verbindung mit dem Tanimoto-Abstandsmaß [Tanimoto, 1957] zum Einsatz kommt.
- Eine Reihe anderer Verfahren setzen abgeleitete geometrische Bedingungen ein. NEWLEAD [Tschinke & Cohen, 1993] erfüllt ein gegebenes Pharmakophor, indem es bereits platzierte Fragmente mit Linkern (*spacers*) unterschiedlichen Typs verbindet. Dabei können einzelne Atome, Ketten und Ringe zur Verknüpfung verwendet werden. Die Methode von Eksterowicz *et al.* gebraucht aus den Strukturen einer oder mehrerer überlagelter Bindetaschen abgeleitete 4-Punkt Pharma-

kophore zur Selektion von Verbindungen aus einer kombinatorischen Bibliothek [Eksterowicz *et al.*, 2002]. Bywater *et al.* verwenden einen gitterbasierten Ansatz zur Berechnung molekularer Ketten, die durch positionierte Atome gekennzeichnete Pharmakophor-Bedingungen miteinander verbinden sollen [Bywater *et al.*, 2004].

- Weitere Verfahren setzen QSAR- bzw. QSPR-Modelle ein. LEA [Douguet *et al.*, 2000] verwendet Fragmente in Form von SMILES [Weininger, 1988] und optimiert diese mithilfe eines genetischen Algorithmus bezüglich eines QSAR-Modells. Für die Berechnung der Fitness wird eine Kombination verschiedener physikochemischer Eigenschaften angewandt, wofür die SMILES jeweils mit Corina [Sadowski & Gasteiger, 1993; Sadowski *et al.*, 1994] in 3D-Strukturen konvertiert werden. Nachbar benutzt ebenfalls einen genetischen Algorithmus und ein QSAR-Modell zur Manipulation und Optimierung der Topologie von Molekülen mithilfe einer Baumstruktur [Nachbar, 2000]. CoG [Brown *et al.*, 2004] repräsentiert Moleküle in Form graphbasierter Chromosome, wobei die Knoten vordefinierten Substrukturen oder Fragmenten aus einer Datenbank entsprechen. Eine Population von Chromosomen wird dann durch genetische Operatoren bezüglich unterschiedlicher Eigenschaften verbessert. Dieses Programm ist Bestandteil einer Methode zur Erzeugung von Molekülen durch mehrdimensionale Optimierung bezüglich eines inversen QSPR-Modells [Brown *et al.*, 2006].

Viele der oben beschriebenen Ansätze basieren auf relativ einfachen synthetischen Regeln zur Verknüpfung chemischer Motive. Bei den ersten Programmen werden lediglich geometrische Kriterien herangezogen. Viele der weiteren Ansätze verwenden sehr kleine Bausteine und damit nur lokale Informationen über die chemische Umgebung. Bei einem Großteil der Methoden erfolgt die Evaluierung der chemischen Eigenschaften erst nach Erzeugung der jeweiligen Moleküle. Um die Berücksichtigung von Aspekten der synthetischen Zugänglichkeit früher zu ermöglichen, beziehen eine ganze Reihe von Ansätzen zusätzliche Informationen direkt bei der Verknüpfung von Fragmenten mit ein. Dies kann in unterschiedlicher Weise geschehen, wie an den folgenden Beispielen verdeutlicht wird.

- Kombinatorische Bibliotheken bestehen aus zentralen Strukturelementen, die in der Regel über mehrere Anknüpfungsstellen verfügen, zu denen es jeweils eine Menge passender Substituenten gibt. Wie bereits zu Beginn des Kapitels erwähnt, führen die meisten Ansätze die Optimierung seitens der Produkte durch und nutzen die kombinatorische Struktur nicht aus [Gillet & Nicolotti, 2000]. Im Folgenden werden zwei alternative Ansätze vorgestellt. Die Methode von Andrews und Cramer basiert auf einem *Topomer-shape* Ähnlichkeitsmaß und identifiziert anhand dessen strukturell verwandte Moleküle zu gegebenen Anfragemolekülen in sehr großen *virtuellen Bibliotheken* [Andrews & Cramer, 2000; Cramer *et al.*, 2007]. Das Programm COLIBREE [Hartenfeller *et al.*, 2008] ist konzeptionell sehr ähnlich und verwendet *Particle Swarm Optimization* [Poli *et al.*, 2007] zusammen mit einem topologischen CATS Deskriptor [Schneider *et al.*, 1999].
- Eine andere Art der Modellierung kann mithilfe von *Fragmenträumen* erfolgen, die

aus Fragmenten mit klar definierten chemischen Umgebungen und einem Regelwerk bestehen, das die Kompatibilität der chemischen Motive spezifiziert. Dabei ist im Gegensatz zu einer kombinatorischen Bibliothek keine feste Topologie der Moleküle vorgegeben. Darauf aufbauend erzeugt TOPAS [Schneider *et al.*, 2000] mit einem evolutionären Algorithmus Moleküle, wobei das Optimierungskriterium die Ähnlichkeit zu einem Anfragemolekül ist. In diesem Zusammenhang wird die strukturelle Ähnlichkeit anhand eines 2D-Deskriptors oder eines topologischen Pharmakophors bewertet. FTree-FS [Rarey & Stahl, 2001; Fischer & Rarey, 2007] basiert auf dem *Feature Tree* Deskriptor [Rarey & Dixon, 1998] und führt mit diesem eine Ähnlichkeitssuche in Fragmenträumen durch. Dabei wird ein kombinatorischer Suchalgorithmus mit dynamischer Programmierung eingesetzt. Flux [Fechner & Schneider, 2006, 2007; Schüller *et al.*, 2008] verwendet einen stochastischen Algorithmus zur Vorhersage von Molekülen, die hinsichtlich ihrer Ähnlichkeit zu einem Anfragemolekül bewertet werden. Dabei werden jeweils zwei unterschiedliche Fingerprints und Ähnlichkeitsmaße angewandt.

- Analog zu den Ansätzen zur wissensbasierten Zerlegung von Molekülen (*Retrosynthese* [Corey & Wipke, 1969; Hanessian, 2005]), versuchen einige Programme chemische Reaktionen in einer umfassenderen Art und Weise zu modellieren. WODCA [Gasteiger *et al.*, 2000] berücksichtigt dafür neben Modellen zur Vorhersage chemischer Reaktivität auch physikochemische Eigenschaften und bildet dadurch organisch-chemische Reaktionen wesentlich umfangreicher ab [Höllering *et al.*, 2000]. SYNOPSIS [Vinkers *et al.*, 2003] verwendet für die virtuelle Synthese 70 unterschiedliche Reaktionstypen, die in SMILES-Format vorliegen. Die Methode von Priok *et al.* basiert ebenfalls auf einer Reaktionsdatenbank, die zusätzliche Regeln enthält, um die Aspekte unterschiedlicher Ausgangssubstanzen (Stereo- und Regioelektivität) zu berücksichtigen [Priok *et al.*, 2006]. Im Gegensatz dazu benutzt das Programm „Molecule Evoluator“ [Lameijer *et al.*, 2006b] einen genetischen Algorithmus mit atombasierter Mutation und Crossover und generiert Strukturen in einem interaktiven, benutzergesteuerten Prozess. Die Methode von van Deursen und Reymond verwendet einen algorithmisch sehr ähnlichen Ansatz, dort werden allerdings mögliche Zwischenstufen für eine „chemische Route“ von einem Ausgangs- zu einem Zielmolekül berechnet [van Deursen & Reymond, 2007].
- Neben den vorgestellten Suchverfahren gibt es auch Ansätze, die aus einer definierten Datenbasis alle theoretisch möglichen Moleküle erzeugen (*enumerieren*). Fink *et al.* haben eine Methode entwickelt, die alle Moleküle aus bis zu 11 Atomen unter Verwendung der Elemente Kohlenstoff, Stickstoff, Sauerstoff und Fluor generiert [Fink *et al.*, 2005; Fink & Reymond, 2007]. Dabei kommen etliche Substrukturfiltern zum Einsatz, um synthetisch nicht zugängliche oder instabile chemische Motive auszuschließen. Konzeptionell verwandt ist die Methode von Pollock *et al.*, die alle topologischen Grundgerüste molekularer Graphen (*Scaffold Topologies*) enumeriert [Pollock *et al.*, 2008; Wester *et al.*, 2008]. SmiLib [Schüller *et al.*, 2007] enumeriert kombinatorische Bibliotheken auf der Basis von SMILES. Pärn *et al.*

haben eine Methode veröffentlicht, die alle Moleküle eines kleinen Fragmentraums unter mehreren physikochemischen Randbedingungen erzeugt [Pärn *et al.*, 2007].

3.2 Ansätze mit Verwendung der Rezeptorstruktur

Bei Verwendung der Rezeptorstruktur für das de-novo-Design ergeben sich die Randbedingungen aus den geometrischen und physikochemischen Eigenschaften der Bindetasche [Kuntz, 1992; Kuntz *et al.*, 1994; Bohacek *et al.*, 1996; Kubinyi, 1998a; Klebe, 2000; Lyne, 2002; Waszkowycz, 2002]. Dabei werden meist in einem vorgeschalteten Schritt geeignete Bindestellen für kleine Moleküle identifiziert und darin potenzielle Wechselwirkungsmöglichkeiten ermittelt.

Es existieren vielfältige Ansätze zur Identifikation potenzieller Bindestellen für Liganden, die hier nicht im Detail besprochen werden. Ein Überblick ist bei Schneider und Fechner [Schneider & Fechner, 2005] sowie bei Harris *et al.* [Harris *et al.*, 2008] zu finden. Viele de-novo-Design-Verfahren verwenden eigene oder adaptierte Methoden. Die in diesem Rahmen am häufigsten angewandten Ansätze sollen allerdings kurz vorgestellt werden. GRID [Goodford, 1985] benutzt unterschiedliche Atome und Fragmente (*probes*) und bestimmt Wechselwirkungsenergien durch Platzierung und Bewertung dieser Proben mithilfe eines gitterbasierten Kraftfelds. MCSS [Miranker & Karplus, 1991] ist konzeptionell sehr ähnlich, bedient sich allerdings multipler Kopien der verschiedenen Proben, die zunächst zufällig platziert werden und anschließend eine Energieminimierung durchlaufen. Ein Vergleich der beiden Ansätze ist bei Bitetti-Putzer *et al.* zu finden [Bitetti-Putzer *et al.*, 2001].

Auf „klassische“ Docking-Verfahren [Bissantz *et al.*, 2000; Muegge & Rarey, 2001; Halperin *et al.*, 2002; Schneider & Böhm, 2002; Bursulaya *et al.*, 2003; Kitchen *et al.*, 2004; Cummings *et al.*, 2005; Leach *et al.*, 2006; Sousa *et al.*, 2006; Warren *et al.*, 2006; Rarey *et al.*, 2007], die die Platzierung existierender Moleküle in der Bindetasche eines Proteins zum Hauptgegenstand haben, wird an dieser Stelle ebenso wenig eingegangen, wie auf die korrespondierenden Bewertungsfunktionen zur Vorhersage von Bindungsaffinitäten [Tame, 1999; Stahl & Rarey, 2001; Gohlke & Klebe, 2002; Wang *et al.*, 2003; Tame, 2005; Jacobsson & Karlen, 2006; Seifert, 2006; Rarey *et al.*, 2007]. Stattdessen werden Programme und Strategien vorgestellt, die neue Moleküle unter Berücksichtigung der Bindetasche und weiterer Randbedingungen erzeugen. Eine Klassifizierung der Verfahren ist dabei nicht immer eindeutig möglich. Neben der angewandten Algorithmen unterscheiden sich die Ansätze hauptsächlich hinsichtlich der zugrunde liegenden molekularen Bausteine. Letztere werden in Kapitel 3.4 noch näher beschrieben.

3.2.1 Gitterbasierte Ansätze

Einer der ersten Algorithmen für strukturbasiertes de-novo-Design wurde Ende der 80er Jahre publiziert [Lewis & Dean, 1989]. Darin werden lediglich geometrische Eigenschaften der Bindetasche berücksichtigt. Lewis und Dean verwenden dabei eine begrenzte Menge von zweidimensionalen molekularen Gerüsten (*spacer skeletons*) und starre Transformationen, um diese in die geometrische Umgebung einer Rezeptortasche einzupassen

und dadurch spezielle Wechselwirkungspunkte zu verbinden. Kurz darauf wurde der Ansatz auf die dritte Raumdimension erweitert [Lewis, 1990]. Die in dem Programm BUILDER weiterentwickelten Konzepte dieses Ansatzes benutzen bereits in der Bindetasche platzierte Moleküle zur Erzeugung von regulären oder irregulären molekularen Gittern, auf denen in einer teilweise interaktiven Vorgehensweise atomare Ketten konstruiert werden [Lewis, 1992; Lewis *et al.*, 1992]. Die Menge an chemischen Informationen, die für diese ersten Ansätze berücksichtigt werden kann, ist sehr beschränkt. Das bedeutet, dass nur sp^3 -hybridisierte Kohlenstoff-Atome für die Erzeugung der atomaren Ketten verwendet werden und lediglich sterische Kriterien für die Bewertung herangezogen werden. Die entsprechenden chemischen Modelle wurden in einer nachfolgenden Version des Programms (BUILDER v2) erweitert, um eine größere Anzahl chemischer Elemente und damit verbundener molekularer Geometrien sowie energetische Effekte umfassender berücksichtigen zu können [Roe & Kuntz, 1995].

3.2.2 Verfahren zum Platzieren und Kombinieren

Im Gegensatz zu den oben beschriebenen gitterbasierten Techniken, die verschiedene Wechselwirkungspunkte innerhalb der Bindetasche mithilfe molekularer Gerüste verbinden, suchen Verfahren zum Platzieren und Kombinieren nach geeigneten Fragmenten (*Linker*), um bereits positionierte molekulare Bausteine zu einem (ganzen) Molekül zusammenzufügen.

Dabei unterscheiden sich die publizierten Ansätze hauptsächlich hinsichtlich der benutzten Linker sowie der Methode, die zur Platzierung der molekularen Bausteine verwendet wird. Das bekannteste und mit am weitesten verbreitete Programm in diesem Zusammenhang ist LUDI [Böhm, 1992a]. Es platziert molekulare Fragmente auf kompatible Wechselwirkungszentren in der Bindetasche und durchsucht eine Datenbank mit einigen Hundert Einträgen nach passenden verbrückenden Fragmenten [Böhm, 1992b]. LUDI kann Bausteine alternativ auch zunächst an ausgewählten Wechselwirkungszentren in der Bindetasche platzieren und anschließend inkrementell erweitern [Böhm, 1995]. Zur Abschätzung der Wechselwirkungsenergien wird eine empirische Bewertungsfunktion verwendet [Böhm, 1994]. In einer Erweiterung werden auch Aspekte der synthetischen Zugänglichkeit berücksichtigt [Böhm, 1996b].

Eine Reihe von weiteren Programmen sind LUDI konzeptionell sehr ähnlich, die Vorgehensweise für die initiale Platzierung der Bausteine unterscheidet sich jedoch geringfügig. So bedienen sich auf der einen Seite einige Ansätze der Ergebnisse von Rechnungen mit GRID [Goodford, 1985], mit dessen Hilfe Bausteine an energetisch günstigen Positionen in der Bindetasche platziert worden sind: SPLICE [Ho & Marshall, 1993] berücksichtigt dabei, neben den geometrischen Kriterien, zusätzlich benutzerdefinierte Pharmakophor-Bedingungen. PRO_LIGAND [Clark *et al.*, 1995; Waszkowycz *et al.*, 1994; Westhead *et al.*, 1995; Frenkel *et al.*, 1995; Clark & Murray, 1995; Murray *et al.*, 1995] integriert unterschiedliche Strategien zum Kombinieren von Fragmenten und verwendet diese in Verbindung mit einer empirischen Bewertungsfunktion.

Auf der anderen Seite basiert eine Reihe anderer Ansätze auf Resultaten von Rechnungen mit MCSS [Miranker & Karplus, 1991]. Darauf aufbauend kombiniert HOOK

[Eisen *et al.*, 1994] die Verwendung molekularer Gerüste aus einer Datenbank mit einem vereinfachten Van-der-Waals Potenzial zur Bewertung der nicht-polaren Wechselwirkungen. Leach und Kilvington haben in diesem Zusammenhang die Anwendung eines *Tweak*-Algorithmus [Shenkin *et al.*, 1987] zur Bewertung azyklischer Linker untersucht [Leach & Kilvington, 1994]. In einer Folgeversion ihres Programms haben sie den Strukturierungsprozess um die Berücksichtigung von potenziellen Ringschlüssen erweitert [Leach & Lewis, 1994].

Das Programm CONFIRM [Thompson *et al.*, 2008] verknüpft bereits platzierte Fragmente und durchsucht dafür eine Datenbank nach passenden Linker-Fragmenten (*bridges*) anhand geometrischer Kriterien, die aus experimentellen oder berechneten Bindungsmodi von Inhibitoren abgeleitet worden sind. Die erhaltenen Ergebnisse werden zu Molekülen zusammengesetzt und anschließend gedockt.

3.2.3 Algorithmen zum sequenziellen Aufbau

Der komplementäre Ansatz zum Platzieren und Kombinieren von bereits platzierten Fragmenten ist die inkrementelle Erweiterung von Fragmenten innerhalb der Bindeta-sche. Verfahren zum sequenziellen Aufbau von Molekülen ergänzen bereits platzierte Start-Fragmente durch Anbau von weiteren Fragmenten. Dabei unterscheiden sich die verschiedenen Ansätze hauptsächlich hinsichtlich der gebrauchten Bausteine sowie der Bewertungsfunktion, die zur energetischen Evaluierung der Moleküle und Teillösungen eingesetzt wird.

Eine Reihe von Programmen verwendet Atome als Bausteine. Der Prozess der Strukturierung ist dabei in der Regel in mehrere Stufen unterteilt. Nach Platzierung eines *Anker-Atoms* in der ersten Phase wird dieses in der zweiten Phase schrittweise erweitert und jeweils die Wechselwirkungsenergie abgeschätzt. LEGEND benutzt für letzteres einen Kraftfeldbasierten Ansatz [Nishibata & Itai, 1991] und sättigt nach erfolgreichem Aufbau des Moleküls offene Valenzen mit Wasserstoffatomen. Ist die berechnete Wechselwirkungsenergie ab einem bestimmten Punkt energetisch zu ungünstig, wird die entsprechende Teillösung verworfen und eine neue Struktur aufgebaut. Dabei sind viele der Entscheidungsprozesse randomisiert, so werden offene Valenzen und Hybridisierungszustände der zu verwendenden Atome zufällig ausgewählt [Nishibata & Itai, 1993]. Oftmals wird nur eine sehr geringe Anzahl an chemischen Elementen und Hybridisierungszuständen berücksichtigt. So verwendet das konzeptionell nahezu identische Programm GenStar [Rotstein & Murcko, 1993] nur sp^3 -hybridisierte Kohlenstoffe, RASSE [Luo *et al.*, 1996] hingegen die Elemente Kohlenstoff, Stickstoff und Sauerstoff in insgesamt acht unterschiedlichen Hybridisierungszuständen. GrowMol [Bohacek & McMartin, 1994; Bohacek *et al.*, 1999] benutzt zusätzlich noch Schwefel und kombiniert Atome und Fragmente als Bausteine. Es integriert jedoch, anstatt randomisierter Entscheidungsprozesse wie im Fall von LEGEND, ein Metropolis Monte Carlo Kriterium.

Neben Atomen verwenden einige Ansätze abstrahierte molekulare Ketten (*Skeletons*) oder Gerüste (*Templates*), die zunächst nur geometrisch charakterisiert sind und keine Informationen über chemische Elemente enthalten. Die Strukturierung wird damit in zwei Phasen unterteilt, die *primäre* und die *sekundäre* Strukturgenerierung. SPROUT

[Gillet *et al.*, 1990, 1993, 1994; Mata *et al.*, 1995] setzt während der ersten Phase Schablonen zu molekularen Gerüsten zusammen und wählt diese anhand mehrerer sowohl heuristischer als auch nicht-heuristischer Kriterien aus, um zunächst die sterischen Randbedingungen der Bindetasche zu erfüllen. In einem zweiten Schritt werden den erzeugten Gerüsten Atomtypen zugewiesen, um neben sterischer auch physikochemische Komplementarität zu erreichen. Barakat *et al.* haben die Zuweisung von Atomtypen auf molekulare Gerüste ausführlich untersucht [Barakat & Dean, 1995a,b,c,d,e]. SPROUT integriert in diesen Prozess auch die Berücksichtigung synthetischer Zugänglichkeit [Gillet *et al.*, 1995; Johnson *et al.*, 2004].

Eine dritte Gruppe von Programmen basiert auf der Verwendung molekularer Fragmente. GROW [Moon & Howe, 1991] benutzt Aminosäure-Bausteine zum schrittweisen Aufbau von Peptiden. Die erzeugten Zwischenlösungen werden dabei mithilfe eines Kraftfeldes bewertet und eine Untermenge anhand eines Greedy-Kriteriums für die nächste Iteration ausgewählt. GroupBuild [Rotstein & Murcko, 1993] folgt einer analogen Strategie, verwendet allerdings 14 chemisch sehr unterschiedliche Fragmente und wählt aus diesen zunächst Start-Fragmente aus, die an benutzerdefinierten Stellen in der Bindetasche platziert werden. Diese werden dann in allen Kombinationsmöglichkeiten erweitert und ebenfalls durch eine kraftfeldbasierte Bewertungsfunktion energetisch evaluiert.

Einige weitere Verfahren realisieren ein schrittweises Vorgehen unter Verwendung unterschiedlicher und klar definierter Klassen von Bausteinen. Diese Art der Modellierung ist sehr ähnlich zu der von kombinatorischen Bibliotheken, bei denen ein zentrales Strukturelement (*Core*) mit Seitenketten (*R-Gruppen*) dekoriert wird. PRO_SELECT [Murray *et al.*, 1997] sucht für ein bereits platziertes Start-Fragment in einer Datenbank nach möglichen Substituenten und evaluiert diese mithilfe einer empirischen Bewertungsfunktion [Eldridge *et al.*, 1997]. Dabei werden gleichzeitig Aspekte der Synthetisierbarkeit berücksichtigt. Anstatt einer Datenbank verwendet FlexX^C [Rarey & Lengauer, 2000] eine kombinatorische Bibliothek mit explizit definierter Struktur, die aus mehreren Hundert Bausteinen bestehen kann. Die daraus ableitbaren Moleküle werden anhand einer empirischen Bewertungsfunktion evaluiert. In der Erweiterung FlexX^C-Pharm werden zusätzlich auch Pharmakophor-Bedingungen berücksichtigt [Gastreich *et al.*, 2006]. Im Gegensatz dazu benutzt COREGEN [Aronov & Bemis, 2004] eine Menge von Ring-Motiven und Linker-Fragmente, die inkrementell und alternierend zu Molekülen erweitert werden. Das Programm SLF_Libmaker [Krier *et al.*, 2005] folgt einem ähnlichen Ansatz, verwendet anstatt des alternierenden Aufbaus jedoch Ringe, Linker und funktionelle Gruppen, die in dieser festgelegten Topologie miteinander verknüpft werden. KNOBLE [Gerlach *et al.*, 2007] enumeriert Moleküle aus einer für Serinproteasen optimierten kombinatorischen Bibliothek mit klar definierter Topologie unter besonderer Berücksichtigung der synthetischen Zugänglichkeit der Produkte.

3.2.4 Simulationsmethoden und randomisierte Verfahren

Der Strukturierungsprozess kann auch den Gesetzmäßigkeiten eines Kraftfeldes in Verbindung mit einer Simulationsmethode unterworfen werden. CONCEPTS [Pearlman & Murcko, 1993] füllt die Bindetasche mit *Partikeln*, die sich wie einzelne

Atome verhalten und nach physikochemischen Gesetzmäßigkeiten bewegen können. Mithilfe von Molekulardynamik-Simulationen werden die Positionen der Partikel hinsichtlich der Wechselwirkungsenergien optimiert. In einem stochastischen und reversiblen Verfahren werden Bindungen zwischen Partikeln ausgebildet und so iterativ Moleküle erzeugt, wobei die einzelnen Schritte jeweils anhand eines Monte Carlo Kriteriums bewertet werden. Die erzeugten Strukturen werden anschließend hinsichtlich ihrer Energie minimiert. Anstelle der Partikel verwendet CONCERTS [Pearlman & Murcko, 1996] molekulare Fragmente.

Einige weitere Ansätze sind konzeptionell sehr ähnlich und unterscheiden sich hauptsächlich hinsichtlich der angewandten Bewertungsfunktionen. Dazu gehören DycoBlock [Liu *et al.*, 1999; Zhu *et al.*, 2001a] und F-DycoBlock [Zhu *et al.*, 2001b], die eine Kombination von Kraftfeld und Abschätzung der lösungsmittelzugänglichen Oberfläche für die Bewertung nutzen. MCDNLG [Gehlhaar *et al.*, 1995] basiert auf einer Kombination von Metropolis Monte Carlo und Simulated Annealing Protokollen und verwendet diese zur analogen Minimierung weniger unterschiedlicher Atome zur Erzeugung realistischer Moleküle (*super molecules*). Dabei werden lediglich die Elemente Kohlenstoff, Stickstoff und Sauerstoff berücksichtigt.

DLD [Miranker & Karplus, 1995] verbindet mit MCCS platzierte funktionelle Gruppen durch die Identifikation und den atomweisen Aufbau von Kohlenstoff-Ketten (*Skeletons*) aus einer Datenbank, wobei die energetische Bewertung dabei mithilfe einer einfachen Potenzialfunktion erfolgt. SMOG [DeWitte & Shakhnovich, 1996; DeWitte *et al.*, 1997; Ishchenko & Shakhnovich, 2002] verwendet insgesamt 40 teilweise funktionalisierte chemische Motive und erzeugt daraus basierend auf einem Metropolis Monte Carlo Ansatz Moleküle durch inkrementellen Aufbau. Dabei werden einige sehr flexible Fragmente in unterschiedlichen Konformationen berücksichtigt. Die Vorhersage der freien Bindungsenthalpie erfolgt mithilfe einer wissensbasierten Bewertungsfunktion.

LEGO [Gubernator *et al.*, 1995] verwendet aufwändige Energieminimierungsverfahren, um die durch Verknüpfung von platzierten Fragmenten zustande gekommenen Moleküle zu optimieren. Dabei ist die Anzahl der mit einbezogenen Linker relativ gering, um die Berechnungen bei gleichzeitiger Berücksichtigung von Aspekten der synthetischen Zugänglichkeit noch in akzeptabler Zeit durchführen zu können. Pellegrini und Field haben eine Methode entwickelt, die Molekulardynamik-Simulationen verwendet, um Strukturelemente einer bereits in der Bindetasche platzierten Verbindung mit Fragmenten auszutauschen, die einer Datenbank entnommen werden [Pellegrini & Field, 2003].

Anstelle von definierten Fragmenten benutzt Skelgen [Todorov & Dean, 1997] abstrahierte Templates und generiert daraus mit einer Kombination aus Simulated Annealing und Simplex-Optimierung im Zuge der primären Strukturierung molekulare Gerüste. In einem zweiten Schritt, der sekundären Strukturierung, erfolgt ähnlich wie bei SPROUT die Zuweisung von Atomtypen, wobei Skelgen dabei das molekulare Gerüst in separate Einheiten von jeweils fünf Atomen einteilt und die Zuweisung für diese zunächst in unabhängiger Weise durchführt [Todorov & Dean, 1998]. Dadurch wird eine deutliche Reduktion der Komplexität des Problems sowie des Rechenaufwands erreicht. In einer Erweiterung wird zusätzlich konformelle Flexibilität der Aminosäure-Seitenketten in der Bindetasche berücksichtigt [Todorov *et al.*, 2006].

3.2.5 Genetische Algorithmen und evolutionäre Programmierung

Im Zusammenhang mit strukturbasiertem de-novo-Design werden genetische Algorithmen für unterschiedliche Aufgaben eingesetzt. Hauptsächlich dienen sie der Strukturierung und -optimierung, bei der die Molekülrepräsentation in der Regel in Form eines Chromosoms vorliegt und mittels genetischer Operationen wie Selektion, Mutation und Crossover verändert wird. Dabei spielt die Evaluation der Fitness eine entscheidende Rolle. Dafür kommen in den meisten Fällen typische Energiefunktion um Einsatz, mit deren Hilfe die unterschiedlichen Beiträge näherungsweise berechnet werden können.

Das Programm *Chemical Genesis* [Glen & Payne, 1995] optimiert zufällig erzeugte oder vom Benutzer vorgegebene Moleküle unter Berücksichtigung unterschiedlicher geometrischer Randbedingungen, wie z.B. einer Bindetasche und/oder eines Pharmakophors. LigBuilder [Wang *et al.*, 2000] bedient sich einer Fragment-Datenbank und verwendet eine empirische Bewertungsfunktion. Zusätzlich werden eine Reihe chemischer Regeln für Abschätzung der Bioverfügbarkeit der erzeugten Moleküle angewandt. PEP [Budin *et al.*, 2001] berechnet lineare Oligopeptide aus Aminosäurebausteinen und benutzt einen genetischen Algorithmus zur Optimierung der Konformation. Die Abschätzung der Energie erfolgt mit einer wissensbasierten Bewertungsfunktion. ENPDA [Belda *et al.*, 2005] erzeugt ebenfalls Aminosäureketten, die Evaluation der Fitness erfolgt allerdings anhand benutzerdefinierter Oberflächen-Ausschnitte der Bindetasche.

Eine Reihe weiterer Ansätze bedient sich der Bewertungsfunktionen aus dem molekularen Docking. ADAPT [Pegg *et al.*, 2001] verwendet eine graphbasierte Repräsentation von Molekülen, wobei die Knoten des Graphen einzelnen Bausteinen des Moleküls, und die Kanten den chemischen Bindungen zwischen diesen Bausteinen, entsprechen. Der genetische Algorithmus optimiert die Graph-Repräsentation durch Crossover und Mutation und basiert dabei auf benutzer-spezifizierten Fragmenten mit dedizierten Anbaustellen. Die Evaluation der Fitness erfolgt mit der DOCK [Ewing & Kuntz, 1997] Bewertungsfunktion. LEA3D [Douguet *et al.*, 2005] benutzt knapp 8000 Fragmente, die aus Molekülkatalogen und Naturstoffdatenbanken abgeleitet wurden, und erzeugt daraus Moleküle durch lineare Verknüpfung von bis zu fünf Fragmenten. Jede entstandene Teillösung wird mit Corina [Sadowski & Gasteiger, 1993; Sadowski *et al.*, 1994] hinsichtlich der Geometrie optimiert und mit der Bewertungsfunktion von FlexX [Rarey *et al.*, 1996a; Böhm, 1998] evaluiert. EAISFD [Liu *et al.*, 2007] verwendet kleine Moleküle oder Fragmente (*tagged fragments*) als Ausgangskonfiguration und dekoriert diese mit ausgewählten Fragmenten aus einer Datenbank biologisch aktiver Verbindungen. Die Evaluierung erfolgt mit der Bewertungsfunktion von Surflex-Dock [Jain, 2003; Pham & Jain, 2006].

GANDI [Dey & Caffisch, 2008] kombiniert einen genetischen Algorithmus mit *Tabu Search* und verwendet einige Tausend benutzerdefinierte Fragmente zur Verbindung bereits platzierter Start-Fragmente. Die Bewertung erfolgt durch ein Kraftfeld und berücksichtigt gleichzeitig die Ähnlichkeit zu bekannten Inhibitoren.

3.3 Vergleich der Verfahren für das de-novo-Design

Ein objektiver Vergleich der verschiedenen de-novo-Design-Verfahren ist praktisch nicht möglich, da es keine etablierten Validierungsstrategien für die Bewertung der Qualität der Ergebnisse eines de-novo-Design-Verfahrens gibt, wie das beispielsweise für Docking-Verfahren und Bewertungsfunktionen der Fall ist [Bursulaya *et al.*, 2003; Wang *et al.*, 2003; Warren *et al.*, 2006]. Auch dort ist jedoch ein direkter Vergleich nicht immer einfach möglich und aussagekräftig [Cole *et al.*, 2005]. Dennoch sollen die Charakteristika der vorgestellten Ansätze zusammengefasst sowie Stärken und Schwächen im Kontext analysiert werden.

Die meisten der oben vorgestellten Programme wurde hinsichtlich ihrer Vorhersagekraft in einer oder mehrerer Publikationen validiert und konnten erfolgreich für bestimmte Szenarien eingesetzt werden [Nishibata & Itai, 1993; DeWitte *et al.*, 1997; Bohacek *et al.*, 1999; Honma *et al.*, 2001; Mancera, 2002; Krier *et al.*, 2005; Ali *et al.*, 2005; Firth-Clark *et al.*, 2006a,b; Herschhorn *et al.*, 2007; Schüller *et al.*, 2008]. Der Umfang dieser Validierung unterscheidet sich jedoch deutlich und die verschiedenen Anwendungsszenarien sind praktisch nicht miteinander vergleichbar. Die meisten der vorgestellten Ansätze und Strategien sind somit unter den jeweiligen gegebenen Bedingungen prinzipiell in der Lage, Moleküle vorherzusagen, deren Eigenschaften experimentell verifiziert werden können.

Dabei unterscheiden sich die Verfahren hauptsächlich hinsichtlich der Randbedingungen, der molekularen Bausteine sowie des verwendeten Algorithmus [Schneider & Fechner, 2005]. Die „frühen“ Verfahren für de-novo-Design zeichnen sich generell dadurch aus, dass viele der Anforderungen an ein präzises Verfahren aus Gründen der Komplexität nicht mit der erforderlichen Genauigkeit berücksichtigt werden konnten. Aus diesem Grund wurden oftmals atomare oder abstrakte molekulare Bausteine verwendet, deren Anzahl ebenfalls sehr begrenzt war. Als Konsequenz haben viele der Programme hinsichtlich ihrer Vorhersagekraft nicht überzeugen können [Stahl *et al.*, 2002]. Die synthetische Zugänglichkeit der erzeugten Moleküle war und ist dabei nach wie vor eine der wesentlichen Herausforderungen eines de-novo-Design-Verfahrens [Baber & Feher, 2004; Gasteiger, 2007].

Die chemischen Modelle und Algorithmen wurden in der „zweiten Generation“ der de-novo-Design-Verfahren (vgl. [Stahl *et al.*, 2002]) in vielerlei Hinsicht verbessert. So hat sich die Verwendung von Fragmenten als molekulare Bausteine durchgesetzt [Merlot *et al.*, 2002; Cross *et al.*, 2003; Erlanson *et al.*, 2004; Fattori, 2004; Rees *et al.*, 2004; Hartshorn *et al.*, 2005; Kolb & Caffisch, 2006; Hajduk, 2006; Siegel & Vieth, 2007]. Gleichzeitig sind die Strategien für die Erzeugung von Molekülen aus Fragmenten komplexer und ausgereifter geworden [Lewell *et al.*, 1998; Teague *et al.*, 1999; Leach *et al.*, 2000; Schneider, 2002; MacLean & Martin, 2004; Weber, 2005; Mauser & Stahl, 2007]. In diesem Zusammenhang wurden auch physikochemische Eigenschaften von Inhibitoren und Fragmenten analysiert [Livingstone, 2000; Pickett *et al.*, 2000; Oprea, 2000; Oprea *et al.*, 2001; Vieth *et al.*, 2004; Leeson *et al.*, 2004; Vieth & Sutherland, 2006]. Durch eine umfassendere Modellierung der strukturellen und physikochemischen Aspekte sollen die Chancen zur Vorhersage synthetisch zugänglicher und pharmakolo-

gisch wirksamer Moleküle verbessert werden [Lajiness *et al.*, 2004; Allu & Oprea, 2005; Boda & Johnson, 2006; Boda *et al.*, 2007].

Darauf aufbauend verwenden die meisten der ligandbasierten Verfahren evolutionäre Strategien zur Erzeugung und Optimierung von Molekülen. Dadurch können die entsprechenden Programme im Vergleich zu früheren Ansätzen einige Tausend Fragmente zur virtuellen Synthese berücksichtigen [Schneider *et al.*, 2000; Rarey, 2001; Fechner & Schneider, 2006; Cramer *et al.*, 2007; Dey & Caffisch, 2008] und einen größeren Teil des Lösungsraums durchsuchen. Allerdings lassen sich auch mit komplexeren topologischen Deskriptoren oft nur qualitative Aussagen über die zu erwartende Bindungsaffinität eines Moleküls machen. Einige ligandbasierte Verfahren verwenden deshalb einen mehrstufigen Prozess für die Molekülerzeugung und bewerten diese mithilfe komplexerer QSAR- bzw. QSPR-Modelle [Douguet *et al.*, 2000; Nachbar, 2000; Brown *et al.*, 2006]. Dies beinhaltet jedoch in der Regel die Konvertierung der Moleküle in 3D-Strukturen, was die Berechnungen zusammen mit der aufwändigeren Evaluation der Fitness bezüglich des quantitativen Modells relativ zeitintensiv macht.

Im Fall der strukturbasierten Verfahren sind die Berechnungen aufgrund der komplexeren Randbedingungen der Bindetasche an sich schon wesentlich aufwändiger. Aus diesem Grund benutzen Simulationstechniken nur relativ wenige bzw. kleine strukturelle Bausteine [DeWitte & Shakhnovich, 1996; Pellegrini & Field, 2003]. Evolutionäre Ansätze basieren zum Großteil auf der Verwendung abstrahierter molekularer Gerüste (Templates) [Gillet *et al.*, 1995; Stahl *et al.*, 2002; Johnson *et al.*, 2004], die eine schnellere Berechnung erlauben, jedoch die bereits erwähnten Nachteile bei der anschließenden Zuordnung der Atomtypen haben. Einige Verfahren bedienen sich der Bewertungsfunktionen molekularer Docking-Programme oder beinhalten eine abschließende Docking-Rechnung [Pegg *et al.*, 2001; Douguet *et al.*, 2005; Liu *et al.*, 2007; Dey & Caffisch, 2008]. Aufgrund der gewählten Algorithmen können sie einige Tausend Fragmente für die Strukturzeugung berücksichtigen. Allerdings werden bei diesen Verfahren Aspekte der synthetischen Zugänglichkeit und weitere wichtige Randbedingungen, wie physikochemische Eigenschaften bei der Strukturzeugung, nur unzureichend berücksichtigt. Bei der Verwendung kombinatorischer Bibliotheken sind diese Bedingungen zwar gegeben, allerdings können damit aufgrund der zugrunde liegenden Struktur keine neuen molekularen Topologien erzeugt werden.

Aus diesem Grund ist ein Verfahren wünschenswert, das möglichst viele der wichtigen Kriterien in einem integrierten Ansatz vereint. Bisher gibt es jedoch kein strukturbasiertes de-novo-Design-Verfahren, das direkt mit den molekularen Strukturen der Fragmente auf der Basis komplexerer chemischer Modelle und Bewertungsfunktionen sowie zusätzlicher Berücksichtigung synthetischer wie auch physikochemischer Aspekte neue Moleküle erzeugt. Dies war der Ausgangspunkt für die Entwicklung von FlexNovo.

3.4 Ansätze zur Identifikation und Auswahl struktureller Bausteine

Zeitgleich mit der Entwicklung und Anwendung der ersten Verfahren für das de-novo-Design hat sich die Frage nach geeigneten strukturellen Bausteinen gestellt. Nach der anfänglichen Verwendung von Atomen und molekularer Gerüste (Templates) hat sich im Laufe des vergangenen Jahrzehnts die Anwendung von *Fragmen-ten* durchgesetzt [Merlot *et al.*, 2002; Erlanson *et al.*, 2004; Fattori, 2004; Rees *et al.*, 2004; Hartshorn *et al.*, 2005; Howard *et al.*, 2006; Hajduk, 2006; Siegel & Vieth, 2007; Hajduk & Greer, 2007]. Einer der Gründe dafür ist in der Hypothese zu suchen, dass privilegierte und/oder aus bekannten Inhibitoren abgeleitete Fragmente sowohl die synthetische Zugänglichkeit als auch die tatsächliche Aktivität der durch Rekombination dieser Fragmente erzeugten Moleküle erhöhen [Schneider *et al.*, 2000; Cross *et al.*, 2003; Pierce *et al.*, 2004; Lazar *et al.*, 2004]. Viele Ansätze in der Literatur beschäftigen sich deshalb mit der Identifikation und Ableitung von Fragmenten aus bereits verwendeten Wirkstoffen.

Bemis und Murcko haben bekannte Inhibitoren bezüglich prominenter molekularer Gerüste [Bemis & Murcko, 1996] und Seitenketten [Bemis & Murcko, 1999] analysiert. Lewell *et al.* haben ein automatisiertes Verfahren (RECAP) für die automatische Zerlegung (*Fragmentierung*) von Inhibitoren aus dem World Drug Index entwickelt [Lewell *et al.*, 1998], wobei der Fokus auf strukturellen Motiven lag, die relativ einfach im Rahmen kombinatorischer Synthese erzeugt werden können. Diese Art der Vorgehensweise ist mehrfach verwendet und erweitert worden [Schneider *et al.*, 2000; Rarey, 2001; Kolb & Caffisch, 2006; Mauser & Stahl, 2007; Hartenfeller *et al.*, 2008]. Sheridan hat häufig ersetzte chemische Gruppen identifiziert [Sheridan, 2002] und diese hinsichtlich ihrer biologischen Aktivität charakterisiert [Sheridan, 2003]. Weitere Ansätze befassen sich mit der Identifikation von Heterozyklen [Broughton & Watson, 2004], generischen Ringen und Linkern [Aronov & Bemis, 2004; Nilakantan *et al.*, 2006] sowie privilegierten Substrukturen [Lameijer *et al.*, 2006a; Schnur *et al.*, 2006; Siegel & Vieth, 2007].

Neben strukturellen Gesichtspunkten spielt auch die Verwendung physikochemischer Eigenschaften eine große Rolle für die Auswahl und Priorisierung von Fragmenten. Die zugrunde liegende Annahme dabei ist, dass biologisch bzw. pharmakologisch aktive Verbindungen nicht gleichmäßig innerhalb des „chemischen Universums“ verteilt sind, sondern typischerweise in bestimmten Regionen lokalisiert sind, die sich zum Teil mithilfe einiger weniger physikochemischer Eigenschaften charakterisiert lassen [Oprea, 2000; Hann *et al.*, 2001; Dobson, 2004; Lipinski & Hopkins, 2004]. Die erste Formulierung dieser Art ist die *Rule-of-Five* [Lipinski *et al.*, 1997] nach der die Wahrscheinlichkeit, dass eine Verbindung eine nur geringe orale Bioverfügbarkeit hat signifikant zunimmt, sobald zwei der folgenden Eigenschaften verletzt sind: Das Molekulargewicht liegt jenseits von 500, die Anzahl der Wasserstoffbrücken-Donoren (ermittelt als die Anzahl aller Sauerstoff- und Stickstoffatome *mit* adjazentem Wasserstoffatom) ist größer als fünf, die Anzahl der Wasserstoffbrücken-Akzeptoren (ermittelt als die Anzahl aller Sauerstoff-

und Stickstoffatome *ohne* adjazentes Wasserstoffatom) ist größer als zehn oder der Wert des berechneten logP überschreitet fünf Einheiten.

In Anlehnung an die *Rule-of-Five*, die in ihrer ursprünglichen Form nicht auf Fragmente übertragbar ist, haben Congreve et al. eine *Rule-of-Three* formuliert [Congreve et al., 2003], und diese erfolgreich für die Auswahl von Fragmenten (für experimentelle Untersuchungen mithilfe von Röntgenkristallographie) angewandt. Oprea et al. haben die Veränderung typischer physikochemischer Eigenschaften im Zusammenhang mit der Optimierung der Leitstrukturen zu den letztendlichen Wirkstoffen untersucht [Oprea et al., 2001]. Vieth et al. haben charakteristische Eigenschaften von Fragmenten oral verabreichter Medikamente ermittelt [Vieth et al., 2004] und diese auch hinsichtlich der Abhängigkeiten von den jeweiligen Target-Klassen analysiert [Vieth & Sutherland, 2006]. Basierend auf der *ligand efficiency* [Hopkins et al., 2004], die ein Verhältnis aus der Anzahl der Atome einer Verbindung und deren Bindungsaffinität bildet, haben Reynolds et al. eine Untersuchung hinsichtlich der Abhängigkeit vom Molekulargewicht durchgeführt [Reynolds et al., 2007], womit dieses Maß prinzipiell auch für die Auswahl von Fragmenten verwendet werden kann. Verdonk und Rees haben eine ähnliche Erweiterung vorgenommen, ihre Definition einer *group efficiency* bezieht sich aber auf kleine chemische Gruppen [Verdonk & Rees, 2008].

Bisher gibt es keine Publikationen, die sich mit der Ableitung und Verwendung pharmakologischer bzw. pharmakokinetischer Eigenschaften im Hinblick auf Fragmente beschäftigen. Die Vorhersage dieser Eigenschaften ist selbst für Moleküle schwierig und mit großen Unsicherheiten behaftet [Norinder & Bergstrom, 2006; Gola et al., 2006]. Allerdings sind Untersuchungen zur Identifikation von reaktiven [Baurin et al., 2004a,b] sowie potenziell toxischer und mutagener [Kazius et al., 2005] Substrukturen durchgeführt worden.

Kubinyi sowie Leeson et al. setzen sich kritisch mit der Verwendung unterschiedlicher Filterkriterien im Zusammenhang mit dem rationalen Wirkstoffentwurf auseinander [Kubinyi, 2003; Leeson et al., 2004].

4 Modellierung

Die Qualität der Ergebnisse eines de-novo-Design-Verfahrens hängt entscheidend von den verwendeten chemischen Modellen ab und davon, inwieweit es mithilfe der verwendeten Konzepte möglich ist, die realen Gegebenheiten möglichst exakt abzubilden. Die Komplexität der zugrunde liegenden physikochemischen und pharmakologischen Prozesse und die Charakteristika der bisherigen Ansätze aus der Literatur wurden bereits in den Kapiteln 2 und 3 dargelegt.

In diesem Kapitel werden zunächst die grundlegenden Konzepte und Modelle des molekularen Docking-Programms FlexX erläutert. Darauf aufbauend wird das Konzept des chemischen Fragmentraums und die Ableitung von Fragmenten sowie die Modellierung chemischer Reaktionen beschrieben. Schließlich werden die angepassten, erweiterten und zusätzlichen Modelle für das strukturbasierte de-novo-Design beschrieben.

4.1 Das molekulare Docking-Programm FlexX

Bei FlexX handelt es sich um ein Programm zum semi-flexiblen Protein-Ligand-Docking. Dabei werden unterschiedliche Konformationen für den Liganden berücksichtigt, wohingegen der Rezeptor starr gehalten wird. Die Details der Methode sind in den jeweiligen Veröffentlichungen zu finden [Rarey *et al.*, 1996a,b, 1997, 1999a,b]. Im Folgenden wird insbesondere auf die relevanten Aspekte für die Entwicklung eines de-novo-Design-Verfahrens eingegangen.

4.1.1 Repräsentation von organischen Molekülen

Die Darstellung von Molekülen erfolgt durch ungerichtete Graphen: Die Knoten repräsentieren dabei die Atome des Moleküls und die Kanten repräsentieren die kovalenten Bindungen zwischen den Atomen. Um ein Molekül innerhalb des Programms entsprechend behandeln zu können, bedarf es neben der Verwendung des Molekülgraphen vieler zusätzlicher Informationen. Im Wesentlichen handelt es sich dabei um strukturelle, geometrische und physikochemische Angaben an Atomen und Bindungen. Aus Effizienzgründen werden diese direkt am Molekülgraphen annotiert und nach einer vorgeschalteten Konsistenzprüfung im Zuge der *Initialisierung* berechnet.

- Annotierte Informationen an einem Atom:
 - chemisch: Element, Hybridisierung, formale Ladung, Protonierung, van-der-Waals-Radius, Wechselwirkungstyp, Stereoinformation
 - geometrisch: 3D-Koordinaten, Wechselwirkungsgeometrien

- strukturell: ausgehende Bindungen, Zugehörigkeit zu Ringsystemen und aromatischen Systemen, Symmetrie-Klasse, Komponenten-Nummer
- Annotierte Informationen an einer Bindung:
 - chemisch: Bindungstyp, mögliche Torsionswinkel und dazugehörige Energien
 - geometrisch: Bindungslänge, lokale Koordinaten, Referenzatome für Torsionswinkel
 - strukturell: zugehörige Atome, Zugehörigkeit zu einem Ringsystem

Die Initialisierung umfasst mehrere aufeinander aufbauende Teilschritte, auf die an dieser Stelle im Einzelnen nicht eingegangen wird. Eine vollständige Zusammenstellung der Initialisierungsschritte ist bei der Beschreibung des FlexNovo-Initialisierungsmodells in Tabelle 4.2 zu finden.

4.1.2 Die Behandlung von flexiblen Molekülen

In FlexX wird der Ligand in einzelne *Komponenten* zerlegt, die jeweils durch eine oder mehrere drehbare azyklische Bindungen begrenzt werden. Abbildung 4.1 zeigt eine solche Zerlegung am Beispiel des Inhibitors Gefitinib. Ringsysteme und deren endständige Substituenten bilden immer jeweils eine eigene Komponente. Durch die Auswahl einer *Basiskomponente* entsteht ein gewurzelter *Komponenten-Baum*, anhand dessen der Ligand inkrementell aufgebaut werden kann.

Demzufolge wird die Konformationsisomerie von Molekülen mithilfe lokaler Informationen modelliert. Dazu wird an jeder drehbaren azyklischen Bindungen ein Satz von bevorzugten Torsionswinkeln gespeichert, die ursprünglich mit dem Mimumba-Modell bestimmt worden sind und einer Datenbank mit etwa 900 Substrukturen entnommen werden [Klebe & Mietzner, 1994]. Jedem Eintrag werden maximal 12 Torsionswinkel zugeordnet, wobei die verwendeten Winkel in der Regel auf einem 30° Raster liegen. Für Ringsysteme wird eine Menge niederenergetischer Konformationen mithilfe des Programms Corina berechnet und an den entsprechenden Komponenten selbst annotiert [Sadowski & Gasteiger, 1993; Sadowski *et al.*, 1994]. Die zugewiesenen Torsionswinkel werden symmetriekorrigiert, wobei neben den topologischen Eigenschaften des Fragments auch die geometrischen Aspekte berücksichtigt werden.

4.1.3 Das Wechselwirkungsmodell

Den Atomen des Liganden und denen des Aktiven Zentrums werden Wechselwirkungstypen und -geometrien zugeordnet, die zur Optimierung der Position und Konformation des Liganden in der Bindetasche verwendet werden. Jedes wechselwirkende Atom hat demzufolge einen oder mehrere Wechselwirkungstypen mit jeweils einem Wechselwirkungszentrum. Letzteres ist definiert als das Zentrum einer Kugel, auf der die Wechselwirkungsfläche liegt. Die verwendeten Wechselwirkungstypen sind in Tabelle 4.1 aufgelistet.

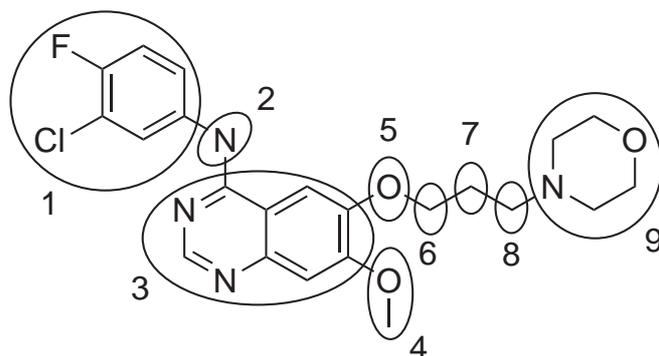


Abbildung 4.1: Zerlegung des Inhibitors Gefitinib in Komponenten, die durch nummerierte Ellipsen angedeutet sind.

Tabelle 4.1: Wechselwirkungstypen in FlexX

Name	Wechselwirkungsgruppe	wechselwirkende Gegengruppe
Gerichtete Wechselwirkungen (Kategorie 3 WW - stark)		
h_don	Wasserstoffbrücken-Akzeptor	h_acc
h_acc	Wasserstoffbrücken-Donor	h_don
metal	Metall-Atom	metal_acc
metal_acc	Metall-Akzeptor	metal
Hydrophobe „gerichtete“ Wechselwirkungen (Kategorie 2 WW)		
phenyl_center	Phenylring (Zentroid)	phenyl_center, ch3, phe, amide
phenyl_ring	Atom eines Phenylrings	phenyl_center
ch3_phe	Methyl-Gruppe	phenyl_center
amide	Amidbindung	phenyl_center
Hydrophobe „ungerichtete“ Wechselwirkungen (Kategorie 1 WW - schwach)		
ch	CH-Gruppe	ch, ch2, ch3, sulfur, aro
ch2	CH2-Gruppe	ch, ch2, ch3, sulfur, aro
ch3	Methyl-Gruppe	ch, ch2, ch3, sulfur, aro
sulfur	Schwefel-Atom	ch, ch2, ch3, sulfur, aro
aro	aromatisches Kohlenstoff-Atom	ch, ch2, ch3, sulfur, aro

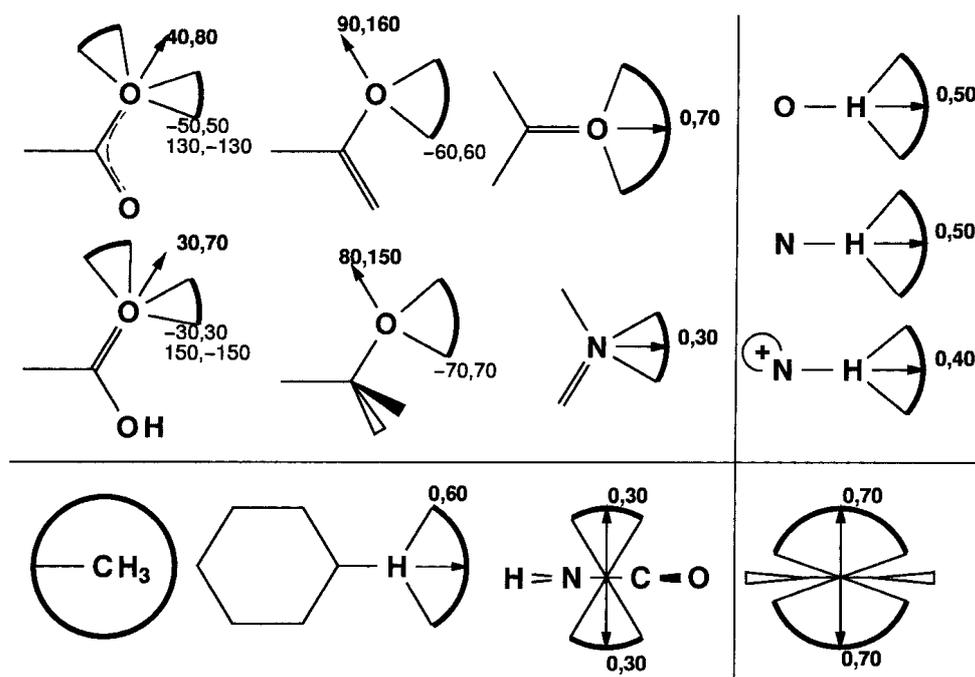


Abbildung 4.2: Wechselwirkungsgeometrien in FlexX: Wasserstoffbrücken-Akzeptoren (oben links), Wasserstoffbrücken-Donoren (oben rechts), Wasserstoffatome an aromatischen Ringen, Methylgruppen und Amidgruppen (unten links), Zentrum aromatischer Ringe (unten rechts). Wechselwirkungsgeometrien mit Metallatomen sind nicht aufgeführt. (Quelle: [Rarey *et al.*, 1996a])

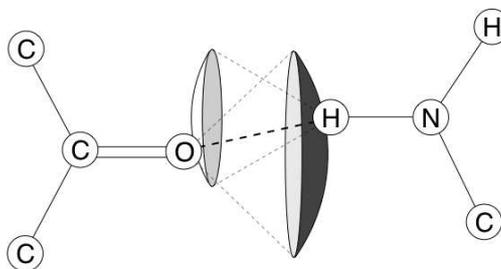


Abbildung 4.3: Geometrisches Modell einer Wasserstoffbrücken-Wechselwirkung zwischen einem Carbonyl-Sauerstoff und einem Amin-Stickstoff. Die Bedingung für die Ausbildung der Wechselwirkung ist, dass die Oberflächen der Wechselwirkungsflächen jeweils auf dem Wechselwirkungszentrum der Gegengruppe liegen. Die Abweichung von der Haupttrichtung der Wechselwirkung wird zugelassen, jedoch in der Bewertungsfunktion entsprechend bestraft. (Adaptiert aus: [Rarey *et al.*, 1996a])

Die dazugehörigen Wechselwirkungsgeometrien sind in Abbildung 4.2 dargestellt und nach Typ gruppiert. Dabei befinden sich Wasserstoffbrücken-Akzeptoren oben links, Wasserstoffbrücken-Donoren oben rechts, Wasserstoffatome an aromatischen Ringen, Methylgruppen und Amidgruppen unten links sowie das Zentrum aromatischer Ringe unten rechts. Die idealen Abstände der Wechselwirkungen betragen 1.9 Å für Wasserstoffbrücken, 2.0 Å für Metall-Wechselwirkungen und 4.5 Å für hydrophobe Kontakte. Die fett geschriebenen Winkelbereiche sind relativ zum gezeichneten Vektor in der Zeichenebene angegeben. Alle anderen Winkel beziehen sich auf die Ebene senkrecht zur Zeichenebene. Der aromatische Ring unten rechts steht senkrecht auf der Zeichenebene. Die angegebenen Wechselwirkungsgeometrien beziehen sich auf die Hybridisierung des jeweiligen Atoms und können somit für mehrere funktionelle Gruppen stehen.

FlexX bildet Wechselwirkungen aus, indem es kompatible Wechselwirkungsgruppen des Liganden und des Rezeptors so überlagert, dass das Wechselwirkungszentrum der einen Gruppe nahe der Wechselwirkungsfläche der Gegengruppe zum Liegen kommt. Das Ergebnis einer derartig ausgebildeten Wechselwirkung ist in Abbildung 4.3 schematisch dargestellt.

4.1.4 Die Bewertungsfunktion

Die Abschätzung der freien Bindungsenthalpie für die einzelnen Platzierungen des Liganden erfolgt mithilfe einer *Scoring-Funktion*, in der die unterschiedlichen energetischen Beiträge zur Gesamtenergie berücksichtigt werden. Der verwendete Ansatz ist abgeleitet von der im Jahr 1994 von Böhm veröffentlichten Funktion [Böhm, 1994]. Darin wird die freie Bindungsenthalpie ΔG näherungsweise als Summe von Beiträgen aus Wasserstoffbrücken ΔG_{wb} , ionischen Wechselwirkungen ΔG_{io} , Wechselwirkungen zwischen π -Systemen ΔG_{aro} , hydrophober Kontaktfläche zwischen Protein- und Ligand-Atomen ΔG_{lipo} sowie der Anzahl der frei drehbaren Bindungen im Ligand-Molekül ΔG_{rot} beschrieben.

$$\begin{aligned}
 \Delta G &= \Delta G_0 + \Delta G_{rot} \times N_{rot} \\
 &+ \Delta G_{wb} \sum_{\text{neutrale } H-Br.} f(\Delta R, \Delta \alpha) \\
 &+ \Delta G_{io} \sum_{\text{ion. } WW} f(\Delta R, \Delta \alpha) \\
 &+ \Delta G_{aro} \sum_{\pi-WW} f(\Delta R, \Delta \alpha) \\
 &+ \Delta G_{lipo} \sum_{\text{lipo. Kontakt}} f^*(\Delta R)
 \end{aligned} \tag{4.1}$$

Im Falle der neutralen, ionischen und π -Wechselwirkungen werden Abweichungen von der idealen Geometrie mithilfe von Termen des Typs $f(\Delta R, \Delta \alpha)$ bestraft, die sowohl vom Abstand R als auch vom Winkel α zwischen den jeweils beteiligten funktionellen Gruppen abhängen.

4.2 Chemische Fragmenträume

Das Konzept des chemischen Fragmentraums geht ursprünglich auf Arbeiten von Lewell et al. zurück. Das Ziel dieses Ansatzes war die Modellierung chemischer Motive, die sich besonders für kombinatorische Synthese-Protokolle eignen [Lewell et al., 1998]. Der daraus entstandene Fragmentraum ist bereits für Verfahren zum ligandbasierten de-novo-Design und zur Ähnlichkeitssuche verwendet worden [Schneider et al., 2000; Rarey, 2001].

Im Rahmen dieser Arbeit wurde ein erweitertes Konzept für Fragmenträume entwickelt, auf das detailliert im experimentellen Teil in Kapitel 6 eingegangen wird. Im Folgenden soll deshalb zunächst das prinzipielle Konzept des Fragmentraums erläutert, und die dabei wichtigsten Gesichtspunkte für die Entwicklung des neuen Ansatzes herausgestellt, werden. Dabei wird insbesondere auf die Strategien und Modelle zur Erzeugung von Fragmenten, zur Modellierung chemischer Reaktionen und zur Berücksichtigung synthetischer Aspekte eingegangen.

4.2.1 Retrosynthetische Zerlegung von Molekülen

Die meist verbreitete Methode zur Erzeugung von Fragmenten ist die retrosynthetische Zerlegung von Molekülen. Dabei werden chemische Motive identifiziert, die experimentell verhältnismäßig gut erzeugt werden können, d.h., unter Anwendung eines einzelnen oder zumindest weniger Reaktionsschritte sowie leicht zu realisierenden physikalischen Bedingungen, mit keinen oder nur geringen Mengen an entstehenden Nebenprodukten und mit einfacher Aufreinigung des gewünschten Reaktionsprodukts sowie möglichst einfach zugänglichen Ausgangssubstanzen in entsprechender Reinheit und Qualität.

Die Identifizierung retrosynthetisch relevanter Motive erfolgt über die Spezifikation molekularer Substrukturen, die so beschaffen sind, dass sie eindeutig die chemische Umgebung einer spezifischen Bindung beschreiben. Die retrosynthetische Zerlegung eines Moleküls, das *Shredding*, besteht konzeptionell aus zwei aufeinanderfolgenden Stufen. In einem ersten Schritt werden zunächst alle im Regelwerk spezifizierten Bindungen identifiziert und im Anschluss daran simultan gespalten. Die resultierenden Anknüpfungstellen werden entsprechend durch *Link-Atome* markiert. Bei letzteren handelt es sich um „Dummy-Atome“ mit entsprechender Typisierung, die von der jeweiligen chemischen Umgebung abhängt und im Shredding-Regelwerk spezifiziert ist.

In Abbildung 4.4 ist beispielhaft die Fragmentierung des Inhibitors Sorafenib dargestellt. Nach Spaltung aller relevanten Bindungen sind die entstandenen offenen Valenzen mit „*“ markiert. Die Typen der Link-Atome entsprechen denen des neu erstellten Modells, das im experimentellen Teil der Arbeit in Kapitel 6.1.2 vorgestellt wird.

4.2.2 Shredding-Strategie

Die gewählte Vorgehensweise zur Erzeugung von Fragmenten unterscheidet sich konzeptionell insbesondere in zwei Aspekten von der „klassischen“ Retrosynthese. Vor allem im experimentellen Kontext, aber auch bei mehreren computergestützten Ansätzen, spielt

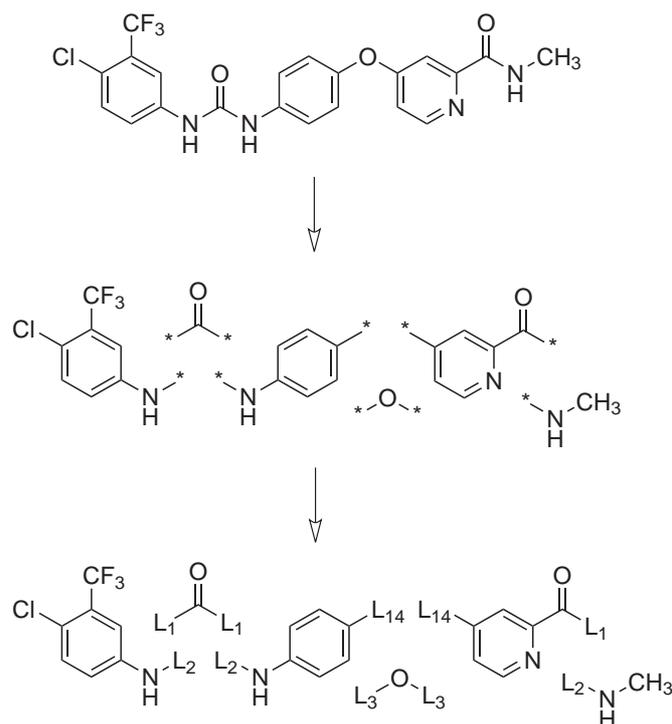


Abbildung 4.4: Zerlegung des Inhibitors Sorafenib in Fragmente. In einem ersten Schritt werden alle retrosynthetisch relevanten azyklischen Bindungen identifiziert und simultan gespalten. Anschließend erfolgt die Substitution der Anknüpfungsstellen mit Link-Atomen, deren Typ die jeweilige chemische Umgebung repräsentiert.

die Spaltung von Ringbindungen bzw. die Verwendung von Ringschlussreaktionen eine bedeutende und zentrale Rolle. Des Weiteren werden die chemischen Motive schrittweise gespalten und entsprechend *funktionalisiert* [Hanessian, 2005]. Im Kontext dieser Arbeit und in Bezug auf den neu erstellten Fragmentraum wird hingegen auf die Spaltung bzw. Knüpfung von Ringbindungen aus konzeptionellen und vor allem algorithmischen Gründen verzichtet. Der Fokus eines Fragmentraums liegt auf der Modellierung der aus der Verknüpfung von Fragmenten *resultierenden* chemischen Motive. Die Details der tatsächlich durchzuführenden Reaktionen, und damit auch die Funktionalisierung der Fragmente selbst, spielen dabei nur eine untergeordnete Rolle. Dieses Vorgehen stellt *per se* keine Einschränkung dar, sondern es wird vielmehr ein entsprechend angepasstes Modell verwendet, um die resultierenden strukturellen Motive adäquat abbilden zu können.

Dieser Zusammenhang soll in Abbildung 4.5 verdeutlicht werden. Diese zeigt den Cyclooxygenase-Inhibitor Celecoxib (durch ein Rechteck markiert) und zwei mögliche Vorgehensweisen bei dessen Zerlegung. Der rechte Teil der Abbildung entspricht der „klassischen“ retrosynthetischen Vorgehensweise. Sie zeigt aus Gründen der Übersicht-

lichkeit nur einen letztendlich resultierenden Ast des kompletten retrosynthetischen Baumes, der durch Evaluation aller möglichen Reaktionswege ausgewählt worden wäre [Penning *et al.*, 1997]. Die Zerlegung des Inhibitors erfolgt schrittweise und die jeweils gespaltenen Motive werden gemäß der entsprechend zugrunde liegenden chemischen Reaktion funktionalisiert. Der linke Ast entspricht der im Zusammenhang dieser Arbeit umgesetzten Vorgehensweise. Alle relevanten Bindungen werden simultan gespalten und die Anknüpfungspunkte in einem zweiten Schritt durch entsprechende Link-Atome ersetzt, die die jeweilige chemische Umgebung repräsentieren.

Da die beiden Ansätze grundsätzlich verschieden sind, können die resultierenden Bausteine bzw. Fragment-Prototypen nicht direkt miteinander verglichen werden. Beispielsweise repräsentiert das Pyrazol-Fragment auf der linken unteren Seite exakt das chemische Motiv des Inhibitors, aus dem es hervorgegangen ist. Im Gegensatz dazu resultiert das unten rechts abgebildete Imidazol-Derivat aus der Zerlegung und Funktionalisierung des Diketons, das zusammen mit dem Hydrazin-Derivat für die Ringschlussreaktion zum Pyrazol verwendet wird.

Demnach dienen die auf der linken Seite abgebildeten Fragmente lediglich der Modellierung der durch Fragmentverknüpfung zustande kommenden chemischen Motive. Sie werden deshalb aus bereits existierenden Verbindungen abgeleitet. Dem gegenüber stehen die molekularen Bausteine auf der rechten Seite, die zwar ebenfalls aus retrosynthetischer Zerlegung des Inhibitors hervorgehen, die jedoch an sich chemisch real existierende Verbindungen darstellen und aus denen in diesem Fall durch entsprechende chemische Reaktionen der Inhibitor Celecoxib gebildet werden kann.

Zunächst bleibt festzuhalten, dass die Struktur von Celecoxib in einem Fragmentraum abgebildet werden kann obwohl mit der gewählten Shredding-Strategie keine Ringbindungen gespalten bzw. umgekehrt keine Ringschlussreaktionen durchgeführt werden. Gemäß der „realen“ retrosynthetischen Zerlegung wäre zwar eine Spaltung bzw. Bildung des Ringmotivs notwendig, allerdings bedarf es für die Durchführung der tatsächlichen Reaktion entsprechend funktionalisierter Vorstufen, weshalb auch die „reale“ Zerlegung letztendlich zu strukturell verwandten Edukten führt.

Demnach ist zunächst für diesen Fall zu schlussfolgern, dass durch die entsprechend angepasste Art der Modellierung und aufgrund der Beschaffenheit des Syntheseweges die resultierenden chemischen Motive in einer entsprechenden Weise abgebildet werden können. Dennoch ist dies sicherlich nicht für alle durch Ringschlussreaktionen zustande kommenden chemischen Motive möglich. Beispielsweise können Reaktionen zur Steroid-Synthese so nicht modelliert werden. Diese Grundgerüste können deshalb gegebenenfalls direkt als Fragment-Prototypen verwendet werden.

Da die Durchführung vieler Ringschlussreaktionen entsprechende funktionalisierte Vorstufen beinhaltet, kann davon ausgegangen werden, dass ein Großteil der wichtigen, häufig angewandten und effektiv durchführbaren chemischen Umwandlungen [Kolb *et al.*, 2001; Kolb & Sharpless, 2003; Service, 2008] in der vorgestellten Art und Weise abgebildet werden können. Gleichzeitig zeigen die resultierenden Fragment-Prototypen zum Teil große strukturelle Ähnlichkeit mit den tatsächlich verwendeten Bausteinen, auch wenn sie unterschiedliche Konzepte repräsentieren.

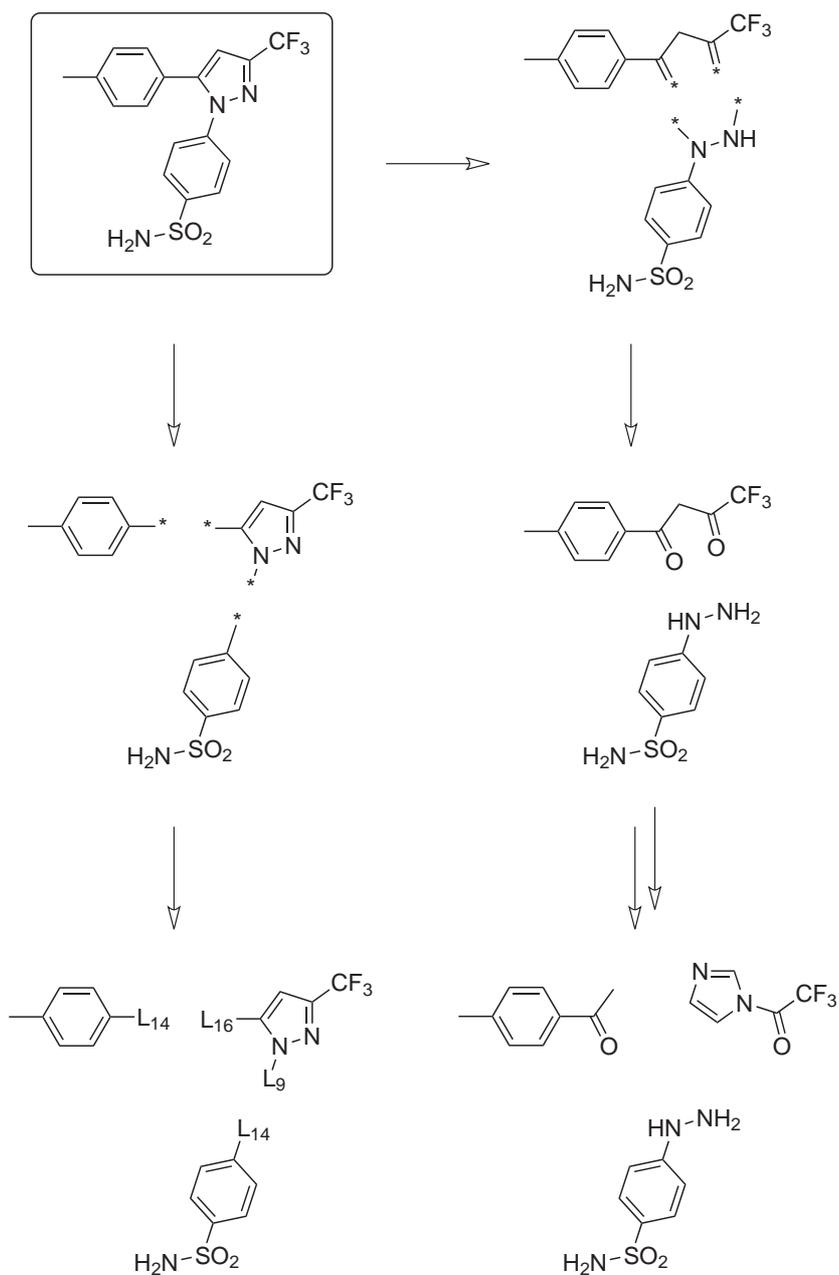


Abbildung 4.5: Retrosynthetische Zerlegung des Inhibitors Celecoxib (durch ein Rechteck markiert). Die Abbildung enthält zwei mögliche Vorgehensweisen. Die rechte Seite entspricht der „klassischen“ schrittweisen Zerlegung und zeigt einen resultierenden Ast des retrosynthetischen Baumes zusammen mit den jeweiligen funktionalisierten Reaktionszentren [Penning *et al.*, 1997]. Die linke Seite zeigt die simultane Spaltung aller relevanten azyklischen Bindungen und die anschließende Einführung von Link-Atomen.

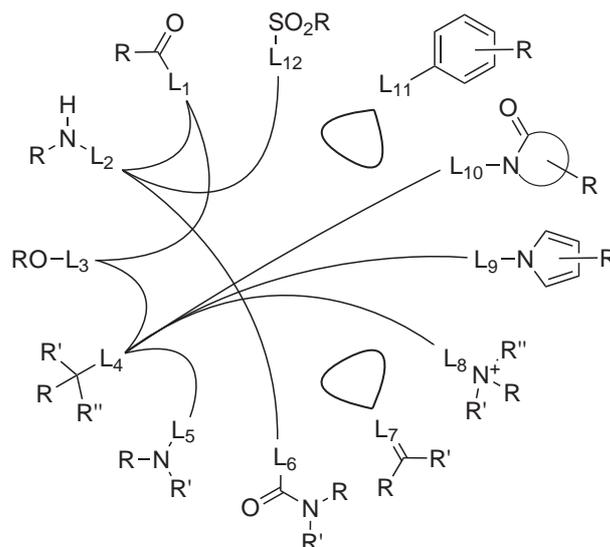


Abbildung 4.6: Die F'Tree-Fragment-Prototypen. Die Diversität der Fragmente wird durch unterschiedliche Substitutionen mithilfe der R-Gruppen modelliert. Diese können weitere Link-Atome enthalten bzw. nur aus einem Link-Atom bestehen. Bei den R'-bzw. den R''-Gruppen kann es sich um weitere (auch chemisch unterschiedliche) R-Gruppen handeln oder um ein Wasserstoff-Atom. Die Kreisausschnitte repräsentieren Ringe unterschiedlicher Größe, die auch anneliert bzw. verbrückt sein können. Die Fragment-Kompatibilität wird durch verbindende Linien zwischen den Link-Atomen angedeutet.

4.2.3 Von einzelnen Fragmenten zum Fragmentraum

Fragmente unterscheiden sich von Liganden strukturell lediglich dadurch, dass sie über ein oder mehrere explizite Link-Atome verfügen; sie können damit prinzipiell als kleine Liganden angesehen werden. Aus diesem Grund können die in Kapitel 4.1 beschriebenen Modelle in nahezu gleicher Weise auch auf Fragmente angewandt werden. Ein Fragmentraum besteht nun aus einer bestimmten Menge von Fragmenten und einem Regelwerk, in dem zum einen die Kompatibilität der Fragmente bzw. der Link-Typen untereinander spezifiziert ist. Zum anderen ist für jeden Link-Typ eine so genannte terminale Gruppe definiert, die das entsprechende Link-Atom ersetzen kann, sofern es nicht zur Verknüpfung verwendet wird.

Abbildung 4.6 enthält eine Darstellung des „Feature Tree“ Fragmentraums [Rarey, 2001]. Dabei sind die Fragmente im Gegensatz zur Abbildung 4.4 abstrahiert, generalisiert und nur noch die direkten Umgebungen der jeweiligen Link-Typen prototypisch dargestellt. Die Diversität der Fragmente resultiert deshalb aus den unterschiedlichen R-Gruppen. Die verbindenden Linien zwischen den Link-Atomen stellen die Kompatibilität der entsprechenden chemischen Motive dar.

An den grundlegenden Fragmenten eines Fragmentraums werden insbesondere keine strukturellen oder geometrischen Veränderungen durchgeführt. Es werden lediglich zusätzlich benötigte, abgeleitete Daten während der Initialisierung annotiert. Sie werden deshalb im Folgenden als „atomare“ Fragmente bezeichnet.

4.2.4 Modellierung chemischer Reaktionen

Die Modellierung unterschiedlicher chemischer Reaktionen erfolgt über die Definition der Kompatibilität der Link-Atome. Da die Erzeugung neuer Moleküle nicht nur durch topologische Verknüpfung, sondern im vorliegenden Fall insbesondere unter Verwendung der dreidimensionalen Geometrie der jeweiligen Fragmente erfolgt, können in der Kompatibilitätsspezifikation zusätzliche Eigenschaften der jeweiligen Bindung angegeben werden. Dabei handelt es sich um die Länge der neuen Bindung, die Hybridisierung (via Sybyl-Typisierung [Tripos, 2008]) der beteiligten Atome sowie ein bevorzugter Torsionswinkel für die initiale Geometrie des neuen Fragments (vgl. [Rarey, 2001]).

Abbildung 4.7 enthält eine schematische Darstellung der möglichen Operationen auf Fragmenten bzw. den beteiligten Link-Atomen. Die zentrale Amid wird durch Verknüpfung eines Carbonyl- und eines Amin-Fragments gebildet, wobei die dabei beteiligten Link-Atome verworfen werden. Dabei ändert sich nach Sybyl-Nomenklatur die Hybridisierung des Kohlenstoffs von *C.2* zu *C.am* und die des Stickstoffs von *N.pl3* bzw. *N.3* zu *N.am* sowie der Bindungstyp von einer ursprünglichen Einfach- hin zu einer Amidbindung. Gleichzeitig sollte die Amidbindung im Dreidimensionalen idealerweise einen initialen Torsionswinkel von 180 Grad haben. Sollten die jeweiligen Link-Atome nicht zur Fragmentverknüpfung verwendet werden, können sie durch die entsprechende terminale Gruppe ersetzt werden, wobei in diesem Fall die Carbonsäure bzw. das Amin entstehen und sich zum Teil ebenfalls die Sybyl-Typisierung ändert. Aus Gründen der Vollständigkeit ist die analog mit den terminierten Strukturen durchführbare Amidbildung unter Vernachlässigung der Reaktionsbedingungen mit angegeben.

Weitere Informationen fließen nicht in die Spezifikation ein. Darin unterscheidet sich der verwendete Ansatz wiederum von Synthesebetrachtungen im experimentellen Kontext, in dem viele zusätzliche Parameter wie z.B. die Polarisierbarkeit, das Dipolmoment, das verwendete Lösungsmittel und die Reaktionsbedingungen eine entscheidende Rolle spielen. All diese Kriterien werden soweit wie möglich in der gewählten Strategie zur Fragmenterzeugung berücksichtigt und spiegeln sich deshalb in den jeweiligen Strukturen wider. So werden z.B. lokale elektronische Effekte von chemischen Gruppen, die sich typischerweise nur auf deren unmittelbare Nachbarschaft auswirken und selten weiter als drei Bindungen reichen, sowohl direkt in der Shredding-Spezifikation als auch auf Fragment-Ebene berücksichtigt. Dies geschieht durch die gewählten Mindestgrößen der Substrukturen, die die zu schneidenden Bindungen spezifizieren.

4.2.5 Berücksichtigung der synthetischen Zugänglichkeit

Aspekte der synthetischen Zugänglichkeit werden bereits bei der Erzeugung der Fragmente berücksichtigt. Neben den Informationen über lokale elektronische Eigenschaften

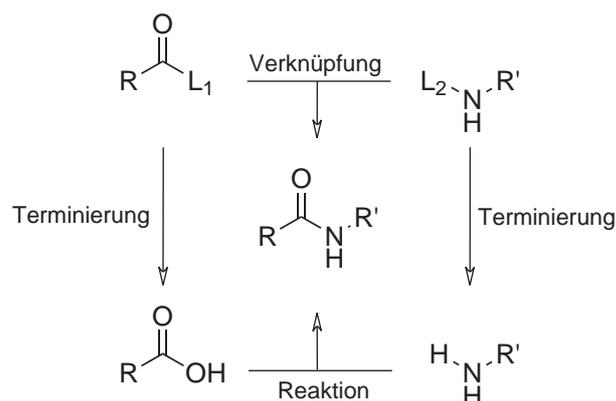


Abbildung 4.7: Operationen auf Fragmenten und Link-Atomen. Durch Verknüpfung des Carbonyl- und des Amin-Fragments wird das zentral dargestellte Amid gebildet. Die Terminierung der Link-Atome führt zur Bildung der Carbonsäure bzw. desamins, mit denen analog die „reale Reaktion“ durchgeführt werden kann. In beiden Fällen ändern sich die Sybyl-Typen der beteiligten Atome und Bindungen. Zusätzlich muss im dreidimensionalen Fall der Torsionswinkel angepasst werden, um eine niederenergetische initiale Geometrie zu erhalten.

können in dem Shredding-Regelwerk weitere Kriterien spezifiziert und verwendet werden. Mit deren Hilfe lassen sich beispielsweise chemisch instabile Fragmente direkt verwerfen. Zusätzlich können Regeln bezüglich der Größe der Fragmente, der Anzahl der gleichzeitig erlaubten Schnitte sowie deren Abstand zueinander formuliert werden. Weiterhin kann eine Liste von Ausschluss-Substrukturen verwendet werden, anhand der bestimmte toxische, reaktive oder ungewollte chemische Motive von vorne herein verworfen werden. Die vollständigen Spezifikationen der verwendeten Shredding-Regelwerke sind in Anhang C zu finden.

Unabhängig von der Formulierung des Shredding-Regelwerks können nach der Erzeugung der Fragmente weitere Filter-Regeln verwendet werden. Diese werden dann direkt bei der Fragmentverknüpfung berücksichtigt. Dadurch lassen sich sowohl wesentliche physikochemische Eigenschaften der erzeugten Strukturen steuern, wie auch bestimmte chemische Motive ausschließen, die erst durch den Zusammenbau von Fragmenten entstehen würden. Auf die Modellierung physikochemischer Eigenschaften wird in Kapitel 4.3.3 eingegangen, die algorithmischen Aspekte werden in Kapitel 5.3.4 behandelt und die tatsächlich verwendeten Filterkriterien an den jeweiligen Stellen im experimentellen Teil der Arbeit sowie in Anhang C zusammengefasst.

4.3 Der FlexNovo-Ansatz

Als Grundlage für die Entwicklung von FlexNovo, einer neuen computerbasierten Methode für den strukturbasierten de-novo-Wirkstoffentwurf mit chemischen Fragmenträumen,

wurden im Vorfeld eine Reihe von Entwicklungskriterien erarbeitet und spezifiziert, die im Folgenden zunächst kurz zusammengefasst werden. Anschließend wird auf spezifische Erweiterungen der Modelle und Konzepte eingegangen.

4.3.1 Entwicklungskriterien

Die Basis für die Entwicklung von FlexNovo stellt das molekulare Docking-Programm FlexX dar. In diesem Zusammenhang sollen insbesondere die chemischen Modelle zur Behandlung molekularer Wechselwirkungen und die Konzepte für die Berücksichtigung struktureller Flexibilität für Fragmente verwendet werden. Gleichzeitig ist der Einsatz wohl definierter Regeln und qualitativ hochwertiger Fragmente für die virtuelle Synthese von Molekülen ein zentraler Aspekt. Zusätzlich sollen weitere Randbedingungen wie physikochemische Eigenschaften sowie umfangreichere Bindungsgeometrie-Kriterien und Pharmakophor-Informationen direkt in die molekulare Aufbaustrategie integriert werden. Letztendlich soll daraus ein äußerst flexibles Programmpaket für das molekulare Design entstehen, das einfach und zu einem hohen Grad vom Benutzer steuerbar ist und umfangreiche problemspezifische Anpassungen ermöglicht.

Die Umsetzung dieser Kriterien erfordert sowohl Anpassungen und Erweiterungen der zugrunde liegenden chemischen Modelle als auch der darauf basierenden Algorithmen. Es stellt gleichzeitig die Grundlage dar, um eine große Anzahl von Fragmenten effizient behandeln zu können. Auf die algorithmischen Aspekte wird in Kapitel 5.3 im Detail eingegangen. Im Folgenden werden zunächst die wichtigsten Anpassungen der chemischen Modelle sowie die Erweiterungen vorgestellt.

4.3.2 Initialisierung von Fragmenten

Für die effiziente Behandlung einer großen Menge von Fragmenten und der daraus zusammengesetzten Zwischenlösung wird die ursprüngliche Initialisierung von FlexX durch die Einführung eines Phasenmodells modularisiert und flexibilisiert. Dabei bauen die einzelnen Phasen klar aufeinander auf, sodass die jeweils benötigten Informationen je nach Anforderung sukzessive berechnet werden können. Die einzelnen Schritte der Initialisierung und deren Zuordnung zu den jeweiligen Phasen sind in Tabelle 4.2 aufgeführt.

Aus Effizienzgründen wird insbesondere darauf geachtet, dass in den jeweiligen Initialisierungsphasen nur die Information berechnet wird, die für die entsprechende Operation bei der Behandlung und Erzeugung von Fragmenten erforderlich ist. Der benötigte Zeitaufwand für die Berechnung einzelner Informationen spielt dabei eine zentrale Rolle. Die Dauer einer kompletten Initialisierung eines Fragments beträgt durchschnittlich 0,1 Sekunden. Dabei entfallen auf die erste Phase der Initialisierung etwa zehn Prozent der Rechenzeit, auf die zweite Phase etwa 40 Prozent und auf die dritte und letzte Phase in etwa 50 Prozent. Auf den Einfluss der Initialisierung, auf die Gesamtlaufzeit sowie die Relation zu anderen Operationen wird in Kapitel 7.3 im Detail eingegangen.

Für die zwei- oder dreidimensionale Darstellung von Fragmenten sowie deren Speicherung ist der Abschluss der ersten Initialisierungsphase ausreichend. Dies ist insbesondere für das Betrachten großer Mengen von Fragmenten entscheidend, wobei einige Tausend

Tabelle 4.2: Phasenmodell der Initialisierung in FlexNovo

Phase	Aufgaben
	(Initiale Überprüfung) Konsistenz des Molekülgraphen und der 3D-Struktur
1. Phase	(Einfache Atom- und Bindungseigenschaften)
	Identifizierung von Ringen und Ringsystemen Überprüfung und Annotation der SYBYL-Atom- und Bindungstypen Überprüfung und Annotation der Hybridisierungszustände Identifizierung aromatischer Systeme Bestimmung der Formalladungen Delokalisierung aromatischer Systeme Identifikation rotierbarer Bindungen
2. Phase	(Grundlegende Molekül-Informationen)
	Bestimmung der Kraftfeld-Parameter (optional) Berechnung der unifizierten Atomradien Aufstellung der lokalen Koordinatensysteme Berechnung der molekularen Hydrophobizität Identifizierung der PMF-Atomtypen (optional) Berechnung der Stereodeskriptoren Identifizierung der Symmetrie-Klassen Zuordnung der Wechselwirkungstypen Zuweisung der Atominkremente für die log P Berechnung Generierung der diskreten Wechselwirkungspunkte (optional)
3. Phase	(Vollständige Molekül-Informationen)
	Berechnung von Ringkonformationen Annotierung der Torsionswinkel-Daten Aufstellen des Komponentenbaums Abschätzung der Wechselwirkungsenergien der einzelnen Komponenten
Der Abschluss der jeweiligen Initialisierungsphase wird mindestens benötigt für:	
1. Phase	Zeichnen, Speichern und Verknüpfen von Fragmenten
2. Phase	Berechnung des Deskriptors
3. Phase	Platzieren des Fragments, Transfer von Teillösungen

Fragmente in kurzer Zeitfolge initialisiert werden müssen. Zur Platzierung von Fragmenten und den inkrementellen Aufbau von Molekülen in der Bindetasche ist eine vollständige Initialisierung notwendig. Für alle anderen Operationen genügt der Abschluss der zweiten Initialisierungsphase, die im Folgenden als *Basis-Initialisierung* bezeichnet wird.

Bei der Initialisierung müssen speziell für Link-Atome einige Besonderheiten beachtet werden. Zum einen müssen Link-Atome aus algorithmischen Gründen eine eigene Komponente bilden. Zum anderen sollten sie aus Konsistenz-Gründen mindestens das Volumen eines Wasserstoff-Atoms zugewiesen bekommen, um eine realistischere Volumenabschätzung für Fragmente zu gewährleisten. Weiterhin müssen sie bei der Zuweisung der Torsionswinkel sowie späterer Cluster-Schritte gesondert berücksichtigt werden, damit ein möglichst feines Sampling des lokalen Konformationsraums an ausgehenden Bindungen ermöglicht werden kann (siehe auch Kapitel 7.1).

4.3.3 Physikochemische und geometrische Eigenschaften

Die Beschreibung von Molekülen bzw. Fragmenten wurde ausgehend von der ursprünglichen Repräsentation in FlexX dahingehend erweitert, dass zusätzlich eine Reihe von physikochemischen Eigenschaften verwendet werden können. Daraus resultiert ein entsprechender *Deskriptor*, der momentan die folgenden Informationen enthält:

- chemisch: Molekulargewicht, Anzahl der Schweratome und Bindungen zwischen Schweratomen, Anzahl der Ringe, Anzahl rotierbarer Bindungen, Anzahl der Wasserstoffbrücken-Akzeptoren und -Donoren, Anzahl der Stereozentren sowie berechneter molekularer logP-Wert und Brechungsindex
- geometrisch: polare/apolare Oberfläche, polare lösungsmittelzugängliche Oberfläche (PSAS)

Das Molekulargewicht wird in diesem Zusammenhang anhand einer Tabelle mit Atomgewichten berechnet, wobei Link-Atome keinen Beitrag leisten. Die Anzahl der Ringe wird aus den Bizusammenhangskomponenten abgeleitet, die während der Initialisierung berechnet werden. Die Bestimmung der Wasserstoffbrücken-Akzeptoren und -Donoren erfolgt gemäß der Definition, die von Lipinski et al. im Rahmen der Formulierung der *Rule-of-Five* verwendet worden ist [Lipinski *et al.*, 1997]. In diesem Zusammenhang wird lediglich die Anzahl der protonierten bzw. nicht-protonierten Heteroatome Stickstoff und Sauerstoff herangezogen. Für die Bestimmung der Anzahl rotierbarer Bindungen wird die von Böhm im Rahmen der Formulierung der Bewertungsfunktion verwendete Definition herangezogen [Böhm, 1998]. Dabei werden lediglich Bindungen zwischen nicht-entständigen, sp^3 -hybridisierten Schweratomen berücksichtigt.

Einige der oben beschriebenen und im folgenden verwendeten Eigenschaften werden bewusst nicht physikalisch-chemisch korrekt bestimmt. Das trifft insbesondere auf die Anzahl der Wasserstoffbrücken-Akzeptoren und -Donoren sowie die Anzahl rotierbarer Bindungen zu. Vielmehr steht die Verwendung einer einfachen, transparenten, in der Literatur anerkannten sowie weitverbreiteten Definition dieser Eigenschaften im Vordergrund.

Über die Spezifikation eines entsprechenden *Filters* können Bereiche für die oben genannten physikochemischen Eigenschaften des Deskriptors festgelegt werden. Zusätzlich kann der Filter auch Substrukturen enthalten, die in SMARTS-Syntax [Daylight, 2007] formuliert werden können und mit deren Hilfe die Anwesenheit bzw. Abwesenheit bestimmter chemischer Motive spezifiziert werden kann. Der Filter kann sowohl zur Selektion von Fragmenten verwendet werden als auch direkt während der Erzeugung neuer Moleküle Berücksichtigung finden.

Bei den physikochemischen Eigenschaften spielt die Tatsache eine entscheidende Rolle, dass aufgrund der gewählten Definitionen alle chemischen Eigenschaften bis auf die Anzahl der Stereozentren und der berechneten Werte für den logP als strikt additiv angesehen werden können. Aus diesem Grund können die Filterkriterien aller strikt-additiver Eigenschaften für ein beliebiges Molekül bereits vor dessen Erzeugung anhand der Werte der Deskriptoren der einzelnen „atomaren“ Fragmente überprüft werden.

4.3.4 Erzeugung neuer Fragmente und Moleküle

Die Erzeugung von Fragmenten und Molekülen erfolgt nach dem bereits in Kapitel 4.2.4 vorgestellten Schema zur Modellierung chemischer Reaktionen. Bei FlexNovo werden Fragmente inkrementell miteinander verknüpft, d.h., es werden immer nur zwei Fragmente gleichzeitig verbunden. Dabei erfolgt die Verknüpfung unter Verwendung der 3D-Koordinaten, wobei zunächst eine niederenergetische initiale Geometrie erzeugt und etwaige Anpassungen der Sybyl-Typisierung vorgenommen werden.

In diesem Zusammenhang treten einige Besonderheiten speziell im Hinblick auf die Behandlung der Stereoisomerie auf. Bei der Fragmentverknüpfung wird zunächst ein willkürliches Stereoisomer erzeugt, d.h., sofern eines der Fragmente ein Stereozentrum aufweist oder bei der Verknüpfung ein neues entstanden ist. Diese Information wird zunächst nicht weiter verwendet, da die initiale Konformation lediglich für die Initialisierung verwendet wird und die absolute Konfiguration des Stereozentrums zu diesem Zeitpunkt nicht von Belang ist.

Die Behandlung der unterschiedlichen Stereoisomere eines Fragments oder Moleküls wird analog zu FlexX auf Ebene der Platzierungsinformationen realisiert. Das bedeutet, dass die Stereozentren während des inkrementellen Aufbaus von Molekülen ihre lokale Konfiguration ändern können, sofern das entsprechende Atom azyklisch ist. Für zyklische Stereozentren muss dies im Zusammenhang mit der Berechnung der Ringkonformationen durch ein externes Programm geschehen. So können während des inkrementellen Aufbaus alle möglichen Stereoisomere effizient erzeugt und direkt evaluiert werden.

Bei dieser Vorgehensweise wird aus algorithmischen Gründen davon ausgegangen, dass sich Stereoisomere allein anhand der Geometrie und durch unterschiedliche Atom-Bezeichner am molekularen Graphen beschreiben lassen. Dabei wird bewusst vernachlässigt, dass es sich streng genommen um vollkommen verschiedene Fragmente bzw. Moleküle handelt. Im Gegensatz zu Enantiomeren unterscheiden sich insbesondere Diastereomere nicht nur in der Konfiguration der Stereozentren sondern auch in vielen wesentlichen physikalischen und chemischen Eigenschaften. Da die modellierten Eigenschaften der Fragmente und Moleküle allerdings keine entsprechenden Informationen enthalten

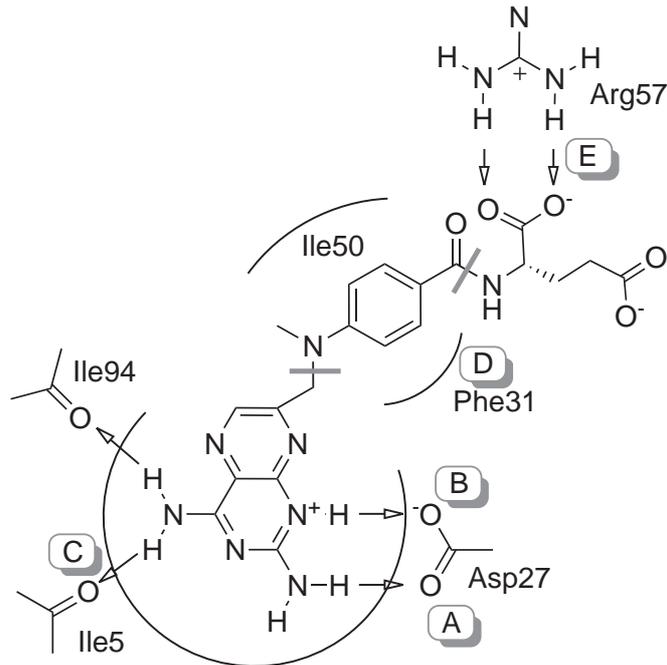


Abbildung 4.8: Pharmakophor für Dihydrofolat-Reduktase: Schematische Darstellung der Bindetasche der Dihydrofolat-Reduktase mit den wichtigsten Aminosäuren und dem Liganden Methotrexat. Hydrophobe Bereiche sind durch Kreissegmente, Wasserstoffbrücken-Wechselwirkungen durch Pfeile angedeutet. Die Einzelbedingungen des Pharmakophors sind durch die Buchstaben *A* bis *E* gekennzeichnet. Die Zerlegung von Methotrexat in Fragmente ist durch Schnitte an der Amin- bzw. Amidbindung angedeutet.

bzw. sie nicht in die Erstellung der zugrunde liegenden Modelle eingeflossen sind, kann an dieser Stelle auf die explizite Erzeugung aller möglichen Diastereomere verzichtet werden.

4.3.5 Pharmakophore

Neben den bisher beschriebenen geometrischen, energetischen und physikochemischen Randbedingungen kann zusätzlich auch ein rezeptorbasiertes *Pharmakophor* verwendet werden, wie es bereits in FlexX-Pharm realisiert ist. Die Definition eines solchen Pharmakophors kann unterschiedliche Einzelbedingungen enthalten, die aus spezifischen Wechselwirkungen sowie Einschluss- bzw. Ausschlussvolumina bestehen können. Für jede dieser Einzelbedingungen kann spezifiziert werden, ob es sich um eine *essenzielle* oder *optionale* Bedingung handelt [Hindle *et al.*, 2002]. Gemäß der Bezeichnung müssen essenzielle Einzelbedingungen in jedem Fall erfüllt werden, wohingegen das bei optionalen nicht unbedingt der Fall sein muss. Insbesondere kann für eine Menge optionaler

Bedingungen eine Unter- sowie eine Obergrenze spezifiziert werden, d.h. also beispielsweise, dass in einer Spezifikation mindestens zwei aber maximal vier von fünf optionalen Bedingungen gefordert werden kann.

Im Zusammenhang mit der Verwendung von Fragmenten und daraus zusammengesetzten Moleküle ergeben sich bei der Berücksichtigung von Pharmakophoren einige Besonderheiten. Das Beispiel-Pharmakophor in Abbildung 4.8 enthält fünf Einzelbedingungen *A* bis *E*. Dabei wird in diesem Zusammenhang zunächst ausgegangen, dass alle Einzelbedingungen essenziell sind und somit gleichzeitig erfüllt werden müssen. Anhand der Abbildung wird deutlich, dass zwar der Ligand Methotrexat das entsprechende Pharmakophor erfüllen kann, dies für die angedeuteten Fragmente des Liganden allein aber nicht möglich ist.

Demzufolge wird im Zusammenhang mit FlexNovo zwischen zwei Arten von Pharmakophor-Definitionen unterschieden. Der erste Typ entspricht dem oben beschriebenen Pharmakophor, $A\&B\&C\&D\&E$, und bezieht sich auf Moleküle und somit ausschließlich auf Lösungen des de-novo-Design-Verfahrens. Der zweite Typ enthält eine Untermenge dieser Einzelbedingungen, z.B. $A\&B\&C$, die von einem einzelnen Fragment erfüllt werden können. Letzteres wird als *Anker-Pharmakophor* bezeichnet.

Bei der Definition eines Pharmakophors muss sichergestellt werden, dass dieses inkrementell durch iterative Verknüpfung von Fragmenten erfüllt werden kann. Ausgehend von einem *Start-Fragment*, das zunächst nur das Anker-Pharmakophor erfüllt, werden die übrigen noch nicht erfüllten Einzelbedingungen sukzessive in folgenden Fragment-Anbauschritten abgedeckt.

Auf die Besonderheiten bei komplizierteren Pharmakophor-Definitionen und bei gleichzeitiger Verwendung essenzieller und optionaler Einzelbedingungen wird im algorithmischen Teil im Detail eingegangen.

5 Algorithmische Konzepte

Im folgenden Kapitel werden die algorithmischen Konzepte für das de-novo-Design-Problem vorgestellt. Dabei wird ausgehend von globalen Vorgehensweisen zur Realisierung struktureller Variation und den Strategien zur Behandlung molekularer Bausteine die verwendete Methode zur Platzierung molekularer Fragmente in der Bindetasche vorgestellt. Dem schließt sich die detaillierte Beschreibung des entwickelten Verfahrens zum strukturbasierten de-novo-Design an.

Aufgrund der Tatsache, dass der chemische Raum praktisch unendlich groß ist, die gesuchten Lösungen jedoch nur einen kleinen Teil dieses Raums ausmachen und die Beurteilung der Qualität der erzeugten Moleküle sehr subjektiv ist, wird eine effiziente Methode benötigt, die flexible und erweiterungsfähige Modelle und Konzepte zur virtuellen Synthese von Molekülen verwendet sowie multiple benutzerdefinierte Randbedingungen unterstützt.

5.1 Globale Vorgehensweisen beim de-novo-Design

Die zentrale Aufgabe eines de-novo-Design-Verfahrens ist die Erzeugung neuer Moleküle, die in der Regel aus einer Menge von definierten Bausteinen zusammengesetzt werden. Dabei hat die Natur dieser Bausteine selbst sowie die Strategie zu deren Auswahl und Kombination einen entscheidenden Einfluss auf die Qualität der Ergebnisse. Aufgrund der Komplexität des Optimierungsproblems wird deshalb eine entsprechende Methode benötigt, die bereits sowohl auf der Ebene der Bausteine als auch in der virtuellen Synthesestrategie möglichst viele der für die Ergebnisse wichtigen Eigenschaften berücksichtigt.

5.1.1 Strukturelle Variation durch Verwendung von Molekül-Bausteinen

Eine häufig angewandte Strategie zur Reduktion der Komplexität des Optimierungsproblems ist die Verwendung definierter molekularer Bausteine, die in der Bindetasche zu neuen Molekülen zusammengesetzt werden. Nimmt man an, dass diese Bausteine zunächst voneinander unabhängig sind und selbst chemisch bzw. strukturell nicht verändert werden, lässt sich dadurch das zu lösende Gesamtproblem auf die Optimierung der Kombinationsmöglichkeiten der Bausteine beschränken.

Die meisten de-novo-Design-Methoden verwenden *Fragmente* als molekulare Bausteine, wobei deren Größe je nach Methode sehr stark variieren kann. Die Spanne reicht dabei von wenigen Schweratomen bis hin zu Fragmenten mit einem Molekulargewicht von etwa 300 Dalton. Gleichzeitig kommen je nach gewählter Strategie unterschiedliche Arten von Fragmenten zum Einsatz, die sich anhand ihrer chemischen Charakteristika

in verschiedene Klassen einteilen lassen. So ist es in bestimmten Fällen sinnvoll, beispielsweise zwischen zyklischen und azyklischen Fragmenten oder zwischen Grundgerüsten, Linker-Fragmenten und funktionellen Gruppen zu unterscheiden.

Die Erzeugung von Molekülen durch die Kombination von Fragmenten kann prinzipiell auf zwei Arten erfolgen, die im Folgenden vorgestellt und verglichen werden.

- **Platzieren und Kombinieren:** Mehrere Fragmente werden unabhängig voneinander in der Bindetasche platziert und anschließend untereinander verbunden. Liegen zwei Fragmente räumlich ausreichend nah zusammen, können sie direkt durch Ausbildung einer neuen Bindung miteinander verknüpft werden. Ist dies nicht der Fall, kann nach kompatiblen *Linker-Fragmenten* gesucht werden, die eine entsprechende niederenergetische Konformation einnehmen können müssen, um beide Fragmente zu verbinden.
- **Inkrementeller Aufbau:** Ein oder mehrere *Start-Fragmente* werden in der Bindetasche platziert und inkrementell mit kompatiblen Fragmenten erweitert. Dabei ist auch der Austausch von Fragmenten in existierenden Teillösungen möglich. Die Start-Fragmente und Teillösungen werden so erweitert, dass geometrische und chemische Randbedingungen direkt erfüllt werden.

Verfahren zum Platzieren und Kombinieren haben den Vorteil, dass für die Fragmente unabhängig voneinander optimale Positionen in der Bindetasche bestimmt werden können. Dadurch kann lokal eine hohe Komplementarität zum Aktiven Zentrum realisiert werden. Bei der Verknüpfung von Fragmenten ist entscheidend, dass die resultierende Struktur anschließend in einer niederenergetischen Konformation vorliegt. Durch Verbinden nicht optimal zueinander angeordneter Fragmente entstehen Strukturen mit gespannten Konformationen, die eine Nachoptimierung der Platzierung erforderlich machen. Diese hat aber in der Regel zur Folge, dass die lokal optimale Anordnung der ursprünglichen Fragmente dabei verloren geht und dadurch die gesamte Teillösung unbrauchbar werden kann. Bei der Verwendung von Linker-Fragmenten hängt die Qualität der Ergebnisse entscheidend von der Menge an verfügbaren Linker-Fragmenten sowie von der Granularität der Abdeckung des jeweiligen Konformationsraums ab. Da hier zwei neue Bindungen gleichzeitig gebildet werden, ist die Notwendigkeit einer Nachoptimierung auch in diesen Fällen in der Regel gegeben.

Iterative Verfahren haben zunächst den Nachteil, dass explizite Start-Fragmente ausgewählt werden müssen. Dadurch ist die Menge an Ausgangskonfigurationen heuristisch und in der Regel sehr viel stärker eingeschränkt als bei Strategien zum Platzieren und Kombinieren. Allerdings wird die Platzierung der Start-Fragmente beim Anbau von weiteren Fragmenten in der Regel erhalten und das erweiterte Fragment bezüglich seiner Lage und Konformation angepasst. Dadurch ist eine schrittweise und permanente Optimierung der Platzierung und Struktur der Teillösung hinsichtlich der lokalen Umgebung der Bindetasche möglich. Aufgrund der beschränkten Anzahl an Start-Fragmenten ist die Menge potenzieller Lösungen zunächst stärker eingeschränkt als beim Platzieren und Kombinieren. Bei Anwendung eines effizienten Verfahrens zum inkrementellen Aufbau kann jedoch eine große Anzahl an Start-Fragmenten berücksichtigt werden. Gleichzeitig

müssen wesentlich weniger Teillösungen aufgrund von sich durch Fragmentverknüpfung verschlechternden Platzierungen verworfen werden.

Verfahren zum Platzieren und Kombinieren versprechen zunächst genereller einsetzbar zu sein und eine höhere Abdeckung des Lösungsraums zu ermöglichen. Allerdings sind die geometrischen Kriterien für die Erzeugung einer niederenergetischen Lösung sehr strikt. Insbesondere bei Bindungslängen und Bindungswinkeln aber auch bei den Überlappvolumina von Atomen bestehen sehr geringe Toleranzen, deren Einhaltung für die Erreichung eines nahezu optimalen Energiewerts zwingend notwendig ist. In geringerem Maße trifft dies auch auf ausgebildete Wechselwirkungen und die Kontaktfläche eines Moleküls zu. Aus diesem Grund gelten sehr strikte geometrische Randbedingungen bei der Verknüpfung zweier unabhängig voneinander platzierter Fragmente. Die daraus resultierende Menge an Platzierungen in niederenergetischen Konformationen ist dementsprechend gering. Allerdings ist aufgrund der großen Anzahl an Fragmenten die Wahrscheinlichkeit recht hoch, dennoch ausreichend viele Kombinationen von Fragmenten zu finden, die zu niederenergetischen Gesamtlösungen führen. Dabei besteht allerdings die Schwierigkeit aus der Vielzahl an Kombinationsmöglichkeiten von Fragmenten in multiplen Orientierungen und Konformationen diejenigen auszuwählen, die insgesamt sowohl niederenergetische Platzierungen als auch chemisch sinnvolle Moleküle mit hoher Komplementarität zur jeweiligen Bindetasche ergeben.

Aus diesem Grund liegt die Berücksichtigung weiterer wichtiger Randbedingungen nahe. Einige Aktive Zentren bestehen aus wohl-definierten Teilbereichen, *Subtaschen* oder anderen speziell ausgezeichneten Stellen wie z.B. *Hinges*, die geometrisch und chemisch sehr gut charakterisiert sind. Anhand von experimentellen Untersuchungen lässt sich ableiten, dass für solche Aktive Zentren in der Regel bestimmte Arten von Fragmenten sehr spezifische Positionen in solchen Teilbereichen einnehmen [Fattori, 2004; Rees *et al.*, 2004; Hajduk & Greer, 2007]. Diese Information kann nun ausgenutzt werden, um einerseits den Lösungsraum einzuschränken und andererseits, die Qualität der erzeugten Ergebnisse zu erhöhen.

Der inkrementelle Aufbau hat dabei neben der inhärenten Notwendigkeit der Festlegung expliziter Start-Fragmente einige wesentliche Vorteile. Aufgrund der gewählten Aufbaustrategie kann in einem deutlich höheren Maße gewährleistet werden, dass, basierend auf den initialen Positionen des Start-Fragments, im Laufe des Anbaus weiterer Fragmente eine ausreichend hohe Anzahl niederenergetischer Konformationen erzeugt werden kann. Dies ist insbesondere für die Erzeugung einer nahezu optimalen Geometrie der Platzierung des gesamten Moleküls nach mehreren Fragmentverknüpfungsschritten von Bedeutung. Durch die schrittweise Optimierung der Platzierungen der vorherigen Teillösungen kann die Einhaltung der geometrischen Randbedingungen bedeutend besser sichergestellt werden. Gleichzeitig kann dadurch der Lösungsraum signifikant eingeschränkt werden, was einen beachtlichen Vorteil für die Umsetzung effizienter Verfahren zur Lösung des Optimierungsproblems mit sich bringt. Dies ist insbesondere deshalb von Bedeutung, da die Platzierungsdaten aller Fragmente, und auch der daraus abgeleiteten Teillösungen, bei entsprechender Abdeckung des Konformationsraums nicht gleichzeitig im Hauptspeicher vorgehalten werden können. Auf die Aspekte der Laufzeit- und Speicherplatzeffizienz wird in den Kapiteln 5.3.6, 7.2.4.7 und 7.3 im Detail eingegangen.

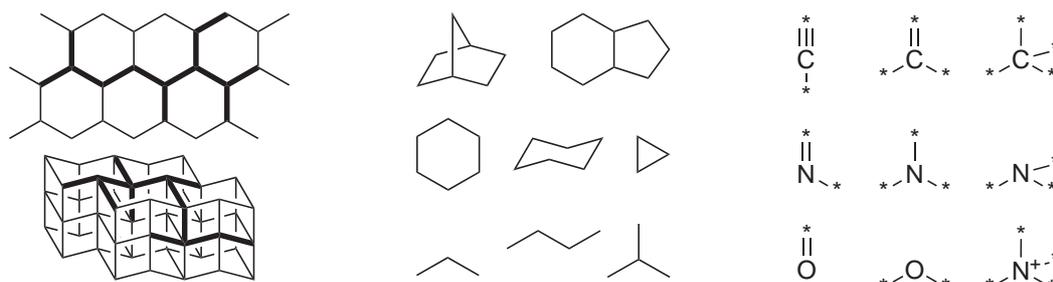


Abbildung 5.1: Beispiele alternativer Verfahren zur Strukturgenerierung. Der linke Teil zeigt ausschnittsweise die schematische Konstruktion molekularer Graphen auf der Basis zwei- bzw. dreidimensionaler Gitterrepräsentationen. In der Mitte sind Beispiele molekularer Gerüste bzw. Template-Strukturen dargestellt. Der rechte Teil zeigt exemplarisch drei Elemente in unterschiedlichen Hybridisierungszuständen.

5.1.2 Alternative Verfahren zur Realisierung struktureller Variabilität

Soll die chemische Variabilität der erzeugten Moleküle nicht von vorne herein durch die Verwendung strukturell klar definierter Fragmente eingeschränkt werden, müssen abstraktere bzw. atomare Bausteine verwendet werden. Die mit diesen Strategien vorhergesagten Moleküle sind allerdings in der Realität nur schwer herzustellen und erfüllen andere wichtige Eigenschaften typischerweise auch nur in geringem Maße [Stahl *et al.*, 2002]. Aus diesem Grund sollen darauf basierende Verfahren hier nur kurz beschrieben werden.

- **Graphenbasierte Verfahren:** In das Aktive Zentrum wird zunächst ein hexagonales oder kubisches Gitter gelegt. Mithilfe des Gitters werden dann in einem zweiten Schritt dreidimensionale molekulare Graphen erzeugt, die zunächst nur die sterischen Eigenschaften der Bindetasche erfüllen. Dabei entsprechen die Knotenpunkte des Gitters möglichen Atompositionen und die Kanten möglichen Bindungen. In einem weiteren Schritt werden den Knoten und Kanten der erzeugten Graphen Atom- und Bindungstypen zugewiesen, sodass chemisch sinnvolle Moleküle entstehen. Die Menge möglicher Graphen ist in der Regel sehr hoch und die Anzahl möglicher Kombinationen wächst dabei exponentiell mit der Größe der Graphen, sodass die Verfahren in der Regel sehr zeitaufwändig sind. Gleichzeitig werden die realisierbaren Hybridisierungszustände durch die verwendete Gitter-Topologie bestimmt [Lewis *et al.*, 1992; Bywater *et al.*, 2004].
- **Molekülgerüste:** Eine Menge definierter Molekül-Gerüststrukturen, so genannt *Templates*, werden aus bestehenden Molekülen abgeleitet oder aus Fragmenten abstrahiert. Diese werden zunächst in der Bindetasche platziert und anhand rein geometrischer Kriterien miteinander kombiniert, sodass die sterischen Randbedingungen der Bindetasche erfüllt werden. Die Hybridisierungszustände können dabei

in der Regel umfassender berücksichtigt werden. Aufgrund der hohen Komplexität der Zuordnung von Atom- und Bindungstypen erfolgt eine Unterteilung der erzeugten Graphen. Dabei werden für die Teilbereiche zunächst unabhängige Mengen gültiger Zuordnungen bestimmt, die dann in einem zweiten Schritt zu gültigen Molekülen kombiniert werden. Letztendlich verwendet solch ein Verfahren dann Konzepte, die den fragmentbasierten Methoden sehr ähnlich sind [Gillet *et al.*, 1993; Todorov & Dean, 1997].

- **Atombasierte Verfahren:** Anstelle molekularer Gerüste oder Template-Strukturen werden Atome in multiplen definierten Hybridisierungszuständen verwendet. Dadurch wird der oben beschriebene zweistufige Prozess zu einem Schritt zusammengefasst. Die unterschiedlichen Atomtypen werden direkt in der Bindetasche platziert und dort miteinander kombiniert. Dadurch wird eine gleichzeitige Evaluation der sterischen und chemischen Komplementarität zur Bindetasche ermöglicht und notwendig. Im Vergleich zur Verwendung von Molekül-Gerüsten sind jedoch wesentlich mehr Anbauschnitte notwendig, um ein fertiges Molekül von vergleichbarer Größe zu erhalten. Zusammen mit der komplexeren Bewertung der Zwischenlösungen hat dies einen wesentlich höheren Rechenaufwand. Aus diesem Grund erfolgt die Bewertung der Zwischenlösungen häufig unter Verwendung gittergestützter Verfahren [Miranker & Karplus, 1991; Pearlman & Murcko, 1993; Gehlhaar *et al.*, 1995].

Bei den oben beschriebenen Ansätzen kann oftmals nur eine geringe Anzahl an Elementen und dazugehörigen Hybridisierungszuständen berücksichtigt werden. Die verwendete Datenbasis hat dabei analog zu den fragmentbasierten Ansätzen einen entscheidenden Einfluss auf die Ergebnisse. In organischen Molekülen sind in der Regel jedoch ohnehin nur wenige Elemente in einer begrenzten Menge an Hybridisierungszuständen zu finden. Allerdings ist, unabhängig von der chemischen Vollständigkeit der Datengrundlage, die Menge der aus den Kombinationsmöglichkeiten prinzipiell resultierenden unterschiedlichen chemischen Motive immens groß. Dabei ergeben aber nur verhältnismäßig wenige aller möglicher Kombinationen ein chemisch sinnvolles Molekül, insbesondere wenn dieses als Ganzes betrachtet wird.

Aus diesem Grund benötigen alle oben genannten Verfahren ein umfangreiches Regelwerk, in dem Substrukturen mit nicht erlaubten Kombinationen von Elementen und Hybridisierungszuständen enthalten sind. Aufgrund der vielen Randbedingungen für chemische Strukturen umfasst ein solches Regelwerk typischerweise hunderte Substrukturen, die in ihrer Größe zum Teil Fragmenten oder Teilmolekülen entsprechen [Baurin *et al.*, 2004a; Kazius *et al.*, 2005]. Dadurch trägt die Überprüfung der Regeln signifikant zum Umfang der Berechnungen bei der Zuordnung von Atom- und Bindungstypen bei, da diese für jede Zwischenlösung überprüft werden müssen, sodass die Algorithmen bei angemessener Überdeckung des Suchraums in der Regel sehr zeitaufwändig sind.

5.1.3 Die verwendeten molekularen Bausteine

Die Verwendung molekularer Fragmente scheint eine geeignete Vorgehensweise zu sein, um einerseits den Lösungsraum sinnvoll einzuschränken, andererseits eine ausreichende strukturelle Variabilität zu realisieren und dabei gleichzeitig qualitativ möglichst hochwertige Moleküle zu erzeugen.

Die in dieser Arbeit entwickelte de-novo-Design-Methode basiert auf einem inkrementellen Aufbau-Verfahren. Die Vorteile dieser Strategie gegenüber alternativen Vorgehensweisen wurde bereits in den vorangegangenen Kapiteln erläutert.

Die Beschreibung des eigentlichen Algorithmus erfolgt in Kapitel 5.3. Da die Erzeugung und Bewertung von Platzierungen für Fragmente inhärenter Bestandteil des de-novo-Design-Verfahrens sind, erfolgt zunächst eine kurze Beschreibung der Strategien zur Fragment-Platzierung.

5.2 Platzierung von Fragmenten in der Bindetasche

Der erste Schritt für den Aufbau von Molekülen in der Bindetasche eines Rezeptors besteht aus der Identifikation geeigneter Start-Konfigurationen, d.h., der Bestimmung einer - bezüglich der verwendeten Bewertungsfunktion - nahezu optimalen Orientierung des Fragments unter Ausbildung energetisch günstiger Wechselwirkungen zwischen kompatiblen Fragment- und Rezeptor-Gruppen. Im folgenden Kapitel wird deshalb zunächst eine Strategie zur Platzierung der Fragmente eines Fragmentraums vorgestellt.

Im Gegensatz zur Verwendung atomarer oder templatebasierter Bausteine haben Fragmente mehrere Vorteile. Strukturell haben sie eine gewisse Mindestgröße und weisen damit normalerweise einige wechselwirkende Gruppen auf. Gleichzeitig haben sie aber typischerweise nur wenige azyklische Bindungen und beinhalten wenige kleine Ringsysteme, sodass die Menge an konformellen Freiheitsgraden relativ klein ist.

Wie bereits in Kapitel 4.2.3 beschrieben, können die verwendeten Fragmente strukturell prinzipiell als kleine Liganden angesehen werden. Aus diesem Grund können dieselben Verfahren angewandt werden, die auch zum Platzieren von Liganden verwendet werden. Damit kann das Fragment-Platzierungsproblem auch als Protein-Ligand Docking-Problem angesehen werden. Für letzteres existieren eine ganze Reihe von Lösungsansätzen, die im Folgenden nur kurz beschrieben werden sollen.

5.2.1 Flexibles Docking

Eine häufig angewandte Strategie zur Reduktion der Komplexität des Fragment-Platzierungsproblems ist die Zerlegung in *Komponenten*, die bereits in Kapitel 4.1.2 beschrieben wurde. Bei Annahme der Unabhängigkeit der Komponenten ergeben sich zwei Möglichkeiten, das Fragment daraus wieder zusammensetzen. Dabei handelt es sich um die bereits in oberen Abschnitt beschriebenen Verfahren zum inkrementellen Aufbau bzw. zum Platzieren und Kombinieren. Da die Konzepte identisch sind, die Argumentation aber nicht direkt übertragbar ist, wird an dieser Stelle insbesondere auf die Unterschiede und die sich daraus ergebenden Schlussfolgerungen eingegangen.

Im Gegensatz zu molekularen Bausteinen, die durch ein de-novo-Design-Verfahren zu Molekülen zusammengesetzt werden, hat die Anordnung der Komponenten eines einzelnen Fragments eine klar definierte Topologie. Dabei besteht zwar eine konzeptionelle Entsprechung der Komponenten und Bausteine, allerdings mit dem entscheidenden Unterschied: Die Komponenten können nur in genau einer Weise wieder zusammengesetzt werden, nämlich so, dass sich das ursprüngliche Fragment ergibt. Im Gegensatz zu typischen Liganden bestehen Fragmente aufgrund ihrer geringeren Größe in der Regel nur aus wenigen Komponenten.

Für das Verfahren zum Platzieren und Kombinieren besteht gleichermaßen die Gefahr, dass nicht für alle Komponenten Platzierungen in der Bindetasche ermittelt werden können, weshalb das Fragment unter Umständen nicht vollständig aufgebaut werden kann. Insbesondere ist die Platzierung sehr kleiner Komponenten schwierig, da sie wenige bis gar keine gerichteten Wechselwirkungen ausbilden können. Somit wird trotz der nur geringen Anzahl an Komponenten der Aufbau des gesamten Fragments in einer niedere-nergetischen Konformation erschwert.

Beim inkrementellen Aufbau von Fragmenten besteht analog die Schwierigkeit, geeignete *Start-Komponenten* festzulegen, die allerdings typischerweise aus einer sehr begrenzten Zahl an Komponenten ausgewählt werden müssen. Das bedeutet gleichzeitig, dass nur wenige Anbauschnitte bis zum vollständigen Aufbau des Fragments erforderlich sind. Dadurch kann eine hohe Abdeckung möglicher Start-Komponenten und des jeweiligen dazugehörigen Konformationsraums realisiert werden. Aufgrund der Platzierung der Start-Komponenten ist es bei diesem Verfahren wesentlich besser möglich, auch für kleine Komponenten eine niedere-nergetische Gesamt-Platzierung zu erzeugen.

Der entscheidende Vorteil des inkrementellen Aufbaus ist auch bei der Platzierung von Fragmenten die direkte Berücksichtigung der geometrischen Randbedingungen, die für diese Methode in gleichem Maße gelten. So kann eine geometrisch korrekte Platzierung der Basiskomponenten schrittweise erweitert werden, ohne die initiale Geometrie und die ausgebildeten Wechselwirkungen signifikant zu verändern. Gleichzeitig kann in jedem Anbauschnitt gewährleistet werden, dass geometrisch korrekte und chemisch komplementäre Platzierungen zustande kommen und erhalten bleiben.

5.2.2 Alternative Verfahren zum Docking flexibler Fragmente

Sollen die Komponenten eines Fragments nicht als unabhängig voneinander angesehen werden, muss das Fragment als Ganzes in der Bindetasche platziert werden. Dafür kommen eine Reihe möglicher Ansätze in Betracht, wobei jedoch die Algorithmen in der Regel nur wenig Möglichkeiten zur Umsetzung einer effizienten Strategie bieten und deshalb hier nur kurz behandelt werden.

- **Starres Docking:** Das Fragment wird als starre Struktur in der Bindetasche durch dreidimensionale Transformationen platziert. Um einen größeren Teil des Lösungsraums abzudecken, wird oftmals nicht nur ein Satz an Koordinaten sondern *Konformationsensembles* verwendet. Diese müssen allerdings in einem vorherigen Schritt erzeugt werden, wobei die Bestimmung einer repräsentativen Auswahl

an niederenergetischen Konformationen selbst für ein Fragment in der Regel sehr zeitaufwändig ist [Agrafiotis *et al.*, 2007].

- **Simulation:** Die Platzierung des Fragments wird nach Erzeugung einer initialen Startorientierung mithilfe eines Kraftfeldes optimiert. Dabei werden in diskreten Schritten die Kräfte ermittelt, die auf die Atome des Fragments wirken und die Orientierung sowie die Konformation entsprechend verändert. Die Komplexität solcher Verfahren steigt drastisch mit der Genauigkeit der verwendeten Modelle. Generell sind solche Verfahren sehr zeitaufwändig und erfordern in der Regel die Verwendung multipler Startorientierungen, um einen hinreichend großen Teil des Konformationsraums abzudecken [van Gunsteren *et al.*, 2006].
- **Evolutionäre Strategien:** Nach Erzeugung einer Startkonfiguration wird die Platzierung durch ein Chromosom repräsentiert, dessen Fitness durch entsprechende genetische Operationen schrittweise in nachfolgenden Generationen optimiert wird. Typischerweise ist auch hier für die adäquate Abdeckung des Konformationsraums die Verwendung mehrerer Start-Konfigurationen erforderlich. Da die Evaluation der Fitness einen entscheidenden Teil der Berechnungen ausmacht, und für jede Teillösung in jeder Generation berechnet werden muss, sind solche Verfahren ebenfalls sehr zeitaufwändig.
- **Distanzgeometrie:** Eine alternative Darstellung des Problems lässt sich durch die Formulierung einer Distanz-Matrix realisieren. Der Konformationsraum des Fragments wird dabei in einer geschlossenen, konformationsunabhängigen Form dargestellt, indem die Abstände zwischen den Atomen anstatt der Koordinaten der Atome selbst kodiert werden. Um die Aufzählung aller möglicher Konformationen zu vermeiden, werden die Abstände nicht direkt benachbarter Atome nicht genau ermittelt sondern durch die Verwendung unterer und oberer Schranken mithilfe der Distanzgeometrie angenähert. Aus solch einer Darstellung können Distanzmatrizen berechnet werden, die eine entsprechende Konformation des Fragments beschreiben und zur Platzierung verwendet werden können. Allerdings führen nur sehr wenige aller möglichen Distanzmatrizen zu einer dreidimensionalen Anordnung der Atome, die tatsächlich in einer niederenergetischen Platzierung des Fragments resultiert. Insbesondere kann die Bewertung einer Platzierung aufgrund der vielen und strikten geometrischen Randbedingungen erst nach der Erzeugung einer Konformation aus der entsprechenden Distanzmatrix erfolgen. Somit sind solche Verfahren typischerweise weniger gut zur effizienten Platzierung von Fragmenten geeignet [Crippen & Havel, 1988].

5.2.3 Der verwendete Platzierungsalgorithmus

Aufgrund der Beschaffenheit der verwendeten Fragmente scheint die Behandlung der Flexibilität durch Zerlegung in Komponenten und anschließendem inkrementellen Aufbau, eine geeignete Vorgehensweise zu sein. Die Vorteile dieser Strategie gegenüber anderen wurde in den vorangegangenen Abschnitten erläutert. Insbesondere im Hinblick auf

die Konsistenz der verwendeten chemischen Modelle erfolgt die Berechnung der Platzierungen von Fragmenten in der Bindetasche mithilfe der Algorithmen des molekularen Docking-Programms FlexX. Die Vorgehensweise wird im Folgenden kurz zusammengefasst.

Nach abgeschlossener Initialisierung erfolgt die Platzierung eines Fragments in der Bindetasche in mehreren Phasen. Die erste Phase besteht aus der Auswahl einer oder mehrerer geeigneter *Basiskomponenten*. Eine solche Basiskomponente kann sich auch aus mehreren zusammenhängenden Komponenten zusammensetzen. Strategien zur Auswahl einer geeigneten Basiskomponente sowie einer geeigneten Anbau-Reihenfolge der Komponenten ist in [Rarey *et al.*, 1997] zu finden. In der zweiten Phase werden die Basiskomponenten im Aktiven Zentrum des Rezeptors mithilfe eines Pose-Clustering Algorithmus und anschließendem Clustering ähnlicher Lösungen platziert [Rarey *et al.*, 1996b]. Im dritten und letzten Schritt wird das Fragment durch inkrementellen Aufbau der noch nicht platzierten Komponenten im Rezeptor zusammengesetzt. Dabei wird eine gültige Komponenten-Reihenfolge anhand des in Kapitel 4.1.2 vorgestellten Komponenten-Baums ermittelt. Der Konformationsraum wird mithilfe eines *k*-Greedy Algorithmus durchsucht. In jedem Anbauschnitt wird durch Hinzufügen einer weiteren Komponente des Fragments der dazugehörige lokale Konformationsraum durchsucht und die mit dem Rezeptor überlappende Orientierungen verworfen. Anschließend wird nach neuen Wechselwirkungen gesucht. Die sich daraus ergebenden Platzierungen der Teillösung werden mithilfe der Bewertungsfunktion optimiert und es wird ein Clustering geometrisch sehr ähnlichen Platzierungen durchgeführt. Nach Anbau der letzten Komponente wird eine finale Optimierung der Platzierungen sowie ein abschließendes Clustering durchgeführt [Rarey *et al.*, 1996a].

Für die letzte Phase des FlexX-Platzierungsalgorithmus, dem inkrementellen Anbau der Komponenten, sind bei der Behandlung von Fragmenten im Gegensatz zum Ligand-Docking einige Anpassungen notwendig. Da dieser Teil des Algorithmus auch im Zusammenhang mit dem inkrementellen Aufbau von Molekülen Verwendung findet, wird auf die Modifikationen im folgenden Kapitel näher eingegangen.

5.3 Inkrementeller Aufbau von Molekülen

Die mit dem oben beschriebenen Verfahren bestimmten Start-Platzierungen für Fragmente können als Ausgangskonfiguration für die Erzeugung von Molekülen in der Rezeptorbindetasche verwendet werden. Im Gegensatz zum Docking von Liganden oder kombinatorischen Bibliotheken ist durch die Fragment-Platzierung an sich noch keine signifikante Einschränkung des Lösungsraums gegeben, da zu diesem Zeitpunkt keinerlei Informationen bezüglich des endgültigen Moleküls vorhanden sind. Die Start-Platzierung schränkt zwar die Erweiterungsmöglichkeiten hinsichtlich ihrer Geometrie und Positionierung zum Teil ein, jedoch ergeben sich durch die kombinatorische Struktur des zugrunde liegenden Fragmentraums für jede Konfiguration dennoch eine unendliche Anzahl an möglichen Lösungen. Dieses Kapitel befasst sich mit der Entwicklung einer Strategie und eines Algorithmus zur Optimierung dieser Erweiterungsmöglichkeiten durch virtu-

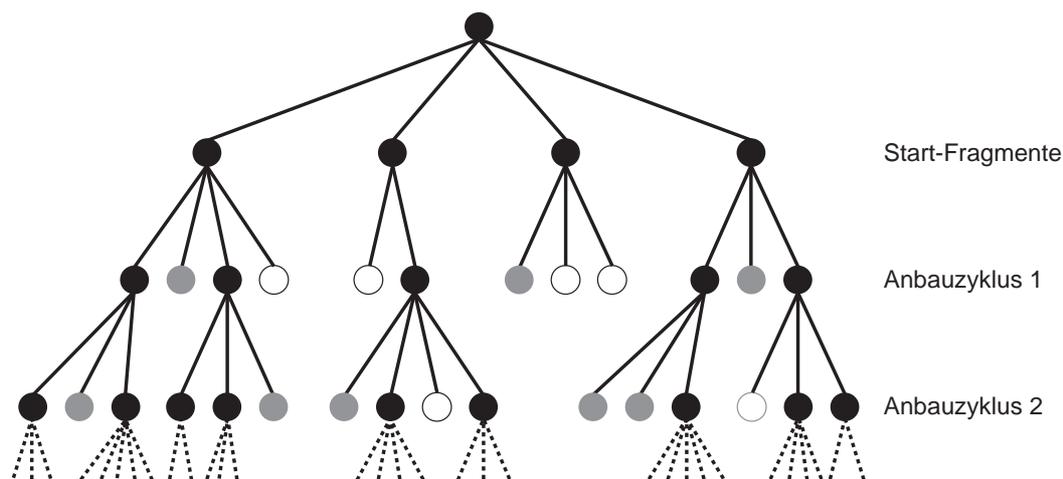


Abbildung 5.2: Suchbaum während des inkrementellen Aufbaus von Molekülen. Die schwarzen Knoten der ersten Ebene repräsentieren die Start-Fragmente, die der anderen Ebenen Teillösungen, die in folgenden Anbausritten expandiert werden. Graue Knoten repräsentieren Lösungen, die nicht mehr erweitert werden können und Teil der finalen Lösungsliste sind. Weiße Knoten stehen für Teillösungen, die nicht weiter betrachtet werden.

elle Synthese von Molekülen unter multiplen benutzerdefinierten Randbedingungen und Verwendung eines inkrementellen Aufbaualgorithmus [Degen & Rarey, 2006].

Durch die Spezifikation der Kompatibilität der Link-Atome spannen die verwendeten Fragmente einen Fragmentraum auf. Durch Auswahl von dezidierten Start-Fragmenten und Verwendung einer inkrementellen Aufbaustrategie entsteht eine gewurzelte Baumstruktur, bei der die Knoten den Fragmenten und die Kanten den mithilfe der Link-Atome modellierten azyklischen Verknüpfungen zwischen Fragmenten entsprechen.

Jedes Fragment, jede Teillösung und jedes Molekül wird analog der Vorgehensweise in FlexX in Komponenten c_0, \dots, c_n zerlegt. Durch Auswahl einer Basiskomponente entsteht der Komponenten-Baum mit ausgezeichneter Wurzel, der zum inkrementellen Aufbau der Struktur verwendet wird.

5.3.1 Die Greedy-Methode

Nach der Festlegung einer Menge an „atomaren“ Start-Fragmenten lässt sich der zu durchsuchende Lösungsraum in einer Baumstruktur abbilden, dem *Fragment-Baum*. Dabei entsprechen die Knoten der ersten Ebene des Baums alternativen Start-Fragmenten. Ein Knoten u auf Ebene i entspricht einer Teillösung die aus i „atomaren“ Fragmenten besteht. Durch Anbau weiterer „atomarer“ Fragmente erhält man neue (Teil-)Lösungen, die auf den Informationen von u beruhen und Kinder von u auf Ebene $i + 1$ sind. Sofern das Ergebnis einer derartigen Verknüpfung erweitert werden kann, wird es als Fragment bezeichnet, andernfalls als Molekül. Im Folgenden wird der Begriff „Fragment“ und „Teil-

lösung“ synonym verwendet. Sollte der Unterschied zu einem „atomaren“ Fragment des Fragmentraums von Bedeutung sein wird darauf explizit hingewiesen.

Für jedes Fragment bzw. Molekül wird der dazugehörige Konfigurationsraum ebenfalls durch eine Baumstruktur beschrieben, dem *Komponenten-Baum*. Diese Art der Modellierung geht auf FlexX zurück [Rarey *et al.*, 1996b]. An dieser Stelle soll das zugrunde liegende Konzept kurz erläutert werden, um die erforderlichen Erweiterungen entsprechend verdeutlichen zu können. Die dabei verwendeten Begriffe stimmen soweit wie möglich mit den ursprünglichen verwendeten Begriffen überein [Rarey *et al.*, 1996a]. Bei dem Komponenten-Baum handelt es sich um einen Teilbaum, dessen Wurzel ein Knoten des oben beschriebenen Fragment-Baums ist. Die erste Ebene des Komponenten-Baums entspricht dabei den verschiedenen *Basis-Platzierungen* des jeweiligen Fragments oder Moleküls, d.h. den Platzierungen der Basiskomponente. Ein Knoten v auf Ebene j entspricht einer Platzierung des Teilfragments oder Teilmoleküls bestehend aus den Komponenten c_0, \dots, c_{j-1} . Durch Anbau von Komponente c_j erhält man neue Platzierungen für das erweiterte Fragment oder Molekül c_0, \dots, c_j , die auf der Platzierung von c_0, \dots, c_{j-1} beruhen und Kinder von v auf Ebene $j + 1$ sind.

Jede Platzierung eines jeden Teilfragments oder Teilmoleküls wird mit der in Kapitel 4.1.4 vorgestellten Bewertungsfunktion evaluiert.

Der oben beschriebene Fragment-Baum ist zunächst nicht in seiner Höhe beschränkt. Allerdings kann dieser und damit das zu lösende Optimierungsproblem durch Berücksichtigung spezifischer und zentraler Aspekte des de-novo-Design-Problems eingeschränkt werden.

- Die optimale Lösung eines de-novo-Design-Algorithmus kann nicht definiert werden. Eine solche Lösung müsste eine optimale Komplementarität zum Rezeptor aufweisen sowie einen optimalen Bindungsmodus haben. Zudem müsste sie viele weitere Randbedingungen optimal erfüllen, wie z.B. synthetische Zugänglichkeit sowie pharmakodynamische und pharmakokinetische Eigenschaften. Für die jeweiligen einzelnen Punkte ist bereits die unabhängige Definition einer optimalen Lösung subjektiv und nicht generell möglich. Die Effekte beim Zusammenwirken der einzelnen Aspekte sind stark voneinander abhängig und nicht-linear. Insgesamt wird dadurch eine exakte Definition und Bestimmung der optimalen Lösung eines de-novo-Design-Verfahrens prinzipiell und praktisch unmöglich.
- In den allermeisten Fällen werden im Zuge der Leitstruktur-Identifikation kleine bis mittelgroße Moleküle zur Beeinflussung der Funktion eines Proteins gesucht. Solche Moleküle bestehen in der Regel aus nicht mehr als 20 bis 30 Schweratomen. Die verwendeten Fragmente können wie in Kapitel 4.2.3 beschrieben prinzipiell als kleine Moleküle angesehen werden. Ein solches Fragment hat typischerweise zwischen 5 und 10 Schweratome. Eine chemisch sinnvolle Lösung eines de-novo-Design-Verfahrens besteht somit in der Regel aus wenigen „atomaren“ Fragmenten.

Aufgrund der oben beschriebenen zentralen Eigenschaften des de-novo-Design-Problems ist die Anwendung eines exakten Verfahrens zur Lösung des Optimierungsproblems nicht möglich. Des Weiteren wird der Lösungsraum durch die maximale Größe

der gesuchten Moleküle beschränkt. Diese Beschränkung geschieht durch den Benutzer, der zunächst die maximale Höhe des Fragment-Baums durch die Spezifikation eines Parameters festlegt, der im Folgenden als die Anzahl der durchgeführten *Anbauzyklen* bezeichnet wird (siehe dazu auch Anh. A.2.2). Diese Vorgehensweise hat den inhärenten Vorteil, dass das resultierende de-novo-Design-Verfahren auch schrittweise durchgeführt werden kann. Darauf wird in Kapitel 8 näher eingegangen.

Die vollständige Durchsuchung des eingeschränkten Lösungsraums mithilfe von Backtracking-Verfahren ist sowohl aufgrund der hohen zu erwartenden Laufzeit als auch des wahrscheinlich immensen Speicherbedarfs nicht durchführbar. Die Bestimmung geeigneter unterer Schranken für die erreichbare Gesamtenergie partieller Lösungen ist dabei eine Voraussetzung, um das Problem in akzeptabler Zeit lösen zu können. Dies ermöglicht prinzipiell die Anwendung von Branch & Bound Techniken. Allerdings ist die verwendete Bewertungsfunktion relativ komplex und sowohl anfällig für kleine strukturelle Unterschiede zwischen Fragmenten als auch für Unterschiede in der Geometrie eines einzelnen Fragments. Dies resultiert bereits für einzelne Wechselwirkungen in zum Teil drastischen Änderungen des berechneten Energiewertes. Eine genauere und robustere Abschätzung des erreichbaren Energiebeitrags erfordert wesentlich zeitaufwändigere Verfahren wie beispielsweise Simulationstechniken.

Die Bestimmung einer effizient berechenbaren und effektiven Schranke mit der verwendeten Bewertungsfunktion scheint selbst für den Konfigurationsraum eines einzelnen Liganden nicht generell möglich zu sein [Griewel & Rarey, 2005]. Schon für sehr kleine bis mittelgroße Fragmenträume mit einigen Dutzend bis wenigen Hundert Fragmenten liegt die Anzahl prinzipiell möglicher Moleküle zwischen 10^3 und 10^{11} , wenn man die Anzahl „atomarer“ Fragmente pro Molekül auf bis zu fünf bzw. die Anzahl der Anbauzyklen auf bis zu vier beschränkt. Für typische Fragmenträume mit mehreren Tausend Fragmenten ergeben sich für diesen Fall 10^{13} und mehr theoretisch zugängliche Moleküle (siehe dazu Kapitel 6.2). Dies zeigt deutlich, dass eine vollständige Durchsuchung des Lösungsraums auch längerfristig nicht möglich sein wird.

Eine Alternative zur Verwendung exakter Verfahren ist die Anwendung heuristischer Methoden. In diesem Zusammenhang sprechen einige weitere Gründe für die Verwendung eines heuristischen Verfahrens.

- Das Optimum eines de-novo-Design-Verfahrens lässt sich praktisch nicht definieren. Die Beschränkung des Fragment-Baums auf eine maximale Höhe hat bereits zur Folge, dass eine Bestimmung der optimalen Lösung nicht mehr sichergestellt werden kann.
- Der Algorithmus zum iterativen Aufbau von Molekülen ist nur ein Teil des gesamten de-novo-Design-Verfahrens. Bereits der verwendete Algorithmus zur Behandlung der Platzierung einzelner Fragmente ist heuristisch, sodass selbst bei der Bestimmung des Optimums des Fragment-Aufbau-Problems nicht auf die Optimalität der Gesamtlösung geschlossen werden kann.
- Die verwendeten chemischen Modelle ebenso wie die Bewertungsfunktion können die realen Vorgänge nur annähernd und unvollständig wiedergeben. Somit wäre

eine optimale Lösung des eingeschränkten Fragment-Baums lediglich bezüglich der verwendeten Bewertungsfunktion sowie der zugrunde liegenden Modelle optimal.

- Ziel des de-novo-Design-Algorithmus sollte es deshalb sein, hochgradig komplementäre Moleküle mit niederenergetischen Platzierungen zu bestimmen, die möglichst viele weitere Randbedingungen erfüllen, wie z.B. synthetische Zugänglichkeit und pharmakologische Eigenschaften.

Eine einfache, robuste und effiziente Heuristik zur Behandlung des oben beschriebenen Lösungsraums ist der Greedy-Algorithmus, in dessen Verlauf sukzessive jede Erweiterungsmöglichkeit expandiert und der resultierenden Teillösung ein Wert zugewiesen wird, der im Laufe des Algorithmus nicht mehr verworfen wird. Der Algorithmus entscheidet sich somit nach jeder Iteration für die lokal optimale Teillösung. Um dieses Kriterium etwas abzuschwächen und gleichzeitig einen größeren Bereich des Lösungsraums zu berücksichtigen, können in jedem Schritt die k besten Teillösungen ausgewählt werden. Dieses Vorgehensweise wird im Folgenden als k -Greedy-Algorithmus bezeichnet (vgl. [Rarey *et al.*, 1996b]).

Verwendet man den k -Greedy-Algorithmus zum iterativen Aufbau von Molekülen, so werden die Teillösungen sukzessive mit kompatiblen „atomaren“ Fragmenten des Fragmentraums erweitert. Der Algorithmus findet genau dann die optimale Lösung, wenn die daraus ableitbare Teillösung Bestandteil der k besten Lösungen jedes Anbauschnitts ist.

Der k -Greedy-Algorithmus bietet einige weitere Vorteile: So lässt sich durch Anpassung des Parameters k die Laufzeit gegenüber der Abdeckung des Lösungsraums, d.h., der Anzahl der generierten Teillösungen, einstellen. Darauf wird bei der Diskussion der Ergebnisse näher eingegangen. Bei näherer Betrachtung sprechen einige weitere Argumente für die verwendete Heuristik.

- Ziel eines strukturbasierten de-novo-Design-Verfahrens ist die Bestimmung komplementärer Moleküle in niederenergetischen Bindungsmodi. Das beschriebene Verfahren ist in besonderem Maße dazu in der Lage, diese Aspekte zu berücksichtigen. So wird in jedem Anbauschnitt sichergestellt, dass Teillösungen mit lokal optimaler Komplementarität und Geometrie bestimmt und, sofern diese erweiterbar sind, für den nächsten Anbauzyklus verwendet werden.
- Die Lösungen eines strukturbasierten de-novo-Design-Verfahrens sollten neben den oben beschriebenen Eigenschaften zusätzlich eine gewisse molekulare Diversität aufweisen. Durch Verwendung und Anpassung des Parameters k kann somit sowohl gewährleistet werden, dass lokal nicht optimale Teillösungen nicht direkt verworfen werden, als auch sichergestellt werden, dass in jedem Schritt eine gewisse strukturelle Variabilität erhalten bleibt.
- Aufgrund der hohen Subjektivität bei der Beurteilung der Qualität der Lösungen eines de-novo-Design-Ansatzes lassen sich viele weitere benutzerdefinierte Randbedingungen einfach und direkt in den Algorithmus integrieren. Dazu zählen physikochemische Eigenschaften der Fragmente und Moleküle, zusätzliche geometrische

Kriterien für Platzierungen, Pharmakophor-Informationen und Kriterien zur Realisierung struktureller Diversität.

Auf die Einschränkungen und die Anwendbarkeit des k -Greedy-Algorithmus wird noch einmal in Kapitel 7.3 eingegangen.

5.3.2 Globale Randbedingungen

Der de-novo-Design-Algorithmus kann einige weitere voneinander unabhängige Randbedingungen berücksichtigen, die direkt bei der Auswahl der Teillösungen verwendet werden und die Qualität der Ergebnisse erhöhen sollen. Diese Randbedingungen basieren auf den in Kapitel 4.3 vorgestellten Modellen und werden vom Benutzer spezifiziert.

Zum einen können Bereiche für physikochemische Eigenschaften sowie chemische Substrukturen in Form einer Filter-Definition festgelegt werden, die dann zum Teil bereits vor der Erzeugung von Zwischenlösungen überprüft werden können. Zum anderen wird sichergestellt, dass die physikochemischen Eigenschaften aller finalen Lösungen innerhalb der spezifizierten Bereiche liegen.

Weiterhin kann ein Pharmakophor G definiert werden, das von allen generierten Platzierungen erfüllt werden muss. Bei der Spezifikation des Pharmakophors ist zu beachten, dass dieser, im Gegensatz zu einem für die initiale Fragment-Platzierung verwendeten Pharmakophor, wesentlich mehr Bedingungen enthalten kann und sollte. Dabei ist allerdings von Bedeutung, dass der Algorithmus die Bedingungen schrittweise erfüllen können muss, d.h., in jeder Iteration zusätzlich eine oder mehrere Einzelbedingungen. Darauf wird in den entsprechenden Passagen dieses Kapitels im Detail eingegangen.

Da der Algorithmus dazu tendiert, sehr ähnliche Lösungen zu generieren, kann der Benutzer anhand zweier Parameter die Diversität der generierten Teillösungen und Moleküle steuern. Für die Menge der Zwischenlösungen in jedem Anbauzyklus kann die Häufigkeit des Auftretens jedes „atomaren“ Fragments auf ein Maximum beschränkt werden. So kann eine gewisse strukturelle Diversität gleich zu Beginn des Algorithmus gewährleistet werden. Für die finale Lösungsliste kann die maximale Anzahl gemeinsamer bzw. eine minimale Anzahl unterschiedlicher „atomarer“ Fragmente zwischen jedem Paar von Molekülen spezifiziert werden (vgl. [Rarey, 2001] und siehe dazu auch Anh. A.2.2). Im Fall von Konflikten wird jeweils die besser bewertete (Teil-)Lösung beibehalten.

5.3.3 Erweiterung einer Menge von Teillösungen

An dieser Stelle wird zunächst die Vorgehensweise während einer Iteration des k -Greedy-Algorithmus beschrieben. Im Gegensatz zum Docking-Problem, bei dem nur vollständig aufgebaute Liganden Teil der finalen Lösungsmenge sind, stellt beim de-novo-Design praktisch jede erzeugte Teillösung auf jeder Ebene des Fragment-Baums gleichzeitig auch eine valide finale Lösung dar. Dies wird während der Evaluierung der Teillösungen entsprechend beachtet.

Zu Beginn der i -ten Iteration stehen die folgenden Informationen zu Verfügung: Eine Menge F_{i-1} von maximal k_F verschiedenen Fragmenten, jeweils bestehend aus i „atoma-

ren“ Fragmenten, zusammen mit jeweils einer Menge $P_{i-1}^{k_{F_j}}$ von maximal k_P verschiedenen Platzierungen.

Im Laufe der i -ten Iteration werden alle Teillösungen F_{i-1} in allen Platzierungen $P_{i-1}^{k_{F_j}}$ mit allen kompatiblen „atomaren“ Fragmenten in allen Kombinationsmöglichkeiten gemäß der Spezifikation der Link-Kompatibilitäten erweitert.

Die neue Teilergebnismenge F_i dieser Iteration besteht aus maximal k_F verschiedenen Fragmenten, die jeweils aus $i + 1$ „atomaren“ Fragmenten bestehen und eine Menge $P_i^{k_{F_j}}$ von maximal k_P verschiedenen Platzierungen haben.

Die finale Ergebnismenge F_{final} besteht aus maximal k_L verschiedenen Molekülen mit jeweils maximal k_P verschiedenen Platzierungen. Folgende Elemente werden für F_{final} in Betracht gezogen:

- Alle in Iteration i erzeugten Moleküle, die aufgrund fehlender Link-Atome nicht mehr durch Anbau von „atomaren“ Fragmenten erweitert werden können.
- Alle in Iteration i erzeugten Fragmente, die aufgrund ihrer Bewertung und anhand des k -Greedy-Kriteriums direkt von F_i ausgeschlossen werden können, nachdem alle Link-Atome durch die entsprechende terminale Gruppe ersetzt worden sind. (siehe unten)
- Alle Teillösungen der Menge F_{i-1} nach Abschluss der Iteration i und nachdem alle Link-Atome durch die entsprechende terminale Gruppe ersetzt worden sind.

5.3.4 Schritte zum Anbau eines Fragments

In diesem Kapitel werden alle Einzelschritte zum Anbau eines Fragments an eine Teillösung beschrieben.

Zu Beginn des Anbaus stehen die folgenden Informationen zur Verfügung: Das zu erweiternde Fragment F^{ref} bestehend aus i „atomaren“ Fragmenten und n Komponenten $c_0^{ref}, \dots, c_{n-1}^{ref}$, zusammen mit einer Menge P^{ref} von maximal k_P verschiedenen Platzierungen sowie das zu substituierende Link-Atom L^{ref} .

Daran wird das „atomare“ Fragment F^{atm} bestehend aus m Komponenten $c_0^{atm}, \dots, c_{m-1}^{atm}$ und Verwendung des Link-Atoms L^{atm} angebaut. Daraus wird ein neues Fragment F^{cur} erzeugt, das aus $i + 1$ „atomaren“ Fragmenten und $l = n + m - 2$ Komponenten $c_0^{cur}, \dots, c_{l-1}^{cur}$ besteht, zusammen mit einer Menge P^{cur} von maximal k_P verschiedenen Platzierungen. Dabei entsprechen die Komponenten $c_0^{cur}, \dots, c_{l-m+1}^{cur}$ des neuen Fragments F^{cur} den Komponenten $c_0^{ref}, \dots, c_{n-2}^{ref}$ der ursprünglichen Teillösung F^{ref} .

Sofern der Benutzer ein Pharmakophor spezifiziert hat, werden die bereits erfüllten Bedingungen von G ermittelt. Dabei müssen die Informationen für jede Teillösung individuell bestimmt werden. Deshalb leitet sich aus der Menge der Platzierungen P^{ref} für das zu erweiternde Fragment F^{ref} ein entsprechendes lokales Pharmakophor G^{ref} ab. Gemäß der in Kapitel 4.3.5 vorgestellten Pharmakophor-Definition behalten alle bereits durch die Platzierungen P^{ref} erfüllten Bedingungen ihre ursprüngliche Gewichtung aus

G (*essenziell* bzw. *optional*). Alle noch nicht erfüllten Bedingungen werden in G^{ref} auf *optional* gesetzt.

5.3.4.1 Überprüfung initialer Filterkriterien

Sofern die Anzahl der Elemente der Teillösungen in der Menge F_i bereits den Wert k_F erreicht hat, kann der mithilfe der Bewertungsfunktion bestimmte Beitrag der bisher schlechtesten Lösung aus F_i als weiteres Auswahlkriterium verwendet werden. Dabei kann für den Score-Beitrag der potenziellen Teillösung F^{cur} eine echte obere Schranke bestimmt werden, die sich aus dem Score-Beitrag der besten Platzierung von F^{ref} und dem maximal erreichbaren Score von F^{atm} zusammensetzt. Letzterer Wert ergibt sich entweder durch eine vorherige Platzierung des Fragments (siehe dazu Kapitel 5.3.6) bzw. aus der Summe der maximalen Energiebeiträge aller Komponenten (während der Initialisierung bestimmt). Sollte der Wert schlechter sein als der, der bisher schlechtesten Lösung aus F_i , kann die potenzielle Teillösung F^{cur} direkt verworfen werden.

Sofern der Benutzer Wertebereiche für physikochemische Eigenschaften in Form eines Eigenschaftsfilters spezifiziert hat (siehe dazu Kapitel 4.3.3), können diese zum Teil bereits vor dem tatsächlichen Anbau des Fragments F^{atm} an die Teillösung F^{ref} als Auswahlkriterium herangezogen werden. Dafür muss für das „atomare“ Fragment F^{atm} zunächst die Basis-Initialisierung und der Fragment-Deskriptor berechnet werden, sofern das noch nicht geschehen ist (siehe dazu Kapitel 4.3.2). Die oberen Schranken aller strikt additiven Eigenschaften können durch Aufsummierung der Werte von F^{ref} und F^{atm} exakt überprüft werden. Sollte es dadurch zu einer Verletzung der Filter-Spezifikation kommen, kann die potenzielle Teillösung F^{cur} direkt verworfen werden.

Die Effektivität der Filter-Funktionen wird im Zuge der Laufzeitanalysen in Kapitel 7.3 ausführlich diskutiert.

5.3.4.2 Fragmentverknüpfung und Initialisierung

Die neue Teillösung F^{cur} wird erzeugt, indem zunächst die Teillösung F^{ref} dupliziert und daran F^{atm} angebaut wird. In diesem Schritt werden auch die Eingabe-Koordinaten der neu hinzukommenden (duplizierten) Atome von F^{atm} angepasst, sodass eine niedereenergetische initiale Geometrie gewährleistet werden kann. Dies ist insbesondere für die nachfolgende Initialisierung wichtig, da dort die Volumina der einzelnen Komponenten bestimmt werden, die für den späteren inkrementellen Anbau von Bedeutung sind.

Da sowohl F^{ref} als auch F^{atm} selbst in einer niedereenergetischen Konformation vorliegen ist es ausreichend, nur die im Zuge der Verknüpfung neu entstandene Bindung sowie die dazugehörigen Winkel anzupassen und die restlichen Teile der Fragmente als starr anzusehen. Die entsprechenden Werte werden entweder der Fragmentraum-Spezifikation entnommen oder aus der vorliegenden Sybyl-Typisierung abgeleitet. Abbildung 5.3 zeigt eine schematische Darstellung der Geometrieanpassung bei der Fragmentverknüpfung.

In sehr seltenen Fällen kann es dazu kommen, dass mithilfe der oben beschriebenen Vorgehensweise eine Konformation erzeugt wird, in der die Atome des Fragments mit sich selbst überlappen. In einem solchen Fall können die Koordinaten mithilfe eines

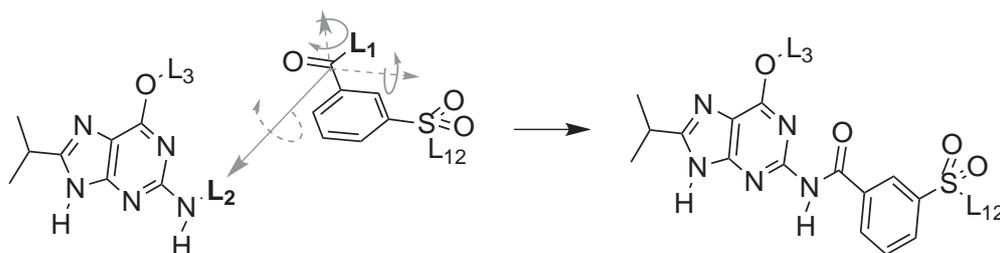


Abbildung 5.3: Geometrieangepassung bei der Fragmentverknüpfung: Das Link-Atom des anzubauenden Fragments und das dazu adjazente Atom werden auf das Link-Atom des zu erweiternden Fragments bzw. das dazu adjazente Atom superpositioniert (schematisch angedeutet durch einen Translationsvektor, um den optional auch eine Rotation stattfinden kann). Zwischen den jeweils adjazenten Atomen wird eine neue Bindung ausgebildet, wobei sich die Bindungslänge aus der Sybyl-Typisierung ableitet. Aus den Bindungsvektoren zu den Link-Atomen sowie den lokalen Geometrien der beteiligten adjazenten Atome lässt sich eine Rotationsmatrix ableiten (schematisch angedeutet durch zwei unterbrochen gezeichnete Rotationsachsen), mit deren Hilfe eine lokale niederenergetische Bindungsgeometrie erzeugt werden kann.

externen Programms (z.B. Corina) neu berechnet werden, sofern das vom Benutzer so spezifiziert worden ist. Sollte danach immer noch keine überlappungsfreie Konformation erzeugt worden sein, wird die Teillösung F^{cur} verworfen.

Bei der Erzeugung von F^{cur} kann es vorkommen, dass ein neues Stereozentrum erzeugt wird oder ein bereits existierendes seine Konfiguration ändert. In einem solchen Fall wird bezüglich der Eingabe-Koordinaten prinzipiell ein willkürliches Stereoisomer erzeugt. Auf eine explizite Generierung aller möglichen Stereoisomere wird aus den in Kapitel 4.3.4 genannten Gründen verzichtet. Stattdessen werden im Laufe des inkrementellen Aufbaus der Komponenten des Fragments alle möglichen Stereoisomere generiert, sofern der Benutzer dies so spezifiziert hat (siehe dazu auch Anh. A.2.1). Die Stereo-Informationen liegen dann in der Menge der resultierenden Platzierungen P^{cur} vor.

Das neu erzeugte Fragment wird gemäß der in Kapitel 4.3.2 beschriebenen Vorgehensweise vollständig initialisiert. In diesem Zusammenhang werden zusätzlich topologisch identische Link-Atome anhand ihrer Äquivalenzklassen identifiziert und markiert, sodass in den folgenden Anbausritten keine redundanten Teillösungen erzeugt werden. Des Weiteren werden kanonische SMILES erzeugt sowie der physikochemische Deskriptor berechnet, sofern eine benutzerdefinierte Filter-Spezifikation vorliegt.

Für F^{cur} wird analog zu F^{ref} ein individuelles lokales Pharmakophor G^{cur} bestimmt. Dieses leitet sich aus G^{ref} ab, wobei die Definition so angepasst wird, dass in G^{cur} die Abdeckung mindestens einer noch nicht erfüllten optionalen Bedingung gefordert wird. Letzteres geschieht allerdings nur, falls dadurch die Anzahl der in G global geforderten essenziellen und optionalen Bedingungen nicht überschritten wird.

Durch den inkrementellen Anbau der noch nicht platzierten Komponenten wird so eine Menge von Platzierungen P^{cur} generiert, wovon alle maximal k_P Elemente die Pharmakophor-Definition G^{cur} erfüllen.

5.3.4.3 Überprüfung zusätzlicher Filterkriterien

Einige der optional vom Benutzer definierbaren Filterkriterien können erst nach der eigentlichen Erzeugung von F^{cur} überprüft werden.

Dies trifft zum einen auf Ausschluss-SMARTS zu. Diese müssen auf den letztendlich resultierenden Fragment-Graphen angewandt werden, da eine entsprechende Substruktur erst durch den Anbau von F^{atm} an F^{ref} zustande gekommen sein kann.

Zum anderen ist dies für die spezifischen Pharmakophor-Informationen von F^{cur} der Fall. In diesem Zusammenhang werden Distanz-Matrizen der maximalen bzw. minimalen Abstände aller wechselwirkenden Gruppen untereinander berechnet, die sich aus den jeweiligen Wechselwirkungsgeometrien und den möglichen Konformationen von F^{cur} ableiten lassen. Anhand dieser Information wird durch Geometrie-Betrachtungen überprüft, ob F^{cur} prinzipiell das lokale Pharmakophor G^{cur} erfüllen kann. Sollte das nicht der Fall sein, kann F^{cur} direkt verworfen werden (vgl. [Hindle *et al.*, 2002]).

5.3.4.4 Transfer der Teilplatzierungen

Bevor das neu angebaute Fragment F^{atm} bzw. die entsprechenden Komponenten platziert werden können, werden die Informationen über die Platzierung und Wechselwirkungen von F^{ref} auf die neue Teillösung F^{cur} übertragen.

Der erste dafür notwendige Schritt beinhaltet analog zum Protein-Ligand-Docking die Auswahl einer geeigneten Basiskomponente. Im vorliegenden Fall bilden die ursprünglich aus F^{ref} stammenden Komponenten $c_0^{cur}, \dots, c_{l-m+1}^{cur}$ sowie die L^{ref} ersetzende bzw. ursprünglich zu L^{atm} benachbarte Komponente die Basiskomponente von F^{cur} .

In einem zweiten Schritt werden analog zum Protein-Ligand-Docking die Basisplatzierungen von F^{cur} erzeugt. Dies geschieht durch Übertragung der Platzierungen und dazugehörigen Wechselwirkungen von F^{ref} . Die Daten zu den Platzierungen sowie den Wechselwirkungen liegen jeweils in einer miteinander verwobenen Baumstruktur vor und werden mithilfe einer *Konformationsaufbau-Struktur* verwaltet [Rarey *et al.*, 1996a].

Da für F^{cur} die Historie der vorherigen Platzierungsschritte der einzelnen Komponenten nicht relevant ist, wird aus Gründen der Speicherplatzeffizienz lediglich die letztendlich resultierende Liste der Platzierungen übertragen, d.h., die Blätter des Platzierungsbaums von F^{ref} . Dies geschieht durch iterative Übertragung und Speicherung der dazugehörigen Koordinaten aller Atome F^{ref} und aller weiteren relevanten Informationen in der neu erzeugten Konformationsaufbau-Struktur von F^{cur} . Da die einzelnen Platzierungen auf zum Teil unterschiedlich ausgebildeten Wechselwirkungen beruhen, wird der Wechselwirkungsbaum komplett übertragen.

Durch die Substitution von L^{ref} kann es vorkommen, dass sich für das benachbarte und ursprünglich bereits mit F^{ref} platzierte Atom der Komponente c_{l-m+1}^{cur} der Wechselwirkungstyp sowie die Wechselwirkungsgeometrie ändert. In diesem Fall können dem Atom

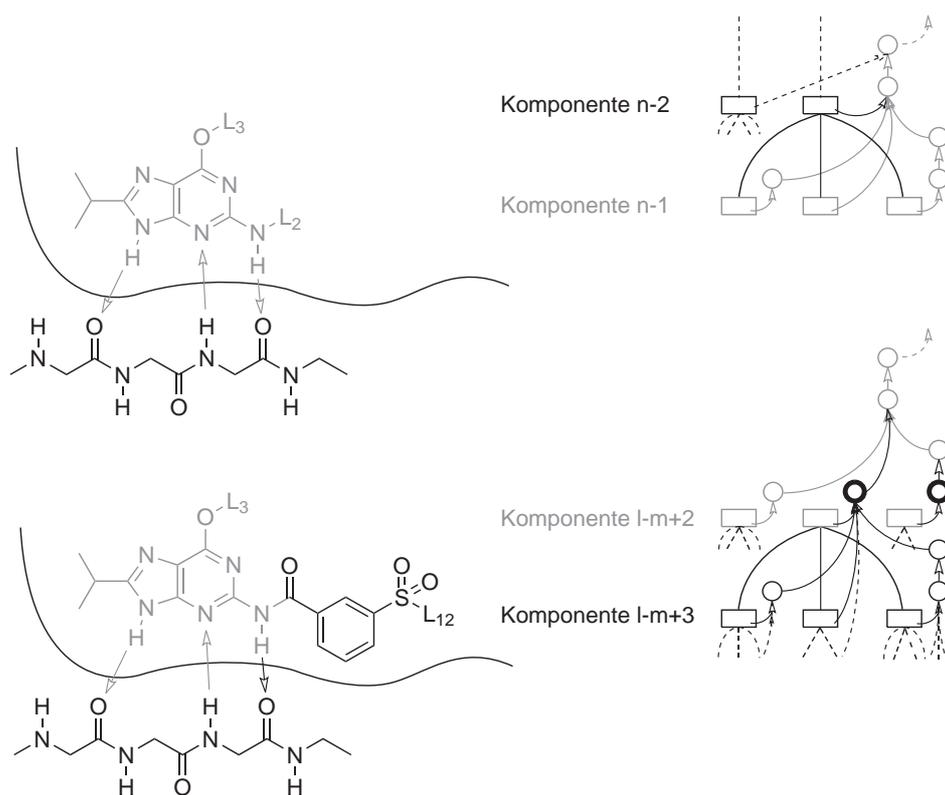


Abbildung 5.4: Übertragung der Platzierungs- und Wechselwirkungsinformation. Die Positionierung eines Start-Fragments in der Bindetasche und dessen Erweiterung mit einem zusätzlichen Fragment sind angedeutet, zusammen mit einem Ausschnitt der dazugehörigen Aminosäurekette und ausgewählten gerichteten Wechselwirkungen (durch Pfeile im linken Teil der Abb. symbolisiert). Im Schema auf der rechten Seite entsprechen dabei die Rechtecke den Platzierungen und die Kreise den ausgebildeten Wechselwirkungen. Im oberen Teil sind die letztendlich resultierenden Platzierungen des Fragments mit den dazugehörigen Wechselwirkungen nach Anbau aller Komponenten grau dargestellt. Diese Informationen werden übertragen und entsprechen den Basis-Platzierung der neuen Teillösung im unteren Teil. Die übertragenen Platzierungen und Wechselwirkungen sind deshalb im rechten unteren Teil ebenfalls grau dargestellt. Die fett gezeichneten schwarze Kreise (und die dazugehörigen Pfeile) symbolisieren die neu zu berechnenden Wechselwirkungen, die sich aus einer eventuellen Änderung der Wechselwirkungstypen und -geometrien ergeben. Die so übertragenen Basis-Platzierungen werden dann durch schrittweisen Aufbau der noch nicht platzierten Komponenten erweitert (durch unterbrochene Pfeile markiert).

nicht die entsprechenden Wechselwirkungsinformationen der Komponente c_{n-2}^{ref} aus F^{ref} zugeordnet werden. Vielmehr müssen sie neu berechnet, mit der Bewertungsfunktion evaluiert und gegebenenfalls entsprechend in die Baumstruktur eingefügt werden.

In jedem Fall müssen für die das Link-Atom L^{ref} substituierende Komponente c_{l-m+2}^{cur} potenzielle Wechselwirkungen neu berechnet, mit der Bewertungsfunktion evaluiert und gegebenenfalls an entsprechender Stelle in die Baumstruktur eingefügt werden. Weiterhin müssen die Koordinaten der Komponente gemäß der tatsächlichen Bindungslänge angepasst werden. Diese leiten sich zwar von L^{ref} ab, das ursprünglich mit F^{ref} platziert worden ist, allerdings kann L^{ref} weder Wechselwirkungen ausbilden, noch entspricht der Abstand zum adjazenten Atom dem der neu gebildeten Bindung.

Abbildung 5.4 zeigt eine schematische Darstellung der Vorgehensweise beim Transfer der Teilplatzierungen und stellt dabei die wesentlichen Aspekte heraus.

Sofern der Benutzer eine Pharmakophor-Definition spezifiziert hat, werden die Informationen über bereits erfüllte Bedingungen von G^{cur} in einem anschließenden Schritt an den Platzierungen P^{cur} der Basiskomponente annotiert, sodass sie in den folgenden inkrementellen Anbausritten der noch nicht platzierten Komponenten berücksichtigt werden können.

5.3.4.5 Inkrementeller Anbau der Komponenten

Zum Anbau der noch nicht platzierten Komponenten von F^{cur} wird der inkrementelle Aufbaualgorithmus von FlexX in entsprechender Reihenfolge durchlaufen, wobei an einigen Stellen spezifische Anpassungen für die Behandlung von Fragmenten vorgenommen wurden, die im Folgenden erläutert werden.

Analog zum Ligand-Docking erfolgt zunächst der Konformationsaufbau der neuen Komponente, dem sich ein erster Überlappungstest mit dem Rezeptor anschließt. Daraufhin wird nach neuen Wechselwirkungen gesucht, die daraus resultierende Platzierung diesbezüglich optimiert und optional hinsichtlich zusätzlicher geometrischer Kriterien sowie des Pharmakophors evaluiert. Im Anschluss daran wird die Bewertungsfunktion auf die resultierende Platzierung angewandt.

Zusätzlich zu den schon in FlexX enthaltenen geometrischen Filterkriterien für Platzierungen wurde die Überprüfung zweier weiterer wichtiger Kriterien integriert. Diese können erst nach dem Aufruf der Bewertungsfunktion angewandt werden, da sie energetisch bewertete Wechselwirkungen voraussetzen. Beide Kriterien sollen bekannte Schwachstellen in der verwendeten Bewertungsfunktion adressieren, in der Desolvationseffekte nur unzureichend berücksichtigt und entsprechende energetisch ungünstige Platzierungen nicht bestraft werden.

- Zum einen sollen Platzierungen erkannt werden, bei denen polare Atome *vergraben*, d.h. in einer Kavität der Bindetasche platziert werden, dort aber keinerlei Wasserstoffbrücken ausbilden. Die beteiligte funktionale Gruppe wird dabei desolvatisiert, der dadurch entstehende Energieverlust aber nicht durch entsprechende neu ausgebildete Wechselwirkungen kompensiert. Eine solche Konstellation ist energetisch sehr ungünstig und sollte nicht Bestandteil der Lösungsmenge sein. Allerdings wird

diese Platzierung in der verwendeten Bewertungsfunktion nicht bestraft und muss anhand geometrischer Kriterien identifiziert werden.

- Zum anderen soll das Wechselwirkungspotenzial des Fragments möglichst optimal ausgenutzt werden. In der verwendeten Bewertungsfunktion werden nicht wechselwirkende hydrophile Atome jedoch nicht bestraft. Eine solche Konstellation ist zwar energetisch ungünstig, kann aber zu einem gewissen Grad toleriert werden. Sie sollte aber einen bestimmten Schwellenwert nicht unterschreiten. Da dieser sehr stark von der Umgebung und der Natur der Bindetasche abhängt, muss er durch den Benutzer vorgegeben werden. Um eine bestimmte *Effizienz* zu gewährleisten, wird die Anzahl aller theoretisch möglichen Wasserstoffbrücken-Wechselwirkungen des Fragments zu der tatsächlich ausgebildeten Anzahl ins Verhältnis gesetzt. Unterschreitet dieses Verhältnis den definierten Grenzwert, wird die Platzierung verworfen.

Eine detailliertere Auflistung der verfügbaren Filterfunktionen ist in Anhang A.2.2 zu finden. Neben inhaltlichen Erweiterungen wurden auch einige spezifische konzeptionelle Anpassungen am inkrementellen Aufbaualgorithmus vorgenommen, die aus Effizienz- und Konsistenzgründen für Fragmente notwendig sind.

Um die Stabilität des für die Sortierung der Platzierungslisten verwendeten Heap-Sort Algorithmus zu gewährleisten, ist es notwendig, die einzelnen Platzierungen in einer eindeutigen Reihenfolge abzuspeichern. Falls zwei Platzierungen nicht anhand des berechneten Energiewertes unterschieden werden können, geschieht dies nun anhand einer eindeutigen Nummer. Dadurch wird nach dem Clustering der Platzierungen nach dem Anbau jeder Komponente ein stabiles Sampling der Torsionswinkel für ausgehende Bindungen zu noch nicht angebauten Komponenten gewährleistet.

Bei der Evaluation des Pharmakophors kann es vorkommen, dass F^{cur} eine größere Menge optionaler Einzelbedingungen im lokalen Pharmakophor erfüllt, als dies gemäß der globalen Definition G erlaubt ist (vgl. Kap. 4.3.5). Allerdings kann sich die Anzahl erfüllter optionaler Einzelbedingungen in späteren Fragment-Anbausritten und den damit verbundenen Optimierungen der Platzierungen noch ändern. Sofern F^{cur} also in einer nachfolgenden Iteration noch erweiterbar ist werden die entsprechenden Platzierungen an dieser Stelle nicht verworfen. Die Pharmakophor-Informationen werden nach erfolgtem inkrementellen Aufbau von F^{cur} aus Effizienzgründen an der Teillösung annotiert, sodass sie für spätere Fragment-Anbausritte sowie die finale Überprüfung bezüglich der globalen Pharmakophor-Definition G verwendet werden können und nicht neu berechnet werden müssen.

Für das finale Clustering der Platzierungen nach dem Anbau aller ursprünglich noch nicht platzierten Komponenten ist eine Fallunterscheidung notwendig. Sofern F^{cur} in einer nachfolgenden Iteration des de-novo-Design-Algorithmus noch erweitert werden kann, d.h., F^{cur} noch offene Link-Atome hat, wird der RMS-Schwellenwert für Zwischenlösungen für das Clustering verwendet, ansonsten der RMS-Schwellenwert für finale Lösungen (vgl. [Rarey, 2008]). Dabei werden die Link-Atome gesondert berücksichtigt, was dazu führt, dass insbesondere die Torsionswinkel für die ausgehende Bindung zu Link-Atomen in einer wesentlich feineren Abstufung vorliegen.

5.3.4.6 Abschließende Evaluation

Sofern nach Anbau der letzten Komponente an F^{cur} die Menge der Platzierungen P^{cur} nicht leer ist, werden die abschließenden Evaluationsschritte durchgeführt. Ansonsten wird F^{cur} verworfen.

Aus Gründen der Speicherplatzeffizienz kann die letztendlich resultierende Liste der Platzierungen nach Abschluss des inkrementellen Aufbaus in ihrer Länge beschränkt werden. Dies bezieht sich allerdings nur auf die Anzahl der behaltene Platzierungen und ist unabhängig von dem Wert des algorithmischen Parameters k_P . Diese Beschränkung ist besonders bei Abdeckung eines großen Teils des Fragment-Baums notwendig, d.h. für große Werte von k_F . Auf den Einfluss der Länge der Platzierungsliste auf die Resultate des de-novo-Design-Algorithmus wird in Kapitel 7.3 näher eingegangen.

Sofern F^{cur} in nachfolgenden Fragment-Anbausritten noch erweitert werden kann, also noch über Link-Atome verfügt, wird F^{cur} nach Überprüfung des Diversitätskriteriums der Menge F_i hinzugefügt.

Falls F_i bereits k_F Elemente enthält und F^{cur} aufgrund des k -Greedy-Kriteriums nicht Bestandteil der Menge F_i sein kann, wird stattdessen versucht, F^{cur} der Menge F_{final} hinzuzufügen. Dies ist insbesondere dann von Bedeutung, wenn nur wenige Anbauzyklen durchgeführt werden und k_F sehr viel kleiner ist als k_L .

5.3.5 Die finale Lösungsliste

Bevor versucht wird, ein Element der Menge F_{final} hinzuzufügen, werden (sofern erforderlich) alle Link-Atome durch die entsprechenden terminalen Gruppen ersetzt. Dabei werden die im vorherigen Abschnitt beschriebenen Einzelschritte des inkrementellen Aufbaualgorithmus für Komponenten in gleicher Weise durchlaufen mit dem Unterschied, dass zunächst alle Link-Atome mit der jeweiligen terminalen Gruppe substituiert werden und erst danach die übrigen Schritte folgen. Dabei ergeben sich einige Besonderheiten.

Bei der Überprüfung der physikochemischen Eigenschaften werden jetzt die oberen und unteren Schranken der spezifizierten Bereiche herangezogen und das Molekül zusätzlich auch auf eventuelle Einschluss-SMARTS getestet. Weiterhin wird anstelle der lokalen Pharmakophor-Definition nun das globale Pharmakophor G verwendet.

Durch die Substitution der Link-Atome mit den terminalen Gruppen kann es dazu kommen, dass das resultierende Molekül im Gegensatz zur vorherigen Teillösung entweder mit sich selbst überlappt, die Filterkriterien nicht mehr erfüllt, ein zu großes Überlappvolumen mit dem Rezeptor hat oder aufgrund des Pharmakophors keine gültigen Platzierungen mehr erzeugt werden können.

Aus Konsistenzgründen kann der Benutzer durch einen Parameter steuern, ob in solchen Fällen sowohl das resultierende Molekül als auch die dazugehörige Teillösung verworfen wird, oder anstatt des Moleküls die Teillösung in die finale Lösungsliste eingefügt wird (siehe dazu auch Anh. A.2.2).

5.3.6 Algorithmische Aspekte

Zu den laufzeitintensiven Funktionen des oben beschriebenen Algorithmus gehören einerseits der inkrementelle Anbau der Komponenten und andererseits die Initialisierung der Teillösungen. Diese Funktionen sind nicht nur an sich komplex, sie werden auch sehr häufig aufgerufen. Aus diesem Grund bedarf es effizienter Verfahren zur Reduktion des durchschnittlichen Rechenaufwandes für Teillösungen. Dazu zählt in erster Linie die Verwerfung von Fragmenten bevor sie initialisiert werden bzw. der inkrementelle Anbau der noch nicht platzierten Komponenten stattfindet. Die vorgestellte Heuristik enthält bereits einiger solcher Konzepte.

- Die Abschätzung des maximal erreichbaren Energiewertes jeder Teillösung liefert eine echte obere Schranke, aufgrund derer die Teillösung gegebenenfalls direkt verworfen werden kann. Die Güte der Abschätzung der maximalen Energiebeiträge der Fragmente ist dabei von zentraler Bedeutung. Der tatsächliche maximale Beitrag eines Fragments wird während der Berechnung im Zuge der Basis-Initialisierung für einen bestimmten Rezeptor in der Regel weit überschätzt. Ein vorheriges Docking-Experiment aller Fragmente liefert jedoch weitaus exaktere Ergebnisse. Im ersten Fall ist die Abschätzung unabhängig von der verwendeten Bindetasche, im zweiten Fall erhält man einen wesentlich präziseren und rezeptorspezifischen Wert.
- Durch die Prozessierung der Start-Fragmente in der Reihenfolge des erreichten Energiewerts der besten dazugehörigen Platzierung, sowie der Sortierung der im Anbau verwendeten „atomaren“ Fragmente nach dem maximal erreichbaren Energiewert, wird das k -Greedy-Kriterium effizient ausgenutzt.
- Aufgrund der verwendeten Filterkriterien für physikochemische Eigenschaften, Substrukturen sowie Pharmakophor-Informationen können Teillösungen gegebenenfalls bereits vor bzw. nach der Basis-Initialisierung verworfen werden.
- Der Zeitaufwand für die Erzeugung der Basis-Platzierungen der Teillösungen fällt durch Verwendung eines effizienten Verfahrens zum Transfer der jeweiligen Informationen im Gegensatz zum Ligand-Docking nicht mehr ins Gewicht.
- Zusätzliche geometrische Filterkriterien während des inkrementellen Anbaus der Komponenten sowie die Beschränkung der Gesamtlänge der resultierenden Platzierungsliste reduziert die Menge der zu prozessierenden Informationen für nachfolgende Anbau-Schritte signifikant.

Neben den laufzeitbestimmenden Faktoren spielt die Speicherplatzeffizienz eine entscheidende Rolle. Der Algorithmus ist prinzipiell eine Breitensuche, weshalb die physikochemischen Daten, die Pharmakophor-Informationen und nicht zuletzt die Platzierungen aller Teillösungen einer Ebene des Fragment-Baums gleichzeitig im Speicher gehalten werden sollten.

6 Erzeugung und Charakterisierung von Fragmenträumen

Die im Rahmen dieser Arbeit entstandenen Ergebnisse lassen sich in drei Teilbereiche gliedern. Im Folgenden entspricht dabei die Reihenfolge der Beschreibung der einzelnen Teilbereiche nicht der chronologischen Abfolge ihrer Bearbeitung, sondern die vorliegende Strukturierung bezieht sich vielmehr auf die inhaltlichen Zusammenhänge.

Zu Beginn steht demnach die Erzeugung von Fragmenten bzw. Fragmenträumen und damit die Auswahl einer geeigneten Datengrundlage, der entsprechenden Shredding-Strategie und insbesondere des dazugehörigen Regelwerks. Dem schließt sich die Charakterisierung und Analyse grundlegender Eigenschaften von Fragmenten und der daraus abgeleiteten Fragmenträume an. Auf die Verwendung von Fragmenträumen für das strukturbasierte de-novo-Design wird in Kapitel 7 eingegangen.

6.1 Erzeugung von Fragmenträumen

Im Rahmen der vorliegenden Arbeit wurde ein verbessertes Konzept für die Spaltung und Verknüpfung retrosynthetisch wichtiger chemischer Motive und der daraus abgeleiteten Fragmenträume entwickelt. Das neue BRICS-Modell soll die zugrunde liegenden chemischen Konzepte besser abbilden und, verglichen mit bisherigen Ansätzen, eine umfassendere Einbeziehung synthetischer Aspekte ermöglichen sowie pharmakologisch relevante strukturelle Informationen berücksichtigen. All dies fließt bereits in die Fragmenterzeugung ein. Des Weiteren soll eine konsistente Behandlung von Fragmenten hinsichtlich ihrer dreidimensionalen Struktur gewährleistet werden. Das ist speziell für das strukturbasierte de-novo-Design entscheidend, da dort explizit mit den Koordinaten der Fragmente gearbeitet wird und die Geometrie der Fragmente selbst sowie der daraus generierten Lösungen von entscheidender Bedeutung ist.

Auf die Modellierung von Fragmenträumen wurde bereits in Kapitel 4.2.3 eingegangen. Im Folgenden wird entsprechend das neu erstellte Modell erläutert und die Unterschiede zu bereits existierenden Regelwerken aufgezeigt. Des Weiteren werden die verwendeten Datensätze vorgestellt und die daraus abgeleiteten Fragmente charakterisiert.

6.1.1 Die verwendete Datengrundlage

Für die Erzeugung der Fragmente wurden Moleküle aus unterschiedlichen Datenquellen herangezogen. Zum einen wurden die zur Erstellung des DUD Datensatzes verwendeten Inhibitoren aus der ZINC Datenbank verwendet, die von Huang et al. in 40 Target-Klassen eingeteilt wurden. Zum anderen wurden Daten aus dem World Drug Index

Tabelle 6.1: Aufbereitete Datenquellen zur Erzeugung der Fragmenträume

Name des Datensatzes	Anzahl Moleküle	Stand (Datum)	Referenz
DUD-Inhibitoren	3961	22.10.2006	[Huang <i>et al.</i> , 2006]
PubChem	16775000	10.11.2007	[PubChem, 2007]
World Drug Index	61203	2004	[Thomson, 2004]
ZINC drug-like	2066906	02.05.2006	[ZINC, 2006]

(WDI), der PubChem Datenbank sowie eine Untermenge kommerziell verfügbarer Strukturen aus der ZINC Datenbank verwendet [Irwin & Shoichet, 2005]. Eine entsprechende Übersicht und die dazugehörigen Referenzen sind in Tabelle 6.1 zu finden.

Bis auf die DUD-Inhibitoren wurden alle Datensätze einer umfangreichen Aufbereitungsprozedur unterzogen. Diese beinhaltete die Entfernung von Metall- und Gegenionen, anorganischen Salzen sowie die Erzeugung konsistenter Protonierungszustände. Weiterhin wurden die enthaltenen Halogene unifiziert, d.h., Iod und Brom durch Chlor ersetzt, Fluor hingegen als solches beibehalten. Auf die explizite Verwendung der schweren Halogene Iod und Brom kann an dieser Stelle verzichtet werden, da sie für die ersten Schritte während der Leitstruktursuche nur eine untergeordnete Rolle spielen und sie außerdem in den chemischen Modellen der folgenden Anwendungen nicht gesondert berücksichtigt werden. Die derart aufbereiteten Datensätze wurden mithilfe kanonischer SMILES [Weininger & Weininger, 1989] unifiziert und aus diesen jeweils eine niederenergetische 3D-Struktur mit Corina erzeugt [Sadowski & Gasteiger, 1993; Sadowski *et al.*, 1994], die im MOL2-Format gespeichert wurde [Tripos, 2008].

6.1.2 Shredding-Prozedur

Die Zerlegung organischer Moleküle in Fragmente erfolgt mithilfe des Programms Recore [Maaß *et al.*, 2007] gemäß der im Kapitel 4.2.2 beschriebenen Shredding-Strategie: Zunächst werden alle zu schneidenden azyklischen Bindungen identifiziert und anschließend simultan unter Einführung unterschiedlicher Link-Atome gespalten. Dabei arbeitet Recore auf den dreidimensionalen Strukturen der Moleküle und erhält dabei die lokale Geometrie der jeweiligen chemischen Umgebungen auf Ebene der Fragmente. Das verwendete Regelwerk besteht aus einer Menge molekularer Subgraphen, die mithilfe der SMARTS-Syntax [Daylight, 2007] formuliert wurden.

Bereits bei der Spezifikation des Regelwerks wurde darauf geachtet, dass keine reaktiven, toxischen oder andere ungewollte chemischen Motive entstehen. Dazu zählen in diesem Zusammenhang unter anderem konjugierte Doppelbindungen, Acetale, Thioacetale, Urethane und Michael-Akzeptoren. Des Weiteren wurde die Erzeugung kleiner endständiger Fragmente vermieden, d.h., Bindungen zu Wasserstoffen, Halogenen, Hydroxy-, Nitro-, Carboxylat-, Methoxy-, Methyl-, Ethyl- oder Isopropyl-Gruppen wurden nicht geschnitten. Die Berücksichtigung der entsprechenden Motive direkt während der Fragmenterzeugung führt dazu, dass die Shredding-Subgraphen sehr genau und mit-

unter redundant formuliert werden müssen, d.h., bestimmte Subgraphen wiederholen sich im linken und rechten Teil der spezifizierten Bindung. Ein weiterer Grund dafür ist die Vorgehensweise des verwendeten Subgraph-Matching Algorithmus, der im Falle von rekursiven Subgraphen die Information über bereits zugeordnete Teile verwirft.

In Abbildung 6.1 sind die Shredding-Substrukturen des BRICS-Regelwerks dargestellt, wobei sie aus Gründen der Anschaulichkeit nicht vollständig abgebildet sind. Die ausführliche Spezifikation der Subgraphen in SMARTS-Syntax ist im Anhang C zu finden.

Neben den Shredding-Regeln kam eine Reihe von Filterkriterien zum Einsatz, anhand derer Fragmente mit mehr als 16 Schweratomen verworfen wurden. Des Weiteren wurden Ringe, die aus mehr als acht Schweratomen bestehen ebenfalls verworfen. Weiterhin wurde eine Reihe von Substruktur-Filtern verwendet, die Fragmente mit reaktiven oder toxischen chemischen Gruppen ausschließen [Kazius *et al.*, 2005]. Dieser zusätzliche Filterschritt ist notwendig, da die entsprechenden Motive, weil sie nicht geschnitten werden, trotzdem in Fragmenten enthalten sein können. Die im Zuge des Shreddings auf Fragmentenebene erzeugten Duplikate werden anhand kanonischer SMILES identifiziert und entfernt. Moleküle ohne Schnittkanten werden ebenso nicht weiter berücksichtigt.

6.1.3 Die verwendeten Shredding-Regelwerke

Das BRICS-Regelwerk wurde zur Erzeugung mehrerer Fragmenträume verwendet, die im Folgenden im Detail beschrieben und analysiert werden. Gleichzeitig wurde innerhalb derselben Shredding-Umgebung das RECAP-Modell nachimplementiert, um eine direkte Vergleichbarkeit der beiden Ansätze zu ermöglichen.

Das RECAP-Regelwerk besteht aus elf chemischen Motiven, deren unmittelbare strukturelle Umgebungen in Abbildung 6.2 dargestellt sind [Lewell *et al.*, 1998]. Die vollständige Spezifikation der nachimplementierten Subgraphen in SMARTS-Syntax ist im Anhang C zu finden. Eine Besonderheit des RECAP-Ansatzes ist, dass die gespaltenen chemischen Motive ausschließlich wieder mit Fragmenten aus derselben Fragmentierungsregel kombiniert werden können. Somit leiten sich aus Abbildung 6.2 gleichzeitig auch die Verknüpfungsregeln für die Fragmente ab.

Eine verallgemeinerte Version des RECAP-Ansatzes wurde von Schneider *et al.* für das ligandbasierte de-novo-Design-Programm TOPAS sowie von Stahl und Rarey im Rahmen der Feature Tree Fragmentraum-Suchmethode (FTree-FS) verwendet [Schneider *et al.*, 2000; Rarey, 2001]. Darin wurden die jeweiligen chemischen Umgebungen beibehalten, jedoch einige der Fragment-Prototypen unifiziert, sodass daraus im Endeffekt zwölf Link-Typen resultieren. Der korrespondierende Fragmentraum wurde bereits in Abbildung 4.6 vorgestellt.

Im Gegensatz zu den obigen Ansätzen berücksichtigt das neue BRICS-Modell eine Reihe weiterer wichtiger Aspekte. Die modellierten chemischen Motive wurden speziell im Hinblick auf medizinalchemische Konzepte erweitert, die Definitionen der chemischen Umgebungen präzisiert und die Fragmente hinsichtlich ihrer Verwendungsmöglichkeiten generalisiert. So werden bioisostere Ersetzungen in Form verbrückender Ether-, Thioether-, Amin-, Carbonyl-, Methyl-, Sulfoxid- und Sulfon-Gruppen explizit modelliert. Des Weiteren wird für Alkyl- und Carbonyl-Fragmente die Information beibehalten,

6 Erzeugung und Charakterisierung von Fragmenträumen

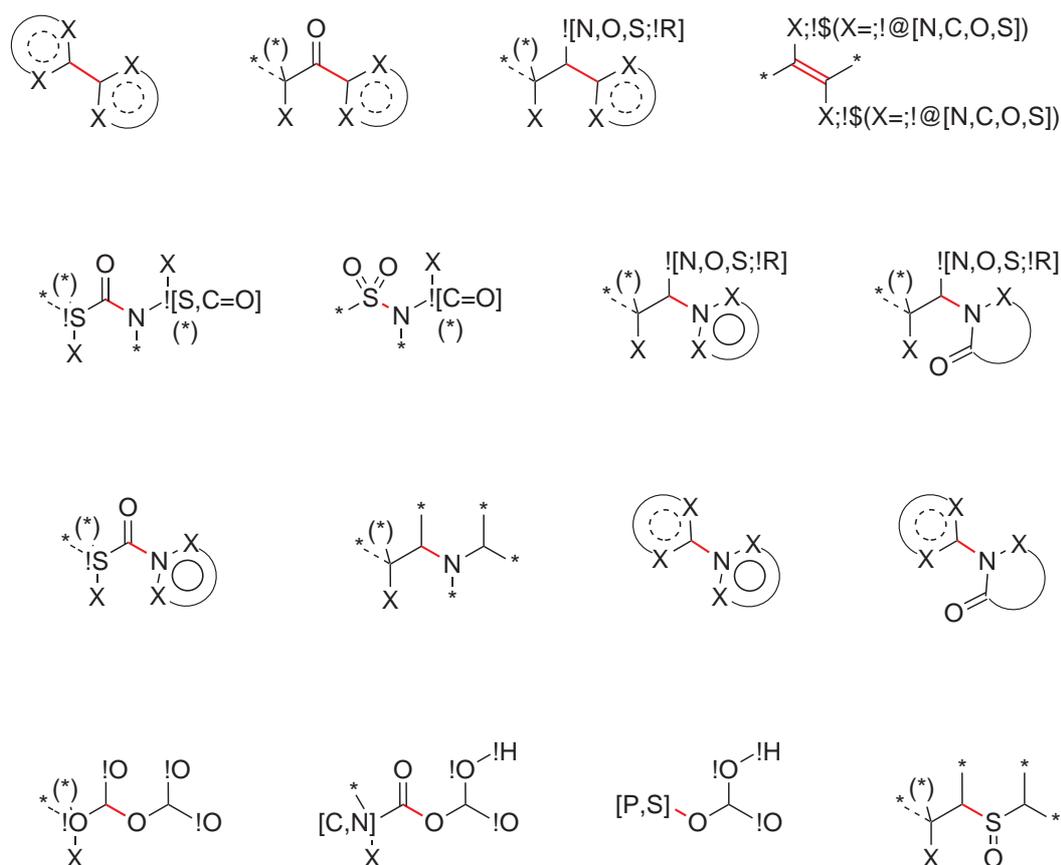


Abbildung 6.1: Die BRICS-Shredding-Subgraphen in vereinfachter Darstellung. Die zu schneidende azyklische Bindung ist jeweils rot markiert. Der Atom-Platzhalter 'X' steht für eines der chemischen Elemente Kohlenstoff, Stickstoff, Sauerstoff oder Schwefel. Der Atom-Platzhalter '*' steht für ein beliebiges Element. Unterbrochen dargestellte azyklische Bindungen spezifizieren entweder eine Einfach- oder eine Doppelbindung, was dazu führt, dass einer der benachbarten Atom-Platzhalter '(*)' aus Valenzgründen optional ist. In Ringen deuten unterbrochene Linien sowie Platzhalter 'X' auf eine Fallunterscheidung zwischen aromatischen und nicht-aromatischen (heterozyklischen) Ringsystemen hin. Grafisch nicht darstellbare Subgraphen sind in dem entsprechenden Atombezeichner zusammengefasst. Zu dem Ether-Subgraphen (unten links) existiert ein entsprechender Thioether-Subgraph, der nicht explizit aufgeführt ist. Der Sulfoxid-Subgraph (unten rechts) schließt den Sulfonyl-Subgraphen ein. Die vollständige Spezifikation in SMARTS-Syntax ist im Anhang C zu finden.

6.1 Erzeugung von Fragmenträumen

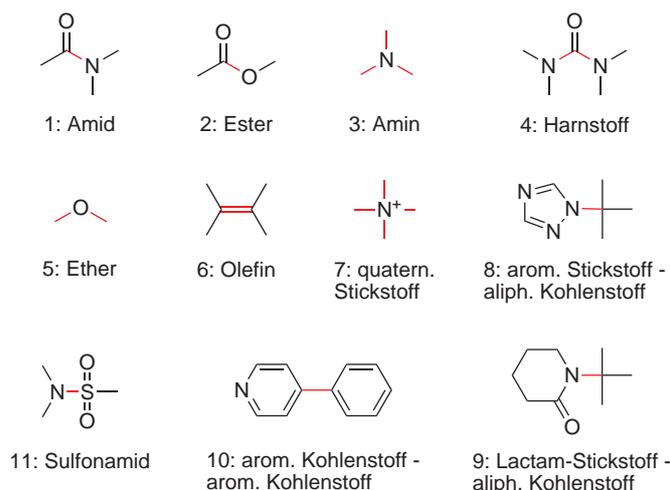


Abbildung 6.2: Die RECAP-Fragmentierungsregeln. Die zu spaltenden azyklischen Bindungen der jeweiligen Motive sind rot markiert. Bei den dargestellten Strukturen handelt es sich lediglich um Beispiele, die eigentlichen Motive sind generischer formuliert, das trifft insbesondere auf die Ringe zu. Die Diversität der Fragmente resultiert aus den nicht dargestellten Substituenten. Die Fragment-Kompatibilität ergibt sich aus den ursprünglichen chemischen Motiven, d.h., jedes Fragment kann nur mit Fragmenten kombiniert werden, die mit derselben Regel erzeugt worden sind. (Adaptiert aus: [Lewell *et al.*, 1998])

ob sie als zyklische oder azyklische Substituenten bzw. verbrückende Gruppen aufgetreten sind (L_4/L_8 bzw. L_1/L_6 in Abb. 6.3).

Auf die Berücksichtigung des quaternären Ammonium-Fragments wird aus mehreren Gründen verzichtet. Prinzipiell hängen Ladungen in erster Linie von der direkten chemischen Umgebung einer Bindetasche und des umgebenden Solvents ab. Diese Informationen sind größtenteils bereits explizit im WDI und zum Teil in der ZINC Datenbank enthalten. Das Regelwerk zur Erzeugung von Fragmenten soll jedoch unabhängig von dieser zusätzlichen Information spezifiziert und verwendet werden, weshalb positiv geladene chemische Umgebungen nicht explizit als synthetisch relevante Anknüpfungstellen modelliert werden. Unabhängig davon können quaternäre Stickstoffe durchaus in den resultierenden Fragmenten auftreten, sofern sie Bestandteil der Datengrundlage sind.

Des Weiteren wurde eine wesentlich umfangreichere Behandlung von Ringmotiven realisiert. Während RECAP ausschließlich biarylische Systeme fragmentiert, werden bei BRICS alle Kombinationen unterschiedlicher Ringmotive gespalten und zusätzlich explizit aliphatische Ringkohlenstoffe als Anknüpfungspunkte verwendet (L_{13} und L_{15} in Abb. 6.3). Außerdem wird zwischen aktivierten und inaktivierten heterozyklischen Systemen unterschieden, womit in diesem Zusammenhang Ringe gemeint sind, die ein Heteroatom in alpha-Position zum Ringkohlenstoff haben, an den das Link-Atom gebunden ist (L_{13}/L_{14} bzw. L_{15}/L_{16} in Abb. 6.3).

Unter Berücksichtigung der so modellierten vielfältigen bioisosteren Ersetzungsmöglichkeiten sowie der Erweiterung und Generalisierung der Verknüpfungsregeln für Fragmente wird eine Vielzahl zusätzlicher chemischer Motive zugänglich gemacht (siehe dazu Kap. 6.2.3). Der entsprechende Fragmentraum und die BRICS-Fragment-Prototypen sind in Abbildung 6.3 dargestellt. Dabei ist aus Gründen der Anschaulichkeit jeweils nur die direkte chemische Umgebung der Link-Atome abgebildet.

Das BRICS-Modell wurde erst gegen Ende der vorliegenden Arbeit fertiggestellt. Aus diesem Grund sind die Ergebnisse der Analysen für die fokussierten Fragmenträume, sowie die meisten der FlexNovo-Rechnungen, mit dem FTree-Fragmentraum durchgeführt worden.

6.1.4 Target-spezifische Fragmente

Für die Erstellung fokussierter und wohl definierter Fragment-Zusammenstellungen wurden die Inhibitoren der 40 Target-Klassen aus dem oben erwähnten DUD-Datensatz verwendet. Aus Gründen der Vergleichbarkeit kam die oben beschriebene verallgemeinerte Variante des RECAP-Modells für die Fragmentierung zum Einsatz. Gemäß der dargelegten Shredding-Prozedur wurden aus den Inhibitor-Zusammenstellungen separate, target-spezifische Fragmenträume erstellt. Dabei wurden keinerlei zusätzliche Filterkriterien angewandt, um eine Verzerrung der erstellten Fragment-Datensätze und der damit erzielten Ergebnisse zu vermeiden.

Nicht alle verwendeten Inhibitoren konnten jedoch in Fragmente zerlegt werden, da das Shredding-Regelwerk zum einen sehr spezifisch ist, und sich die Moleküle zum anderen hinsichtlich ihrer strukturellen Eigenschaften stark unterscheiden. Dementsprechend wurden sehr kleine Substanzen und Moleküle mit keiner oder nur sehr geringer Dekoration aufgrund fehlender Schnittkanten nicht fragmentiert.

Die einzelnen Inhibitor-Sets unterscheiden sich deutlich in ihrer Größe, was sich auch auf die Menge der daraus erzeugten Fragmente auswirkt. Die Ergebnisse des Shreadings sowie die grundlegenden Eigenschaften der daraus resultierenden Fragmenträume sind in Tabelle 6.4 zusammengefasst. Im Durchschnitt haben 75 Prozent aller Fragmente einen Link, 22 Prozent haben zwei Links, 2,5 Prozent haben drei Links und 0,5 Prozent vier Links. Fragmente mit mehr als vier Links wurden nicht erzeugt. Die Eigenschaften der abgeleiteten target-spezifischen Fragmenträume sowie die damit erzielten Ergebnisse werden in Kapitel 6.2.2 behandelt.

6.1.5 Fragmente aus Wirkstoff- und Molekül-Katalogen

Für die Erzeugung generischer Fragment-Zusammenstellungen wurden die beiden vorgestellten Regelwerke auf die aufbereiteten WDI und ZINC Datensätze angewandt. Die Shredding-Statistiken für den WDI sind in Tabelle 6.2, die für ZINC in Tabelle 6.3, zusammengefasst. Bei Betrachtung der Werte fällt zunächst auf, dass der Prozentsatz gespaltener Moleküle beim WDI generell deutlich geringer ist als bei der ZINC Datenbank. Die Ursache hierfür liegt in den darin enthaltenen Strukturen. Der WDI besteht im Gegensatz zu der ZINC Datenbank aus einer großen Menge an Naturstoffen und dar-

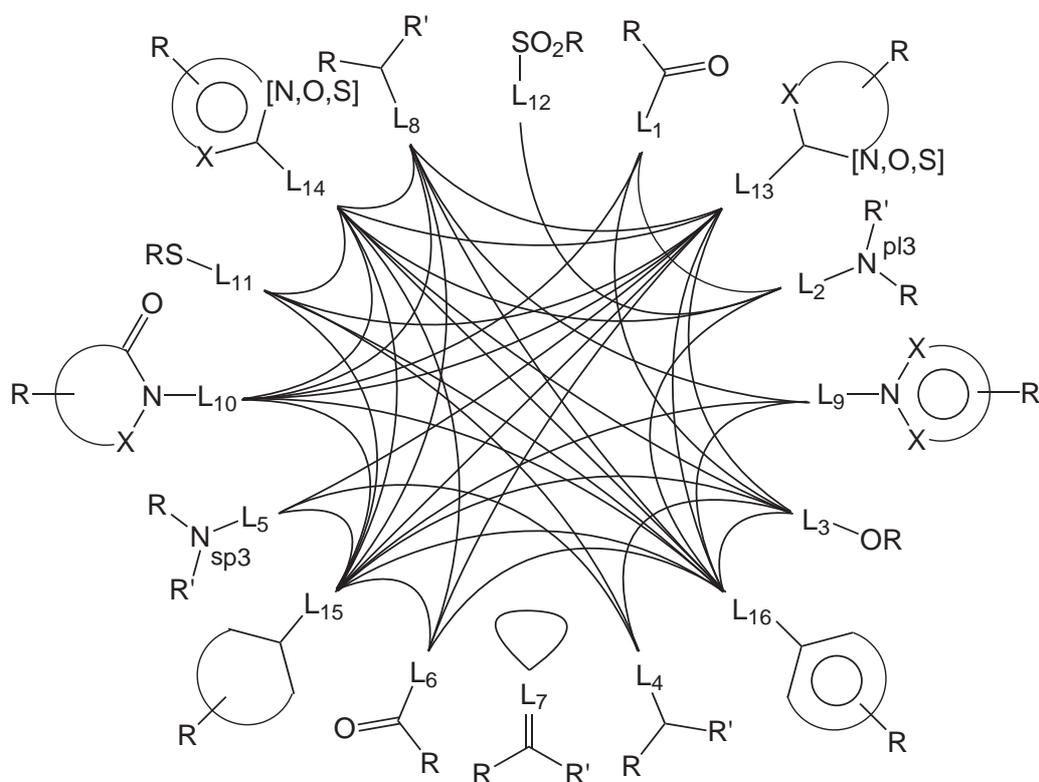


Abbildung 6.3: Die BRICS-Fragment-Prototypen. Die Diversität der Fragmente wird durch unterschiedliche Substitutionen mithilfe der R-Gruppen modelliert. Diese können weitere Link-Atome enthalten bzw. nur aus einem Link-Atom bestehen. Bei den R'-Gruppen kann es sich um eine weitere (auch chemisch unterschiedliche) R-Gruppen handeln, aber auch nur um ein Wasserstoff-Atom. Die Kreisausschnitte repräsentieren Ringe unterschiedlicher Größe, die auch anneliert bzw. verbrückt sein können. Der Atom-Platzhalter 'X' steht für eines der chemischen Elemente Kohlenstoff, Stickstoff, Sauerstoff oder Schwefel. Der Zusatz sp³ bzw. pl³ am Stickstoffatom von L₂ bzw. L₅ bezieht sich auf die jeweilige Hybridisierung. Die Fragment-Kompatibilität wird durch verbindende Linien zwischen den Link-Atomen angedeutet. Eine vollständige und detaillierte Auflistung der Verknüpfungsregeln ist im Anhang C enthalten.

Tabelle 6.2: Shredding-Ergebnisse des World Drug Index (WDI)

	BRICS		RECAP	
Moleküle ohne Schnittkanten	20715	(35%)	28362	(47%)
Moleküle mit Schnittkanten	39163	(65%)	31516	(53%)
Anzahl eindeutiger Fragmente	18291	(100%)	15650	(100%)
Fragmente mit 1 Link	13256	(72%)	12236	(78%)
Fragmente mit 2 Links	4344	(24%)	3079	(19%)
Fragmente mit 3 Links	591	(03%)	290	(02%)
Fragmente mit 4 Links	83	(<01%)	36	(<01%)
Fragmente mit 5 Links	12	(<01%)	7	(<01%)
Fragmente mit 6 Links	5	(<01%)	1	(<01%)
Fragmente mit 7 Links	-	-	1	(<01%)

Tabelle 6.3: Shredding-Ergebnisse des ZINC 'drug-like' Datensatzes

	BRICS		RECAP	
Moleküle ohne Schnittkanten	214793	(11%)	415762	(21%)
Moleküle mit Schnittkanten	1806262	(89%)	1605293	(79%)
Anzahl eindeutiger Fragmente	93309	(100%)	101889	(100%)
Fragmente mit 1 Link	76196	(81%)	90023	(88%)
Fragmente mit 2 Links	15964	(17%)	11395	(11%)
Fragmente mit 3 Links	1118	(01%)	453	(<01%)
Fragmente mit 4 Links	30	(<01%)	17	(<01%)
Fragmente mit 5 Links	1	(<01%)	1	(<01%)

aus abgeleiteter oder strukturell verwandter Moleküle. Die entsprechenden chemischen Grundgerüste von, z.B. Alkaloiden, Polyketiden, Terpenen und Polyphenolen weisen in der Regel keine azyklischen Schnittkanten auf, sodass die meisten dieser Strukturen im Rahmen der verwendeten Shredding-Prozedur nicht fragmentiert werden können. Teilweise verfügen diese Grundgerüste jedoch über größere Substituenten, die dann entsprechend abgespalten werden. Die Grundgerüste selbst werden aber dennoch weitestgehend verworfen, da sie verhältnismäßig groß sind und oft aus vielen Ringen bestehen bzw. eine große Anzahl an Stereozentren aufweisen.

Anhand der Shredding-Ergebnisse wird deutlich, dass mit BRICS ein wesentlich höherer Prozentsatz an Molekülen fragmentiert werden kann. Für den WDI liegt die Steigerung bei zwölf Prozent und für die ZINC Datenbank bei zehn Prozent. Zusätzlich werden deutlich weniger terminale Fragmente erzeugt, d.h., Fragmente mit nur einem Link. Umgekehrt wird somit sowohl prozentual als auch absolut eine wesentlich größere Anzahl von Fragmenten mit zwei und mehr Link-Atomen erzeugt. Dies ist ein wich-

tiger Gesichtspunkt, denn dadurch erhöht sich auch die Menge potenziell zugänglicher Topologien bei der Fragmentverknüpfung, was sich zusammen mit den bereits angesprochenen strukturellen Gesichtspunkten in der Menge prinzipiell zugänglicher Moleküle widerspiegelt (vgl. Tab. 6.6).

Die so erzeugten Fragmente wurden insbesondere hinsichtlich ihrer Größe charakterisiert. Die Verteilungen des Molekulargewichts sind in Abbildung 6.4 enthalten. Anhand der Werte wird deutlich, dass es sich bei den verwendeten Fragmenten prinzipiell um kleine Moleküle handelt, die durchschnittlich ein Molekulargewicht von 200 bis 250 aufweisen. Sehr große Fragmente werden in einem direkt in die Shredding-Prozedur integrierten Filterschritt verworfen, sodass wesentlich größere Werte als die hier ersichtlichen nicht zu erwarten sind. Umgekehrt werden sehr kleine terminale Fragmente während der Zerlegung von Molekülen erst gar nicht erzeugt. Die Tatsache, dass große kondensierte Ringsysteme sowie Makrozyklen nicht gespalten sondern direkt verworfen werden, spiegelt sich ebenfalls in der Verteilung wider. So entspricht das durchschnittliche Molekulargewicht einem Fragment mit bis zu drei Ringen.

Die WDI-Fragmente sind durchschnittlich wesentlich kleiner als die ZINC-Fragmente. Die Ursachen hierfür liegen in den enthaltenen Strukturen und der gewählten Shredding-Strategie. Die Dekoration von Molekülen mit zusätzlichen, auch größeren strukturellen Motiven ist im WDI wesentlich häufiger der Fall als in der ZINC Datenbank. Solche Gruppen werden oft zur Steigerung der Aktivität bzw. zur Verbesserung des ADMET-Profiles einer Verbindung eingeführt. Diese Substituenten werden bei der Fragmentierung eines solchen Moleküls entsprechend abgespalten, sofern sie eine gewisse Mindestgröße haben. Die Menge an verbrückenden Fragmenten ist beim WDI ebenfalls größer als bei der ZINC Datenbank, was ebenfalls in den Shredding-Statistiken zu erkennen ist (siehe hierzu Tab. 6.2 und Tab. 6.3).

Abbildung 6.4 zeigt außerdem, dass die Verteilung der mit BRICS erzeugten Fragmente des WDI (dunkelgrün) im Gegensatz zu den mit RECAP erzeugten (hellgrün) leicht hin zu kleineren Molekulargewichten verschoben ist. Bei den aus der ZINC Datenbank erstellten Fragmenten ist die Verschiebung der BRICS-Ergebnisse (dunkelblau) im Gegensatz zu den RECAP-Ergebnissen (hellblau) noch etwas deutlicher. Allerdings lassen sich aus den Verteilungen keine statistischen Kenngrößen ableiten, mit deren Hilfe eine nennenswerte Größenverschiebung zwischen den beiden Regelwerken beziffert werden könnte.

Neben dem Molekulargewicht wurden aus den erzeugten Fragmenten die Verteilungen der einzelnen Link-Typen für BRICS (linke Grafik in Abb. 6.5) bzw. RECAP (rechte Grafik in Abb. 6.5) ermittelt. Die Link-Typen sind allerdings nicht direkt miteinander vergleichbar, da sie aus unterschiedlichen Regelwerken hervorgegangen sind. Selbst die relativen Anteile derselben, in beiden Regelsätzen verwendeten chemischen Umgebungen, z.B. die des Pyrrol-ähnlichen und Lactam-ähnlichen Stickstoffs (L_9/L_{10} in BRICS bzw. L_8/L_9 in RECAP) können nicht direkt miteinander verglichen werden, da sie bereits während des Shreddings in unterschiedlichen Kombinationen mit anderen chemischen Motiven Berücksichtigung finden.

Dennoch lassen sich zwei Gemeinsamkeiten erkennen: Sowohl der Ester- bzw. Ether-Sauerstoff (L_3 in BRICS bzw. L_2 in RECAP) als auch der aromatische Kohlenstoff

6 Erzeugung und Charakterisierung von Fragmenträumen

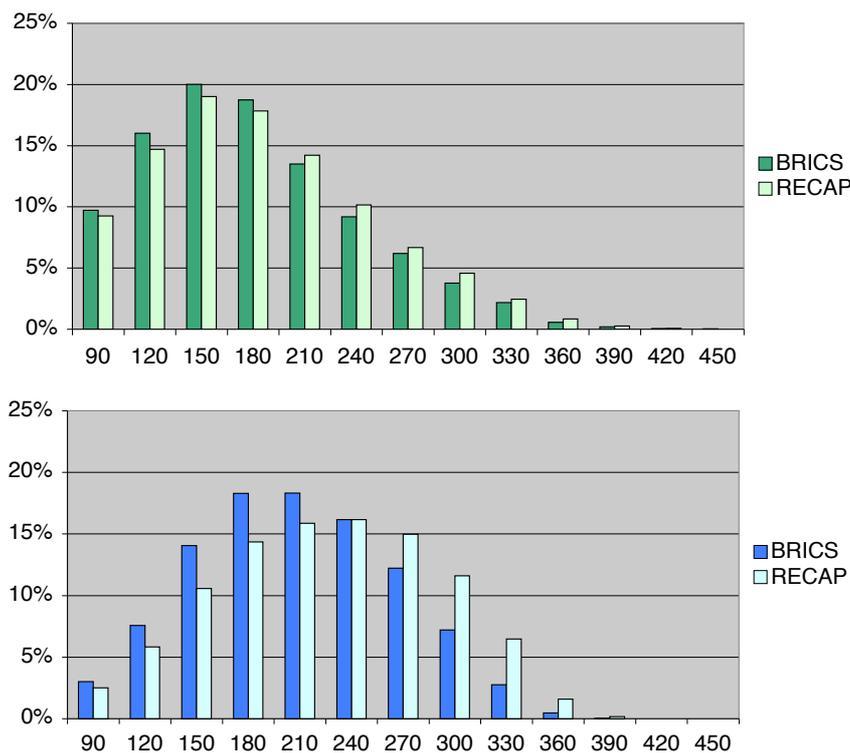


Abbildung 6.4: Verteilung des Molekulargewichts für die Fragmente des WDI (oben, grün) und der ZINC Datenbank (unten, blau).

(L_{14} in BRICS bzw. L_{10} in RECAP) sind in Fragmenten aus der ZINC Datenbank prozentual mindestens doppelt so häufig vertreten wie in den WDI-Fragmenten. Mithilfe des verfeinerten BRICS-Modells kann diese Beobachtung weiter präzisiert werden. Der höhere Anteil aromatischer Systeme geht somit ausschließlich auf in alpha-Position zum Link-Kohlenstoff aktivierte aromatische heterozyklische Ringsysteme zurück (L_{14}). Im Gegensatz dazu sind sowohl die nicht aktivierten aliphatischen als auch die aromatischen Systeme (L_{15} bzw. L_{16}) stärker in den WDI-Fragmenten vertreten. Dies spiegelt teilweise die Zusammenstellung kommerziell verfügbarer Kataloge wider, in denen vor allem für aromatische Ringsysteme die meisten der synthetisch in großem Maßstab herstellbaren Substitutionsmuster enthalten sind, wohingegen im WDI eine kleinere Menge immer wiederkehrender Substituenten vorherrscht.

Aus den Verteilungen zeigt sich weiterhin, dass die absolute Häufigkeit bestimmter Link-Typen durch stärkere Diversifizierung in BRICS deutlich reduziert werden konnte. Dies ermöglicht eine präzisere und feinere Anpassung des Regelwerks für unterschiedliche Anwendungsszenarien. Jedoch lässt die Häufigkeit eines bestimmten Link-Typs *per se* keinen Rückschluss auf dessen Bedeutung zu, ebenso wenig ist dies für die reine Größe eines Fragments möglich. So kommen z.B. den von acylierten Ringkohlenstoffen abgeleiteten Carbonyl-Sauerstoffen (L_6), den Thioethern (L_{11}) sowie den aus Sulfonamiden

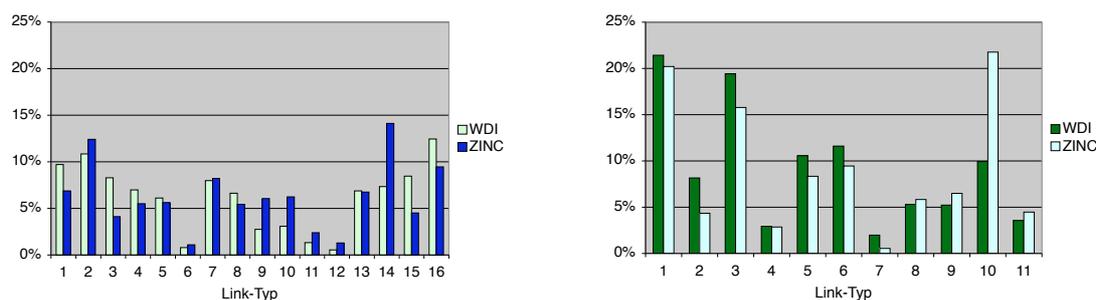


Abbildung 6.5: Verteilung der Link-Typen für BRICS (links) und RECAP (rechts) für die beiden Datensätze WDI (grün) und ZINC (blau). Die einzelnen Link-Typen sind nicht direkt miteinander vergleichbar. Dies trifft auch auf die prozentualen Anteile identischer chemischer Umgebungen zu, da sie aus unterschiedlichen Fragmentierungsregelwerken stammen.

abgeleiteten Sulfon-Fragmenten (L_{12}) trotz des geringen prozentualen Anteils wichtige funktionalisierende wie auch topologische Funktionen zu. Dies gilt in gleicher Weise auch für verbrückende Ether- (L_3), Alkyl- (L_8) und Amin-Gruppen (L_2/L_5). Auf diese Aspekte wird im Laufe der Diskussion der mit BRICS erzielten Ergebnisse sowie im Zusammenhang des Vergleichs mit RECAP in Kapitel 6.2.3 näher eingegangen.

6.1.6 Die BRICS-Fragment-Datensätze

Die erstellten Fragment-Datensätze aus dem WDI und ZINC haben bezüglich ihrer direkten Verwendbarkeit mehrere Nachteile. Zum einen ist der ZINC Fragment-Datensatz mit ungefähr 100000 Fragmenten zu groß, um mit ihm effektiv und in akzeptabler Zeit Berechnungen durchführen zu können. Zum anderen können die WDI-Fragmente aufgrund der verwendeten Datenquelle nicht ohne Weiteres frei verwendet und verteilt werden.

In einem ersten Schritt zur Erstellung eines allgemeinen, leistungsfähigen und frei zugänglichen Fragment-Datensatzes von akzeptabler Größe wurden deshalb zunächst die identischen Fragmente aus den WDI und ZINC Datensätzen mithilfe kanonischer SMILES identifiziert. Für BRICS ergaben sich dabei 4800 und für RECAP 4125 überlappende Fragmente, aus denen zwei neue separate Datensätze erstellt wurden: Der BRICS_4k und RECAP_4k Datensatz, wobei letzterer im Folgenden ausschließlich als Referenz verwendet wird.

In einem zweiten Schritt wurde der BRICS_4k Datensatz mit Fragmenten aus der ZINC Datenbank angereichert. Dazu wurde zunächst die Ähnlichkeit der ZINC-Fragmente zu den WDI-Fragmenten bestimmt, indem alle paarweisen Tanimoto-Abstände [Tanimoto, 1957] der jeweiligen öffentlichen MDL Fingerprints [Durant *et al.*, 2002] berechnet wurden. Um den Einfluss der Fragmentgröße der Verbindungen auf den errechneten Ähnlichkeitswert zu minimieren, wurden nur Fragmente miteinander verglichen, die innerhalb desselben Molekulargewichtsbereichs von 30 Dalton gemäß der Einteil-

lung in Abbildung 6.4 liegen. Daraufhin wurden alle ZINC-Fragmente ausgewählt, deren Ähnlichkeitswert mindestens 0,9 bzw. 0,8 zu einem beliebigen WDI-Fragment innerhalb desselben Bereichs beträgt. Dadurch kann gewährleistet werden, dass eine hinreichende Ähnlichkeit zu den Strukturen der WDI-Fragmente vorhanden ist [Flower, 1998]. Aus den Analysen resultieren zwei neue Fragment-Datensätze: BRICS_9k mit 9344 Fragmenten und BRICS_20k mit 22343 Fragmenten.

Die vier erstellten Fragment-Datensätze und die Eigenschaften der daraus resultierenden Fragmenträume sind in Tabelle 6.6) zusammengestellt. Für die Zusammensetzung neuer chemischer Motive aus den jeweiligen Fragmenten werden generell dieselben Regeln herangezogen, die auch zur retrosynthetischen Zerlegung der ursprünglichen Moleküle verwendet worden sind. Demnach beziehen sich auch die Abschätzungen der Größe der Fragmenträume sowie die durchgeführten Performance-Untersuchungen auf dasselbe Regelwerk (siehe Kap.6.2.3).

6.2 Charakterisierung von Fragmenträumen

Nach der Charakterisierung der erstellten Fragment-Datensätze werden im Folgenden nun die daraus resultierenden Fragmenträume betrachtet. Dabei wird zunächst auf grundlegende Eigenschaften von Fragmenträumen eingegangen. Im Anschluss werden die neu erstellten Fragmenträume im Detail analysiert und das BRICS-Modell hinsichtlich seiner Vorhersagekraft mit bestehenden Konzepten verglichen.

6.2.1 Grundlegende Eigenschaften von Fragmenträumen

Eine triviale Eigenschaft eines Fragmentraums ist seine theoretische Größe, d.h., die Anzahl der Moleküle die daraus erzeugt werden können. Typischerweise ist die Größe eines Fragmentraums unendlich, d.h., es kann eine unendliche Menge an Molekülen aus den Fragmenten gemäß der Kompatibilität der Link-Atome erzeugt werden. Aus offensichtlichen Gründen müssen dafür multiple Kopien desselben Fragments verwendet werden, wodurch die Größe der erzeugten Moleküle ebenfalls unendlich anwächst. Ein realistischeres Maß für die Größe eines Fragmentraums ist daher die Menge an Molekülen, die mit einer begrenzten Anzahl an Fragmenten erzeugt werden kann.

Weiterhin ist es für einen Fragmentraum prinzipiell von Bedeutung, ob theoretisch jede Kombination von Fragmenten in demselben Molekül vorkommen kann. Ein Fragmentraum mit dieser Eigenschaft besteht nur aus einer Komponente und wird als *verbunden* bezeichnet, was für größere Räume typischerweise der Fall ist. Hingegen bestehen Fragmenträume mit einer relativ geringen Zahl an Fragmenten mitunter aus zwei oder mehr Komponenten, die jeweils hinsichtlich ihrer Größe charakterisiert werden können.

6.2.2 Target-spezifische Fragmenträume

Anhand der target-spezifischen Fragmenträume sollen die oben beschriebenen grundlegenden Eigenschaften näher untersucht werden. Die Ergebnisse in Tabelle 6.4 zeigen, dass die kleineren Fragmenträume oft aus mehreren Komponenten bestehen, die selbst

theoretisch endlich oder unendlich groß sein können. Dies ist jedoch *per se* keine Einschränkung, da die Anzahl potenzieller Moleküle immer noch sehr groß sein kann (siehe Tab. 6.5). Sind jedoch alle Komponenten selbst endlich in ihrer theoretischen Größe, oder ist die Anzahl der enthaltenen Fragmente sehr gering, so handelt es sich für praktische Anwendungen nicht mehr um einen Fragmentraum im ursprünglichen Sinne. Die entsprechenden Target-Klassen wurden deshalb in den anschließenden Analysen nicht weiter berücksichtigt.

6.2.2.1 Enumeration innerhalb definierter physikochemischer Bereiche

Wie bereits oben beschrieben, ist die Menge an Molekülen, die mit einer begrenzten Zahl an Fragmenten erzeugt werden kann, ein realistischeres Maß für die Größe eines Fragmentraums. Da während der Shredding-Prozedur sichergestellt wurde, dass die Fragmente selbst eine bestimmte Mindestgröße haben, wurde die Anzahl der in diesem Zusammenhang zu verwendenden Fragmente auf bis zu fünf beschränkt (vgl. [Rarey, 2001]). Dadurch wird auch die Anzahl der theoretisch zugänglichen Moleküle endlich und kann für den jeweiligen Fragmentraum anhand der Fragmente und der Kompatibilität der Link-Atome berechnet werden.

Neben der so bestimmten theoretischen Anzahl ist die tatsächliche Menge an Molekülen, die eine Reihe wichtiger physikochemischer Eigenschaften erfüllen, im Rahmen der Leitstruktursuche von wesentlich größerem Interesse. Um diese Zahl zu ermitteln, wurden zunächst Wertebereiche typischer physikochemischer Eigenschaften aus den ursprünglichen Inhibitoren abgeleitet (Tab. C.3). In diesem Zusammenhang wurden Minimal- und Maximalwerte für das Molekulargewicht, die Anzahl rotierbarer Bindungen, Ringe, Wasserstoffbrücken-Donoren und -Akzeptoren sowie für den berechneten logP Wert bestimmt. Die entsprechenden Werte stellen zusammen mit den jeweiligen Fragmenträumen die Eingabedaten für ein Software-Werkzeug basierend auf einer Enumerationsprozedur dar, mit deren Hilfe alle Moleküle aus dem Fragmentraum generiert werden, deren physikochemische Eigenschaften in den spezifizierten Bereichen liegen [Pärn *et al.*, 2007].

Die Ergebnisse der Enumeration sowie die errechneten Werte für die Obergrenze der Anzahl zugänglicher Moleküle sind in Tabelle 6.5 zusammengestellt. Als dritte Kenngröße wurde der Quotient aus der tatsächlich erzeugten Menge an Molekülen und der theoretisch zugänglichen Anzahl gebildet.

Bei Betrachtung der Ergebnisse fällt zunächst auf, dass die Enumeration nicht für alle Target-Klassen erfolgreich abgeschlossen werden konnten. So mussten die Berechnungen bei einigen Target-Klassen aus technischen Gründen abgebrochen werden, da die Anzahl der erzeugten Moleküle eine Million weit überschritten hätte und somit auf einem Workstation-Computer mit 2 GB Hauptspeicher nicht vollständig durchgeführt werden konnten. Aus Tabelle 6.5 wird ersichtlich, dass dies für Fragmenträume der Fall war, deren Obergrenze für die Menge der zugänglichen Moleküle jenseits von 30 Millionen liegt.

Die Eigenschaften eines Fragmentraums sind somit ein Hinweis darauf, aber nicht ausreichend, um zu entscheiden, ob dieser innerhalb definierter physikochemischer Bereiche enumeriert werden kann oder nicht. Zum Beispiel konnten die Räume von HMGa und

Tabelle 6.4: Shredding-Ergebnisse und Eigenschaften der Fragmenträume

Target-Klasse	Anzahl Inhibitoren	Anzahl Fragmente	Größe des Raums	Anzahl Komponenten
ACE	49	45	∞	2
AChE	107	100	∞	2
ADA	39	25	endlich	12
ALR2	26	17	∞	4
AmpC	21	26	endlich	12
AR	79	15	∞	4
CDK2	72	62	∞	4
COMT	11	7	endlich	12
COX-1	25	25	∞	5
COX-2	426	271	∞	1
DHFR	410	147	∞	3
EGFr	475	125	∞	2
ER _{Agonisten}	67	29	∞	4
ER _{Antagonisten}	39	51	∞	3
FGFr1	120	106	∞	2
FXa	146	189	∞	1
GART	40	23	∞	3
GPB	52	24	∞	2
GR	78	57	∞	5
HIVPR	62	70	∞	2
HIVRT	43	40	∞	2
HMGA	35	31	∞	1
HSP90	37	26	∞	4
InhA	86	92	∞	1
MR	15	3	endlich	12
NA	49	26	∞	5
P38	454	194	∞	2
PARP	35	20	∞	5
PDE5	88	93	∞	2
PDGFr _b	170	141	∞	2
PNP	50	27	endlich	5
PPAR _g	85	90	∞	4
PR	27	26	endlich	9
RXR _a	20	4	∞	5
SAHH	33	27	∞	5
SRC	159	122	∞	2
Thrombin	72	87	∞	2
TK	22	23	∞	5
Trypsin	49	58	∞	2
VEGFr2	88	118	∞	1
Alle Klassen	3961	1942	∞	1

Tabelle 6.5: Ergebnisse der Enumeration

Target-Klasse	Anzahl enum. Moleküle	Theoret. Anzahl Moleküle	proz. Anteil
ALR2	21	1,06e+04	0,2
AR	37	9,40e+01	39,4
NA	56	9,71e+03	0,6
PARP	102	1,06e+04	1,0
SAHH	128	4,54e+03	2,8
GART	188	3,70e+03	5,1
COX-1	298	9,60e+03	3,1
TK	416	2,93e+04	1,4
GR	752	1,74e+04	4,3
GPB	940	5,96e+03	15,8
ER _{Agonisten}	1387	1,61e+06	0,1
HIVRT	2666	4,68e+04	5,7
ACE	5604	2,38e+06	0,2
HSP90	5839	2,28e+06	0,3
CDK2	7610	1,60e+05	4,8
Trypsin	23716	3,07e+05	7,7
HMGA	27264	1,14e+06	2,4
DHFR	36691	9,82e+05	3,7
FGFr1	84608	2,66e+06	3,2
SRC	105529	3,47e+06	3,0
AChE	136025	8,98e+06	1,5
PDGFr _b	205119	7,46e+06	2,7
EGFr	396085	1,29e+07	3,1
InhA	544705	3,45e+06	15,8
Thrombin	779213	5,23e+06	14,9
COX-2	≫ 1000000	8,99e+11	-
ER _{Antagonisten}	≫ 1000000	3,44e+07	-
FXa	≫ 1000000	1,61e+09	-
HIVPR	≫ 1000000	1,07e+09	-
P38	≫ 1000000	5,78e+10	-
PDE5	≫ 1000000	9,15e+07	-
PPAR _g	≫ 1000000	1,86e+08	-
VEGFr2	≫ 1000000	9,30e+07	-
Alle Klassen	≫ 1000000	5,38e+14	-

InhA erfolgreich enumeriert werden, obwohl sie verbunden und theoretisch unendlich groß sind. Im Gegensatz dazu konnten die Berechnungen für HIVPR, P38 und PPAR γ nicht abgeschlossen werden, obwohl die Räume aus mehreren Komponenten bestehen. Des Weiteren konnte beispielsweise der aus 147 Fragmenten bestehende DHFR-Raum enumeriert werden, wohingegen das für den aus nur 51 Fragmenten bestehende ER-Antagonisten-Raum nicht möglich war.

Generell lässt sich feststellen, dass die Menge der tatsächlich erzeugten Moleküle in etwa mit der prinzipiellen Größe des Fragmentraums korreliert. Allerdings variiert das Verhältnis aus der Menge der tatsächlich erzeugten Moleküle und der Zahl der theoretisch zugänglichen Moleküle sehr stark. Die Spanne reicht von Werten im unteren Prozentbereich im Fall der ER-Agonisten, ACE und NA bis hin zu Werten im zweistelligen Prozentbereich bei AR, GPB, InhA und Thrombin.

Die Ergebnisse lassen sich in drei Klassen einteilen. In die erste Klasse fallen alle Targets, bei denen die Enumeration bis zu wenige Hundert Moleküle ergibt und innerhalb weniger Sekunden durchgeführt werden kann, wozu beispielsweise ACE, ALR2, CDK2, ER-Agonisten, HIVRT und NA gehören. Die zweite Klasse wird von den Targets gebildet, die mehrere Zehntausend bis zu einer halben Million enumerierter Moleküle ergeben, wozu in Tabelle 6.5 der Bereich von Trypsin bis einschließlich EGFr zählt. Die Enumeration ist dabei innerhalb weniger Stunden durchführbar. Zu der dritten und letzten Kategorie zählen InhA und Thrombin, bei denen die Berechnung jeweils mehrere Tage in Anspruch genommen hat sowie die Targets, bei denen die Enumeration aus technischen Gründen nicht abgeschlossen werden konnte.

Für einen Großteil der verwendeten Target-Klassen war es demnach möglich, den kompletten zugrunde liegenden chemischen Fragmentraum innerhalb der physikochemischen Schranken zu enumerieren. Dabei hat sich herausgestellt, dass oftmals nur ein kleiner Prozentsatz der theoretisch möglichen Moleküle tatsächlich erzeugt werden musste. Dennoch lässt sich ein Fragmentraum nicht allein anhand einfacher Eigenschaften wie der Anzahl an Fragmenten, der prinzipiellen Größe oder der Anzahl an Komponenten charakterisieren. Die jeweiligen zugrunde liegenden Fragmente müssen somit einen entscheidenden Einfluss haben. Um diese Hypothese näher zu untersuchen, werden im Folgenden die im Rahmen der Enumeration generierten Moleküle analysiert.

6.2.2.2 Analyse der enumerierten Moleküle

Um die Qualität der erzeugten Moleküle zu quantifizieren, wurden diese hinsichtlich ihrer Ähnlichkeit zu den Inhibitoren untersucht, die ursprünglich zur Erzeugung der zugrunde liegenden Fragmente verwendet wurden. Dafür wurden alle paarweisen Ähnlichkeiten bezüglich des FTree-Deskriptors berechnet [Rarey & Dixon, 1998]. Das FTree-Ähnlichkeitsmaß bewertet insbesondere die relative Anordnung funktioneller Gruppen, gemeinsame Substrukturen spielen dabei nur eine untergeordnete Rolle. Aufgrund dieser Eigenschaft eignet sich der FTree-Deskriptor insbesondere für den Vergleich von Molekülen, die aus demselben Fragmentraum erstellt worden sind.

Für alle enumerierten Moleküle einer Target-Klasse wurde jeweils der Inhibitor bestimmt, der gemäß des FTree-Deskriptors den größten Ähnlichkeitswert aufweist. Um die

6.2 Charakterisierung von Fragmenträumen

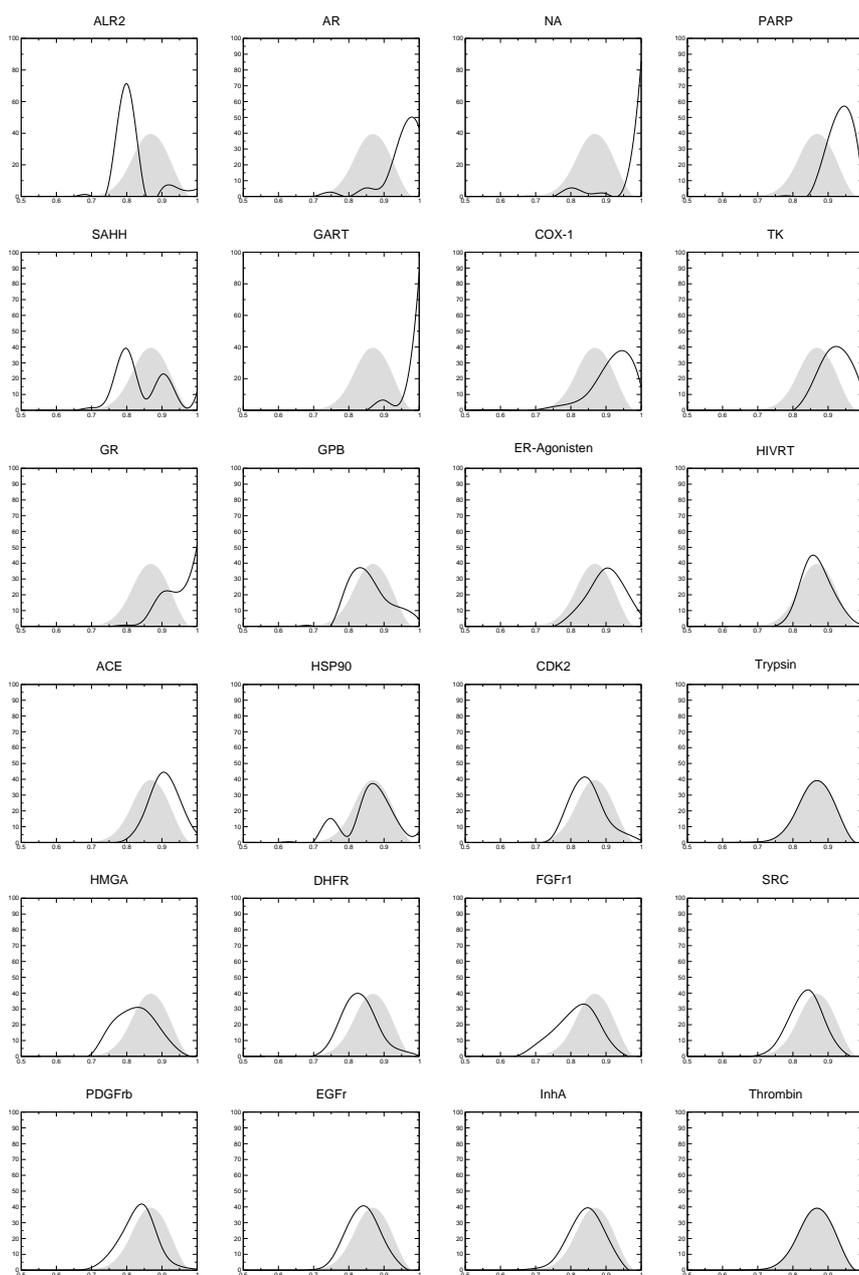


Abbildung 6.6: Verteilungen der FTree-Ähnlichkeitswerte für die enumerierten Moleküle aus den verschiedenen Target-Klassen. Die Histogramme sind gemäß Tabelle 6.5 sortiert, normiert und als geglättete Liniendiagramme dargestellt. Sie zeigen die Verteilungen der maximalen Ähnlichkeitswerte jedes enumerierten Moleküle zu einem beliebigen Inhibitor derselben Target-Klasse. Die Referenz-Verteilung ist grau eingefärbt.

Ergebnisse besser interpretieren zu können, wurde eine Referenz-Ähnlichkeitsverteilung erzeugt. Dazu wurde der Datensatz von Stahl und Rarey herangezogen, der ursprünglich für die Analyse von Scoring-Funktionen im Zusammenhang mit virtuellem Screening erstellt worden ist [Stahl & Rarey, 2001]. Er besteht aus 7528 zufällig aus dem Word Drug Index ausgewählten Molekülen und 452 Inhibitoren bestimmter Target-Klassen. Im Rahmen dieser Arbeit konnten davon aus Lizenzgründen nur 398 Inhibitoren verwendet werden. Alle Strukturen der verwendeten Target-Klassen wurden dabei aus der Zufalls-Zusammenstellung entfernt.

Die resultierenden Verteilungen der Ähnlichkeitswerte sind in Abbildung 6.6 enthalten. Aus Gründen der Anschaulichkeit sind die Histogramme der enumerierten Moleküle als geglättete Liniendiagramme dargestellt und die Referenz-Verteilung grau eingefärbt.

Bei Betrachtung der Verteilungen lässt sich zunächst feststellen, dass die durchschnittlichen Werte alle im Bereich hoher bis sehr hoher FTree-Ähnlichkeit liegen. Demnach haben alle enumerierten Moleküle eine hohe bis sehr hohe topologische Ähnlichkeit zu einem entsprechenden Inhibitor. In Analogie zur Charakterisierung der Resultate der Enumeration lassen sich auch die Ergebnisse der Ähnlichkeitsanalyse in drei Klassen einteilen. Die erste Kategorie besteht aus den Target-Klassen, bei denen die Enumeration bis zu einigen Hundert Molekülen ergeben hat. Diese Menge entspricht in etwa der Menge an Inhibitoren, die zur Erzeugung der jeweiligen Fragmenträume verwendet worden sind. Die enumerierten Moleküle sind den Inhibitoren topologisch sehr ähnlich, was sich auch in den jeweiligen Verteilungen widerspiegelt, sodass diese im Vergleich zu der Referenz-Verteilung deutlich zu höheren Ähnlichkeitswerten verschoben sind. Es gibt jedoch Ausnahmen, wie zum Beispiel bei ALR2, bei der der mittlere Ähnlichkeitswert bei 0,80 liegt, sowie SAHH und COX1, die beide eine deutlich verbreiterte Verteilung aufweisen. Die zweite Klasse besteht aus den Targets, bei denen die Enumeration bis zu einigen Tausend Molekülen ergeben hat. Dort sind die Verteilungen in der Regel immer noch zu höheren Ähnlichkeitswerten verschoben. Der Effekt ist allerdings wesentlich schwächer ausgeprägt je größer die Menge erzeugter Moleküle wird. Zu der letzten Kategorie zählen die Klassen mit mehreren Hunderttausend Molekülen. Die zugehörigen Verteilungen ähneln der Referenz-Verteilung sehr stark und heben sich zum Teil nur wenig davon ab.

Die beschriebenen Charakteristika der Verteilungen erlauben zunächst keine Aussage über die Qualität der Ergebnisse für die einzelnen Target-Klassen, vielmehr spiegeln sie die Eigenschaften des jeweiligen Fragmentraums wider.

Die verwendete Datenbasis, d.h., die jeweilige Zusammenstellung der Inhibitoren bzw. die daraus abgeleiteten Fragmente spielen dabei eine entscheidende Rolle. Haben die Verbindungen innerhalb der Menge nur eine geringe strukturelle Diversität, so sind auch die erzeugten Moleküle sehr ähnlich zueinander. Umgekehrt ist bei großer struktureller Diversität der Inhibitoren auch eine geringere paarweise Ähnlichkeit der erzeugten Moleküle zu erwarten. Dies lässt sich allerdings nicht direkt an der Verschiebung der Verteilung der FTree-Ähnlichkeitswerte bezüglich der Referenzverteilung erkennen, da darin die Ähnlichkeit zu den zugrunde liegenden Inhibitoren enthalten ist. Die Verschiebung der Verteilungen zu höheren bzw. niedrigeren Werten bezüglich der Referenzverteilung ist insbesondere abhängig von den erzeugten Fragmenten.

Demnach sind die beobachteten Effekte ein Resultat des Zusammenspiels aller Ei-

6.2 Charakterisierung von Fragmenträumen

Inhibitor	enum. Molekül	Inhibitor	enum. Molekül
0.95		0.93	
0.90		0.95	
0.92		0.93	
0.93		0.95	
0.92		0.92	

Abbildung 6.7: Ausgewählte Paare von enumerierten Molekülen und dazugehörigen Inhibitoren für die Target-Klassen ACE, CDK2, COX-1, ER-Agonisten und SRC (von oben nach unten). Der jeweilige FTree-Ähnlichkeitswert ist unterhalb jedes Paares angegeben. Für die enumerierten Moleküle sind keine expliziten Stereozentren angegeben, da diese beim Vergleich der Strukturen nicht berücksichtigt werden und während der Enumeration ein beliebiges Stereoisomer erzeugt worden ist.

genschaften eines Fragmentraums. Da die Shredding-Prozedur für alle Target-Klassen identisch war, müssen die beobachteten Unterschiede maßgeblich auf die zugrunde liegenden Fragmente zurückzuführen sein. Demzufolge haben bestimmte Fragmente einen eher generischen Charakter und andere wiederum sind sehr spezifisch, was sich in den jeweiligen Ergebnissen widerspiegelt.

In Abbildung 6.7 sind ausgewählte Paare enumerierter Moleküle und dazugehöriger Inhibitoren mit den jeweiligen FTree-Ähnlichkeitswerten dargestellt. Anhand der Beispiele wird die topologische Ähnlichkeit offensichtlich.

6.2.3 Performance der BRICS-Fragmenträume

In Analogie zu der Analyse der fokussierten Fragmenträume wurden auch die BRICS-Fragmenträume zunächst hinsichtlich einiger grundlegender Eigenschaften charakterisiert. Aufgrund der Tatsache, dass die Anzahl der enthaltenen Fragments sehr viel höher ist als bei den fokussierten Fragmenträumen, sind die Räume vollständig verbunden und prinzipiell unendlich groß. Wird die bereits oben verwendete obere Schranke von maximal fünf Fragmenten pro Molekül zu Grunde gelegt, um die Menge theoretisch zugänglicher Moleküle abzuschätzen, erhält man Werte von etwa 10^{13} für den RECAP-Referenzraum und mindestens 10^{16} für die BRICS-Fragmenträume. Die Ergebnisse sind in Tabelle 6.6 zusammengefasst.

Die Qualität des neuen BRICS-Modells wurde in einem ersten Schritt anhand des Vergleichs des BRICS_4k Fragmentraums und des RECAP_4k Referenz-Fragmentraums validiert. Anschließend wurden die drei neuen Fragmenträume BRICS_4k, BRICS_9k und BRICS_20k hinsichtlich ihrer Vorhersagekraft verglichen.

Die Performance der jeweiligen Fragmenträume wurde mithilfe unterschiedlicher Anfrage-Datensätze gemessen. Diese wurden separat für das jeweilige Regelwerk erstellt. Bei den Datensätzen handelt es sich um drei unabhängige Zusammenstellungen von Molekülen, wobei jedes Molekül mindestens eine Bindung aufweist, die unter Verwendung des jeweiligen retrosynthetischen Regelwerks geschnitten wird. Die ersten beiden Anfragemengen wurden aus denselben Datensätzen erstellt, die auch zur Erzeugung der Fragmenträume verwendet wurden, d.h., aus den aufbereiteten ZINC und WDI Datensätzen. Die dritte Menge wurde aus dem in identischer Weise aufbereiteten PubChem Datensatz erzeugt. Bei letzterem handelt es sich um einen echten Testdatensatz, da keines der darin enthaltenen Moleküle in der ZINC Datenbank und im WDI enthalten ist, und somit nicht zur Erzeugung der Fragmente verwendet werden kann. Um eine vergleichbare Größe der Datensätze zu gewährleisten, wurden aus der ZINC und PubChem Datenbank randomisiert diverse Moleküle ausgewählt, sodass alle Anfrage-Datensätze zwischen 30000 und 35000 Einträge enthalten.

Jedes Molekül aus jedem Anfrage-Datensatz soll möglichst exakt aus den Fragmenten des jeweiligen Fragmentraums mit dem dazugehörigen Regelwerk rekonstruiert werden. Dazu wurde die Feature Tree Fragmentraum-Suchmethode (FTree-FS) verwendet, da sie das einzige verfügbare Software-Werkzeug ist, das es ermöglicht, in großen Fragmenträumen die global ähnlichsten Lösungen zu einem gegebenen Anfrage-Molekül in akzeptabler Zeit zu identifizieren [Rarey, 2001].

Tabelle 6.6: Grundlegende Eigenschaften der BRICS-Fragmenträume

Fragmentraum	Anzahl Fragmente	Größe des Raums	Anzahl Komponenten	Theoret. Anzahl Moleküle
RECAP_4k (Referenz)	4125	∞	1	1,74e+13
BRICS_4k	4800	∞	1	1,27e+16
BRICS_9k	9344	∞	1	5,83e+17
BRICS_20k	22343	∞	1	4,66e+19

Mithilfe des FTree-FS-Programms wurden zu jedem Anfrage-Molekül die 25 besten Lösungen bezüglich des FTree-Ähnlichkeitsmaßes berechnet. Da der Feature Tree ein topologischer Deskriptor ist, wurden die Lösungen zur Berücksichtigung umfassenderer struktureller Details bezüglich des Tanimoto-Abstands [Tanimoto, 1957] der jeweiligen öffentlichen MDL Fingerprints neu sortiert [Durant *et al.*, 2002]. Für die weiteren Analysen wurde jeweils nur die damit identifizierte beste Lösung für jedes Anfrage-Molekül berücksichtigt (vgl. [Mauser & Stahl, 2007]).

Mit den Ergebnissen der nachprozessierten Lösungen der multiplen FTree-FS-Rechnungen wurden auf die Größe der Anfrage-Datensätze normierte Histogramme bezüglich des Tanimoto-Abstands der MDL Fingerprints erstellt. Anhand der Verteilungen kann das Potenzial des jeweiligen Ansatzes quantifiziert werden. Das erste Kriterium für die Vorhersagekraft des jeweiligen Fragmentraums ist die Menge der gemäß der verwendeten Ähnlichkeitsmaße exakt rekonstruierten Anfrage-Moleküle, d.h., der Prozentsatz der bei einem Ähnlichkeitswert von 1,0 erzielt wurde. Weiterhin ist auch der Vergleich der Verteilungen der Ähnlichkeitswerte von Bedeutung, da dies ein Maß für die Fähigkeit des Modells ist, strukturell eng verwandte Lösungen zu erzeugen, sofern das eigentliche Anfrage-Molekül nicht rekonstruiert werden kann.

6.2.3.1 Vergleich von BRICS und RECAP

Abbildung 6.8 zeigt die Ergebnisse des Vergleichs der Fragmenträume BRICS_4k und RECAP_4k. Der Prozentsatz der exakt rekonstruierten Lösungen konnte in allen drei Testfällen mit dem neuen Regelwerk mehr als verdoppelt werden. So konnten für den WDI 16,7 anstelle von 7,2 Prozent, für ZINC 11,5 statt 3,2 und für PubChem 9,0 anstatt 2,3 Prozent der Anfragemoleküle rekonstruiert werden. Bei den Verteilungen für ZINC und PubChem kann weiterhin eine klare Verschiebung der Verteilungen hin zu höheren Ähnlichkeitswerten beobachtet werden.

Die Ergebnisse zeigen deutlich, dass die mithilfe des BRICS-Modells erzeugten Fragmente und die dazugehörigen Verknüpfungsregeln die unterschiedlichen chemischen Motive signifikant besser abbilden als RECAP. Dies trifft sowohl auf die biologisch aktiven Moleküle aus dem WDI zu, als auch auf kommerziell verfügbare Verbindungen aus der ZINC sowie Substanzen aus der PubChem Datenbank. Gleichzeitig wurde die Überschneidung aus biologisch aktiven und aus kommerziell verfügbaren Verbindungen ab-

6 Erzeugung und Charakterisierung von Fragmenträumen

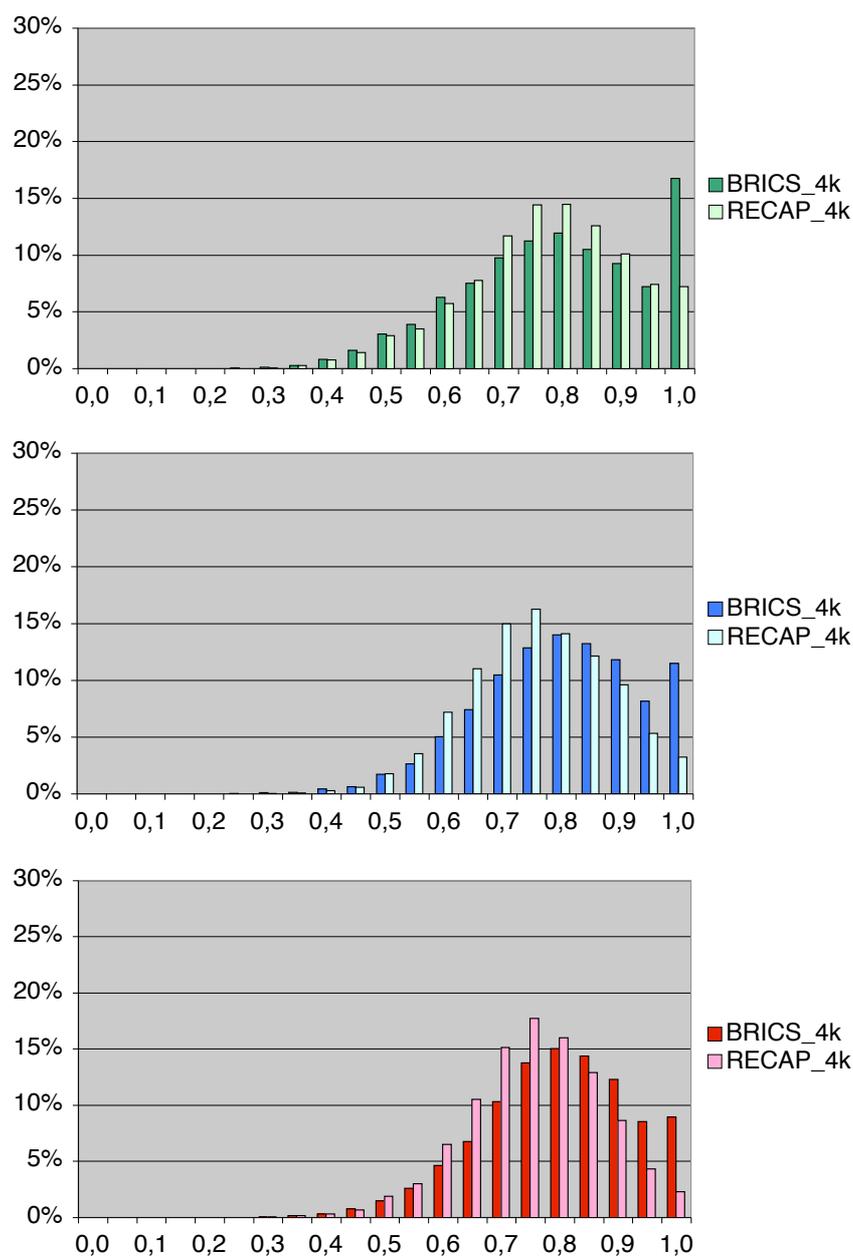


Abbildung 6.8: Performance der BRICS_4k und RECAP_4k Fragmenträume: Die Abbildungen zeigen die Ergebnisse für WDI (oben, grün), ZINC (mittig, blau) und PubChem (unten, rot). Aufgetragen sind die auf die Größe des Anfrage-Datensatzes normierten Histogramme der Ähnlichkeitswerte der jeweils besten Lösung für jedes Anfrage-Molekül.

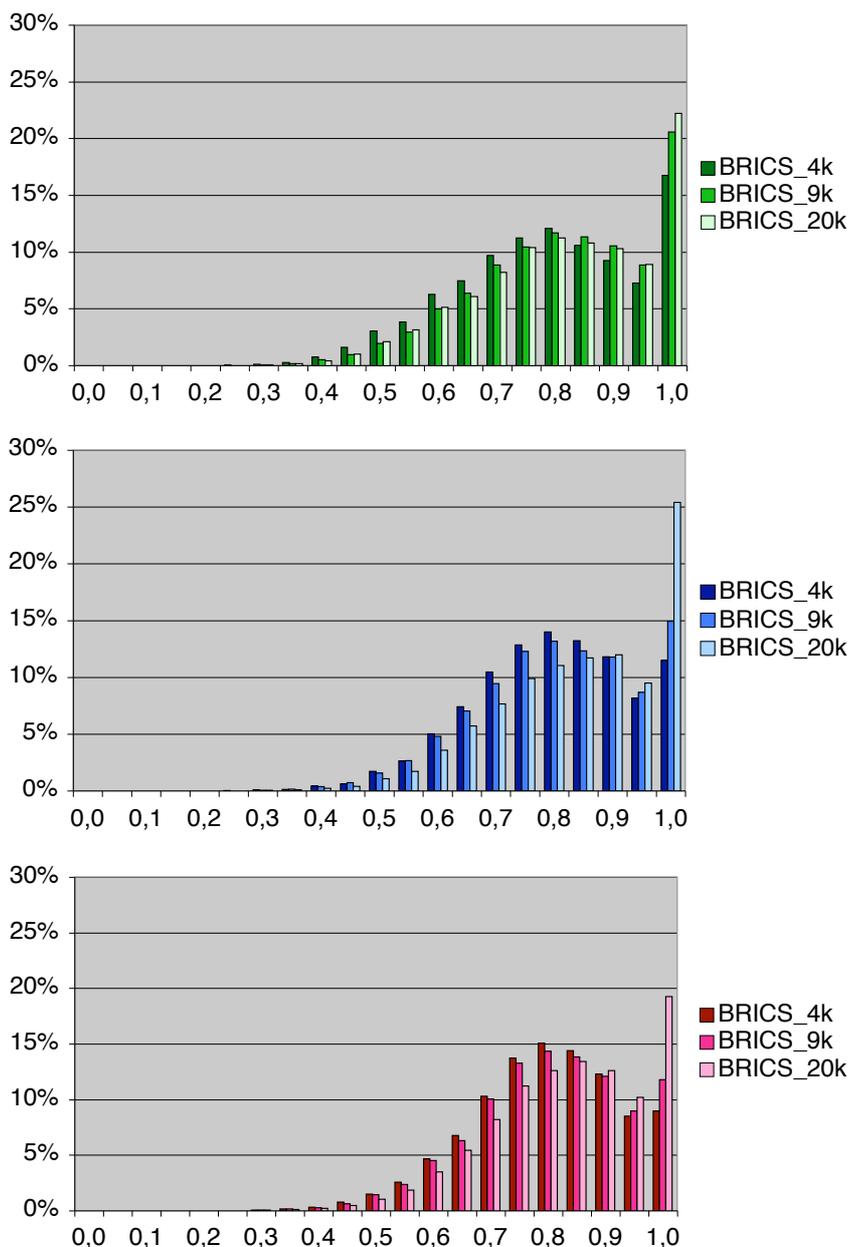


Abbildung 6.9: Performance der BRICS_4k, BRICS_9k und BRICS_20k Fragmenträume: Die Abbildungen zeigen die Ergebnisse für WDI (oben, grün), ZINC (mitig, blau) und PubChem (unten, rot). Aufgetragen sind die auf die Größe des Anfrage-Datensatzes normierten Histogramme der Ähnlichkeitswerte der jeweils besten Lösung für jedes Anfrage-Molekül.

geleiteter (identischer) Fragmente um ungefähr 16 Prozent erhöht. Die größere Anzahl an Fragmenten allein führt dabei zu keiner signifikanten Verbesserung der Performance. Dies wurde mithilfe von fünf randomisierten BRICS_4k Fragmenträumen untersucht, die jeweils auf 4115 bis 4154 Fragmente reduziert worden sind und untereinander einen Fragment-Überlapp von weniger als 75 Prozent haben. Dabei zeigte sich keine signifikante Veränderung der erhaltenen Verteilungen, sodass der Einfluss der Anzahl der Fragmente in diesem Zusammenhang vernachlässigt werden kann.

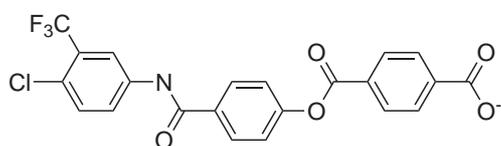
6.2.3.2 Vergleich der angereicherten BRICS-Fragmenträume

Die Ergebnisse des Vergleichs der angereicherten BRICS-Fragmenträumen BRICS_9k und BRICS_20k sind in Abbildung 6.9 dargestellt. Für alle Testfälle konnte eine weitere deutliche Verbesserung der Performance erzielt werden. Für die WDI-Anfragemoleküle wurde eine Steigerung von 16,7 auf 20,6 bzw. 22,2 Prozent erreicht. Bei ZINC konnte eine Verbesserung von 11,5 auf 14,9 bzw. 25,4 Prozent erzielt werden. Die Performance für PubChem zeigt eine Erhöhung von 9,0 auf 11,8 bzw. 19,2 Prozent. In allen Fällen kann auch eine leichte Steigerung bei geringeren Ähnlichkeitswerten beobachtet werden.

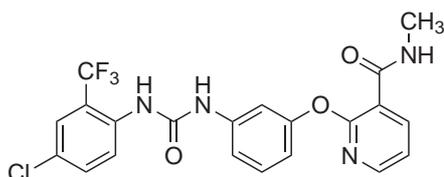
Die Steigerung von BRICS_4k zu BRICS_9k fällt insgesamt deutlich, aber moderat aus und liegt generell zwischen drei und vier Prozent. Die in BRICS_9k zusätzlich enthaltenen Fragmente stammen ausschließlich aus kommerziell erhältlichen Molekülen aus der ZINC Datenbank, sie weisen aber eine hohe Ähnlichkeit zu den Fragmenten der biologisch aktiven Moleküle aus dem WDI auf. Diese Tatsache zeigt, dass auch mit einem Fragment-Datensatz, der nicht ausschließlich aus Inhibitoren erzeugt wurde, eine deutliche Steigerung der Performance für Anfrage-Moleküle aus dem WDI erreicht werden konnte.

Gleichzeitig wird deutlich, dass bezüglich des WDI keine weitere deutliche Verbesserung mehr möglich ist, wohingegen für ZINC und PubChem ein erneuter Zuwachs von 10,5 bzw. 7,5 Prozent zu erkennen ist. Diese Steigerung hat mehrere Ursachen: Zum einen handelt es sich bei dem WDI um den mit Abstand kleinsten Datensatz und zum anderen macht die Menge der Anfrage-Moleküle etwa 50 Prozent des Gesamt-Datensatzes aus. Dieser Wert liegt um ein bis zwei Größenordnungen über den Werten für ZINC und PubChem. Die initiale Menge an rekonstruierten Molekülen fällt unter anderem deswegen beim WDI am größten aus, was allerdings auch eine geringere Steigerungsmöglichkeit bei dem größten Fragment-Datensatz bedeutet.

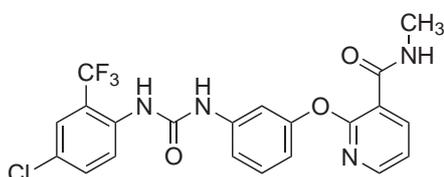
Für die ZINC- und PubChem-Anfragemoleküle konnte die ursprüngliche Menge an rekonstruierten Molekülen mehr als verdoppelt werden. Dies führt insgesamt dazu, dass für alle Beispiele letztendlich sehr gute und miteinander vergleichbare Ergebnisse erzielt werden konnten. Da die Ähnlichkeit zu den Fragmenten aus dem WDI von BRICS_9k zu BRICS_20k weiter abnimmt, können sowohl kommerziell als auch öffentlich verfügbare Moleküle mit diesen Fragmenträumen besser reproduziert werden. Dieser Effekt fällt für ZINC stärker aus als für PubChem, da die in den größeren Fragment-Sets enthaltenen Fragmente ausschließlich aus der ZINC Datenbank stammen, und somit auch eine größere Ähnlichkeit zu den dort auftretenden strukturellen Motiven zu erwarten ist.



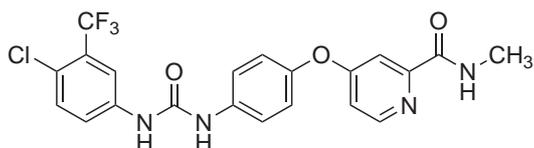
RECAP_4k

0,945 Feature Tree
0,704 MDL Fingerprint

BRICS_4k

0,999 Feature Tree
0,926 MDL Fingerprint

BRICS_9k

0,999 Feature Tree
0,926 MDL Fingerprint

BRICS_20k

0,999 Feature Tree
1,000 MDL Fingerprint

Abbildung 6.10: Rekonstruktion des Kinase-Inhibitors Sorafenib aus den erstellten Fragmenträumen. Dargestellt sind die jeweils am besten bewerteten Moleküle, die mithilfe von FTree-FS und anschließender Neubewertung der Ähnlichkeit zu dem Anfragemolekül unter Verwendung der MDL Fingerprints erhalten wurden. Der Grund für diese Vorgehensweise ist gut zu erkennen: Der FTree-Deskriptor ist nicht in der Lage, unterschiedliche Substitutionsmuster an Ringen zu erkennen. Dies zeigt sich in den jeweiligen Ähnlichkeitswerten. Diese sind für den Feature Tree nicht exakt 1,0 da im Laufe der Berechnungen vom Algorithmus kleine Fehler bei der Volumenabschätzung der Fragmente gemacht werden [Rarey, 2001]. Durch die Neubewertung mit MDL Fingerprints wird deutlich, dass eine exakte Rekonstruktion von Sorafenib in diesem Fall nur durch Verwendung von BRICS_20k möglich ist.

6 Erzeugung und Charakterisierung von Fragmenträumen

Zusammenfassend ist festzuhalten, dass die neuen Fragmenträume Moleküle aus unterschiedlichen Datenquellen wesentlich besser repräsentieren als der Referenz-Fragmentraum. Das zugrunde liegende Modell bildet somit die chemischen Motive genauer und umfassender ab. Abbildung 6.10 zeigt exemplarisch die besten Ergebnisse aus den vier Fragmenträumen für den Inhibitor Sorafenib (s. Abb. 4.4) mit den dazugehörigen Ähnlichkeitswerten. In diesem speziellen Fall war es lediglich durch Verwendung des Fragmentraums BRICS_20k möglich, das Molekül exakt zu rekonstruieren.

7 Strukturbasierte Suche in Fragmenträumen

Die in den Kapiteln 4 und 5 vorgestellten Modelle und Algorithmen sind im Programm FlexNovo realisiert, das auf der Bibliothek von FlexX basiert. FlexNovo ist in ANSI C entwickelt und besteht zurzeit aus etwa 25000 Zeilen Programmcode.

Auf die Erzeugung und Charakterisierung von Fragmenten bzw. Fragmenträumen wurde bereits in Kapitel 6 eingegangen. Im folgenden Kapitel werden sowohl auf Basis bereits existierender als auch mithilfe der neu erzeugten Fragmenträume Vorhersagen getroffen, wofür unter anderem das neu entwickelte Programm FlexNovo zum Einsatz kam.

7.1 Parametrisierung von FlexNovo

Berechnungen mit FlexNovo basieren auf einer Vielzahl von Parametern, die sich aus den verwendeten Modellen und Algorithmen ergeben. Da sowohl das Verfahren zur Platzierung von Fragmenten als auch der inkrementelle Aufbaualgorithmus von FlexX adaptiert wurden, wird zunächst kurz auf die in diesem Zusammenhang wichtigen Aspekte eingegangen. Im Anschluss daran werden die spezifischen algorithmischen Parameter für die strukturbasierte Suche in Fragmenträumen vorgestellt.

7.1.1 Abhängigkeiten von FlexX

Wie bereits in den vorangegangenen Kapiteln beschrieben, können Fragmente bis auf wenige essenzielle Unterschiede als kleine Liganden angesehen werden. Aus diesem Grund kann die zugrunde liegende Parametrisierung von FlexX für die meisten Aspekte übernommen werden. Entsprechend beinhaltet dies die in Kapitel 4.1.1 aufgeführten chemischen Modelle zur Annotation der Eigenschaften von Atomen und Bindungen sowie das Konformationsmodell zur Beschreibung molekularer Flexibilität. Dazu zählen weiterhin die in Kapitel 4.1.3 vorgestellten geometrischen Modelle zur Abbildung molekularer Wechselwirkungen, die erlaubten Überlappvolumina zwischen Fragmenten und dem Rezeptor sowie die energetischen Beiträge der einzelnen Terme in der Bewertungsfunktion. Des Weiteren werden die Parameter zur Steuerung der Diskretisierung der geometrischen Daten sowie der Toleranz der Algorithmen zur Platzierung und für den inkrementellen Aufbau von Fragmenten verwendet.

Sämtliche Parameter sind in den entsprechenden Konfigurationsdateien abgelegt und dokumentiert, weshalb an dieser Stelle nicht weiter darauf eingegangen wird. Im Gegensatz zu FlexX ist dabei allerdings zu beachten, dass die Richtungsinformation der

ausgehenden Bindungen zu Link-Atomen beim Clustering von Platzierungen immer explizit berücksichtigt werden.

7.1.2 Algorithmische Parameter

Die spezifischen Parameter für FlexNovo dienen primär der Steuerung des Aufbaualgorithmus. Es werden höchstens zwei Anbauzyklen durchgeführt. Die Menge der tatsächlich gespeicherten Platzierungen pro Fragment bzw. Teillösung ist auf 60 beschränkt. Die maximale Anzahl an Teillösungen, die in die nächste Iteration übernommen werden, beträgt $k_F = 100$. Die maximale Anzahl jedes „atomaren“ Fragments innerhalb einer Menge von Teillösungen ist auf zehn beschränkt. Die finalen Lösungen dürfen untereinander jeweils maximal ein „atomares“ Fragment gemeinsam haben. Die Menge der gespeicherten Platzierungen für die finalen Lösungen ist auf zehn festgelegt. Die finale Lösungsliste enthält maximal 100 Moleküle.

Alle Parameter können vom Benutzer individuell angepasst werden. Sie sind in einer gesonderten Datei abgelegt und können sowohl dort als auch innerhalb des Programms selbst verändert werden. Eine detaillierte Beschreibung ist im FlexNovo-Benutzerhandbuch [Degen & Rarey, 2007] bzw. im Anhang A.2.2 zu finden. Dort sind auch die weiteren Parameter zur Steuerung der Filterfunktionen für Platzierungen dokumentiert. Auf den Einfluss der algorithmischen Parameter auf die Abdeckung des Lösungsraums sowie das Laufzeitverhalten wird in Kapitel 7.3 im Detail eingegangen.

7.2 Fallstudien

Die experimentelle Erzeugung neuer Moleküle umfasst typischerweise mehrere Syntheseschritte und findet unter kontrollierten Laborbedingungen statt. Dabei bedarf es häufig einer Reihe von Experimenten bis das tatsächlich gewünschte Produkt mit entsprechend hoher Ausbeute und Reinheit sowie mit möglichst geringem experimentellen Aufwand hergestellt werden kann. Zur weiteren Bestimmung der Bindungskonstante bzw. Bindungsaffinität werden in der Regel IC_{50} -Messungen [Cheng & Prusoff, 1973] durchgeführt. Die Aufklärung des Bindungsmodus erfolgt mithilfe der Röntgenstrukturanalyse und NMR-Spektroskopie. Nicht jede Kombination von Molekül und Protein ist dabei für jede experimentelle Technik geeignet, sodass auch die Durchführungen der dazugehörigen Messungen häufig eine Optimierung der dazugehörigen Parameter und Bedingungen erfordern.

Um das tatsächliche Potenzial von FlexNovo zu evaluieren muss zunächst die Tatsache in Betracht gezogen werden, dass es prinzipiell für de-novo-Design-Verfahren keine etablierten Validierungsstrategien gibt, die eine quantitative Evaluation der Güte der Vorhersagen erlauben. Da die Aufgabe von FlexNovo primär aus der Erzeugung neuer Moleküle und Variation bekannter struktureller Motive besteht, können im Gegensatz zu Docking-Verfahren keine entsprechenden retrospektiven Analysen und Anreicherungsstudien durchgeführt werden.

Aus diesem Grund wurden mehrere repräsentative Fallstudien durchgeführt, für die jeweils ein Satz bekannter Inhibitoren zusammengestellt werden konnte, mit denen die

erzeugten Moleküle verglichen wurden. Aus den jeweiligen Ergebnissen werden zunächst repräsentative Moleküle vorgestellt, anhand derer generelle Eigenschaften der dazugehörigen Lösungsmenge ersichtlich werden. Für eine umfassendere Charakterisierung der Ergebnisse insgesamt werden Ähnlichkeitsanalysen zu den bekannten Inhibitoren durchgeführt, die Verteilungen der physikochemischen Eigenschaften vergleichend analysierten, die Diversität der jeweiligen Lösungsmengen bestimmt sowie die synthetische Zugänglichkeit quantitativ ermittelt.

Im Folgenden wird zunächst auf die Testsysteme sowie die Aufbereitung und Prozessierung der Eingabedaten eingegangen. Im Anschluss daran werden die durchgeführten Rechnungen im Detail beschrieben sowie die Ergebnisse vorgestellt und diskutiert. Dabei liegt der Fokus auf der Abschätzung des Potenzials der Methode und der Analyse charakteristischer Phänomene des verwendeten Ansatzes in unterschiedlichen Szenarien. Die Optimierung des Prozesses für ein spezielles Beispiel steht dabei nicht im Vordergrund. Für letzteres werden jedoch aus den vorgestellten Beispielen im Laufe der jeweiligen Diskussion verschiedene Ansatzpunkte zur Optimierung herausgearbeitet.

7.2.1 Überblick über die Testsysteme

Für die Validierungsexperimente wurden vier Rezeptoren ausgewählt, durch die unterschiedliche Charakteristika in typischen Szenarien im Rahmen der Leitstruktursuche abgedeckt werden können. Bei diesen handelt es sich um Dihydrofolat-Reduktase, Cyclin-abhängige Kinase 2, Cyclooxygenase-2 und den Estrogenrezeptor, die zunächst kurz pharmakologisch eingeordnet werden sollen.

Bei der Dihydrofolat-Reduktase (DHFR) handelt es sich um ein wichtiges Enzym für das Zellwachstum. Sie wird im Rahmen der Malaria- und Krebsforschung untersucht. DHFR wird häufig zur Evaluation strukturbasierter molekularer Design-Verfahren verwendet. Die Cyclin-abhängige Kinase 2 (CDK2) spielt neben anderen Kinasen eine wichtige regulierende Rolle bei Krebserkrankungen und ist in diesem Zusammenhang Gegenstand intensiver pharmazeutischer Forschung. Die Hemmung der Cyclooxygenase-2 (COX-2) ist von zentraler Bedeutung bei der medikamentösen Behandlung von Entzündungsprozessen und der damit verbundenen Symptome. Parallel wird sie bezüglich ihrer Funktion bei onkologischen und neurologischen Erkrankungen untersucht. Die selektive Modulation des Estrogenrezeptors (ER) ist im Rahmen von Herz-Kreislauf-Erkrankungen und der Krebstherapie von Bedeutung und stellt ein breites Forschungsfeld in der pharmazeutischen Industrie dar.

Basierend auf den Eigenschaften der Aktiven Zentren lassen sich die Enzyme bzw. Rezeptoren in zwei Klassen einteilen. Die Bindetaschen der Dihydrofolat-Reduktase und der Cyclin-abhängigen Kinase 2 sind stark exponiert zum umgebenden Solvent und zeichnen sich durch eine Reihe spezifischer gerichteter Wechselwirkungsmöglichkeiten aus. Im Gegensatz dazu handelt es sich bei der Bindetasche der Cyclooxygenase-2 und des Estrogenrezeptors hauptsächlich um lipophile, geschlossene Kavitäten, die folglich nur wenige Möglichkeiten zur Ausbildung gerichteter Wechselwirkungen bieten.

Der Fokus liegt demzufolge auf den unterschiedlichen strukturellen Aspekten der Rezeptoren und spiegelt sich ebenso in den Randbedingungen der Rechnungen wider.

7.2.2 Aufbereitung und Prozessierung der Eingabedaten

Die durch Röntgenstrukturanalyse bestimmten Komplexe der Rezeptoren mit kokristallisierten Liganden wurden aus der Brookhaven Protein Datenbank (PDB) entnommen [Berman *et al.*, 2000] (siehe Tab. 7.1). Die Definition der Bindetaschen erfolgte durch Auswahl aller Atome der jeweiligen Aminosäuren, die innerhalb einer Kugel mit einem Radius von 6,5 Å um den Liganden liegen. Die wesentlichen Ausschnitte der resultierenden Struktur werden im Zusammenhang mit den erzielten Ergebnissen vorgestellt.

Für die Untersuchungen wurde der Feature Tree Fragmentraum, *FT-Raum*, verwendet. Dieser enthält in etwa 16000 Fragmente, die ursprünglich aus dem World Drug Index erzeugt worden sind [Lewell *et al.*, 1998] und ist bereits für Rechnungen im Rahmen der Feature Tree Fragmentraum-Suchmethode verwendet worden [Rarey, 2001]. Aus dem FT-Raum wurde ein zweiter Fragmentraum abgeleitet. Dafür kamen zum einen Substruktur-Filter zur Identifikation reaktiver und toxischer Verbindungen zum Einsatz, die den Veröffentlichungen von Baurin *et al.* [Baurin *et al.*, 2004a,b] und Kazius *et al.* [Kazius *et al.*, 2005] entnommen wurden. Zum anderen wurden physikochemische Filterkriterien angewandt, die das Molekulargewicht der Fragmente auf 300, die Anzahl der Wasserstoffbrücken-Akzeptoren und -Donoren auf jeweils sechs und die Anzahl an Stereozentren auf eins beschränkt. Der durch Anwendung dieser Kriterien resultierende *FTsub-Raum* besteht aus etwa 4300 Fragmenten.

Im Zuge der Präprozessierung wurden zwei Docking-Rechnungen für alle Fragmente beider Fragmenträume mit zwei unterschiedlichen Bewertungsfunktionen durchgeführt, zum einen mit der in Kapitel 4.1.4 vorgestellten FlexX-Bewertungsfunktion, *FlexScore* [Rarey *et al.*, 1996a], und zum anderen mit der Bewertungsfunktion *ScreenScore* [Stahl & Rarey, 2001]. Dabei dient jeweils die erste Rechnung der Erzeugung der Startplatzierung und somit gleichzeitig der Auswahl der Start-Fragmente und wird jeweils mit einem entsprechenden Anker-Pharmakophor durchgeführt. Auf letzteres wird bei der Diskussion der Ergebnisse eingegangen. In der zweiten Rechnung werden die receptorspezifischen maximalen Energiebeiträge der Fragmente abgeschätzt. Alle erzeugten Daten wurden in Dateien zwischengespeichert.

Des Weiteren wurde für jeden Rezeptor ein Satz von bekannten Inhibitoren aus der Literatur zusammengestellt. Sofern der Ligand der verwendeten Kristallstruktur nicht schon Bestandteil dieser Menge war, wurde er explizit hinzugefügt. Die Zusammenstellungen wurden unter anderem zur Ableitung von klassenspezifischen Wertebereichen für physikochemische Eigenschaften herangezogen. Die Referenzen sowie die Zusammenstellung der Eigenschaften sind in Tabelle 7.1 zu finden.

7.2.3 Durchführung der Rechnungen

Die Rechnungen für alle Rezeptoren wurden mit denselben bereits in Kapitel 7.1.2 zusammengefassten algorithmischen Parametern durchgeführt. Der Aufbauprozess bestand demnach aus zwei Anbauzyklen, womit die resultierenden Moleküle aus bis zu drei „atomaren“ Fragmenten bestehen. Aufgrund der breiten Molekulargewichtsverteilung der verwendeten Fragmente ist dies zunächst kein Maß für die tatsächliche Größe der Mole-

Tabelle 7.1: Daten und Referenzen zu den vier Targets

	DHFR	CDK2	COX-2	ER
PDB-ID der Kristallstruktur	4dfr ^a	1di8 ^b	6cox ^c	1err ^d
Anzahl der Inhibitoren	69 ^e	73 ^f	32 ^g	53 ^h
Molekulargewicht	200 – 480	200 – 480	200 – 460	200 – 480
Anzahl Ringe	0 – 4	0 – 6	0 – 4	0 – 6
Anzahl rotierbarer Bindungen	0 – 12	0 – 12	0 – 10	0 – 12
Anzahl H-Brücken Akzeptoren	0 – 13	0 – 9	0 – 8	0 – 8
Anzahl H-Brücken Donoren	0 – 7	0 – 6	0 – 3	0 – 5
Berechneter logP	-2.5 – 2.5	-2.5 – 6.0	1.0 – 7.0	1.0 – 9.0

^a[Bolin *et al.*, 1982] ^b[Shewchuk *et al.*, 2000] ^c[Kurumbail *et al.*, 1996] ^d[Brzozowski *et al.*, 1997] ^e[Selassie *et al.*, 1989] ^f[Rosania & Chang, 2000; Sielecki *et al.*, 2000; Dumas, 2001] ^g[Carter, 1998; Friesen *et al.*, 1998; Kalgutkar, 1999] ^h[Magarian *et al.*, 1994; Coghlan & Kort, 1999; Gegnas, 2000; Sippl, 2000]

küle. Nach jedem Anbauzyklus wurden die 100 besten Fragmente für den nächsten Zyklus verwendet und pro Fragment bis zu 60 unterschiedliche Platzierungen berücksichtigt. Die finale Ergebnisliste bestand jedoch jeweils nur aus bis zu 50 unterschiedlichen Molekülen.

Für jeden Rezeptor wurden mehrere Rechnungen mit verschiedenen Diversitätsparametern durchgeführt, die so gewählt wurden, dass der Algorithmus prinzipiell dasselbe Start-Fragment für alle erzeugten Moleküle verwenden konnte. Dadurch sollte eine möglichst große Anzahl an Molekülen mit hoher Ähnlichkeit zu existierenden Inhibitoren erzeugt werden. Weiterhin wurden analog zu der Präprozessierung jeweils Rechnungen mit zwei unterschiedlichen Bewertungsfunktionen durchgeführt, wobei die bereits erwähnten Funktionen FlexScore und ScreenScore zum Einsatz kamen. Zusätzlich wurden die in Tabelle 7.1 aufgeführten physikochemischen Filterkriterien verwendet.

7.2.4 Suche nach alternativen Leitstrukturen

Bei der Analyse der Ergebnisse liegt der Fokus auf der Identifikation von Molekülen, die eine hohe Komplementarität zu der jeweiligen Bindetasche aufweisen, chemisch sinnvoll sind sowie ein vielversprechendes physikochemisches Profil und einen niedereenergetischen Bindungsmodus aufweisen. Aufgrund der Größe der verwendeten Fragmenträume ist es jedoch unrealistisch zu erwarten, dass der Algorithmus hauptsächlich Strukturen erzeugt, die bekannten Inhibitoren sehr ähnlich sind. Des Weiteren kann nicht davon ausgegangen werden, dass ein einzelnes Molekül alle spezifizierten Randbedingungen nahezu optimal erfüllt.

Wie bereits in Kapitel 5 beschrieben, bewirkt der verwendete Algorithmus, dass die Moleküle der Ergebnismenge sowohl ähnlich zueinander sind, als auch bezüglich der vorhergesagten freien Bindungsenthalpien relativ dicht zusammenliegen. Aus diesem Grund

sind die exakten Informationen über die Ränge der einzelnen Moleküle in der finalen Lösungsliste und die dazugehörigen vorhergesagten freien Bindungsenthalpien in diesem Zusammenhang nur von untergeordnetem Interesse. Vielmehr steht die Identifikation sinnvoller struktureller Variationen und die Erfüllung weiterer wichtiger Eigenschaften durch die jeweiligen Moleküle im Vordergrund.

Von einem automatisierten de-novo-Design-Verfahren ist in diesem Zusammenhang nicht zu erwarten, dass alle erzeugten Moleküle auch tatsächlich als potentielle Leitstrukturen in Frage kommen. Aufgrund der starken strukturellen Fokussierung der erzeugten Moleküle werden bei der Diskussion der Ergebnisse ausgewählte, handselektierte Lösungen gezeigt. Dies entspricht der üblichen Vorgehensweise im Rahmen der Identifikation und Optimierung von Leitstrukturen und wird deshalb auch an dieser Stelle so durchgeführt. Anhand repräsentativer Ergebnisse werden typische Charakteristika für die jeweiligen Szenarien diskutiert.

7.2.4.1 Ergebnisse für die vier Rezeptoren

Aus den Lösungsmengen wurden jeweils mehrere repräsentative Moleküle zur Erläuterung der unterschiedlichen spezifischen Aspekte der Ergebnisse ausgewählt. Die Beschreibung und Diskussion der prinzipiellen strukturellen Eigenschaften erfolgt dabei anhand zweidimensionaler Molekülrepräsentationen, die mit der jeweiligen verwendeten Kristallstruktur verglichen werden. Weitere Beispiele sind in Abbildung 7.10 enthalten, wobei dabei der Fokus auf den vorhergesagten Bindungsmodi und der Komplementarität zur Rezeptoroberfläche liegt. Schließlich enthält Abbildung 7.11 die vorhergesagten Bindungsmodi weiterer repräsentativer Moleküle überlagert mit den strukturell ähnlichsten Inhibitoren aus den jeweiligen Zusammenstellungen.

Dihydrofolat-Reduktase Für die Rechnungen wurde die Kristallstruktur (PDB-ID: 4dfr [Bolin *et al.*, 1982]) verwendet, die den kokristallisierten Inhibitor Methotrexat enthält. Das Anker-Pharmakophor zur Erzeugung der Start-Platzierungen besteht aus einer Wasserstoffbrücken-Wechselwirkung mit Asp27. Abbildung 7.1 enthält eine zweidimensionale Repräsentation des Aktiven Zentrums mit einigen wichtigen Aminosäuren und dem angedeuteten Bindungsmodus von Methotrexat.

Ungefähr 40 Prozent aller Fragmente der beiden Fragmenträume konnten erfolgreich im Aktiven Zentrum des Rezeptors unter Erfüllung des Anker-Pharmakophors platziert werden. Die vorhergesagten Scores reichen dabei von leicht positiven Werten bis zu -32 (kJ/mol), was zum Teil auf die Unterschiede in der Größe der Fragmente zurückzuführen ist, aber hauptsächlich aus den ausgebildeten Wechselwirkungen resultiert.

Für die meisten Moleküle in der Lösungsliste hat FlexNovo aufgrund des hohen anfänglichen Scores das Pteridin-Ringsystem als Start-Fragment ausgewählt, wie es auch in Methotrexat vorkommt. Der dazu benachbarte aromatische Ring wurde ebenso weitestgehend in der gesamten Lösungsmenge reproduziert. Innerhalb der jeweiligen Molekülmengen gibt es kleine strukturelle Variationen, so kommen überbrückende Ether- und Amingruppen vor wie auch annellierte und einzelne aromatische Ringe. Die sich diesem Motiv anschließende Amidbindung wurde häufig durch ein sekundäres oder ter-

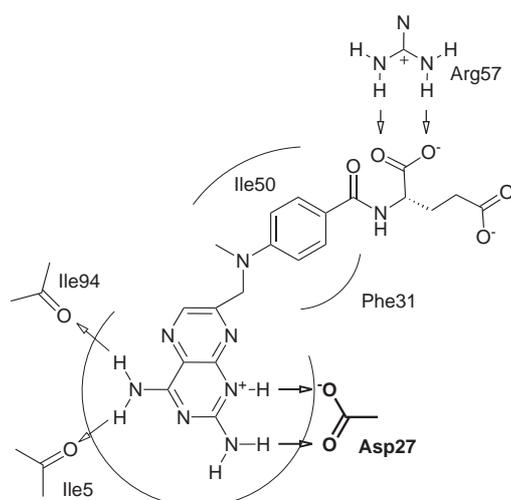


Abbildung 7.1: Schematische Darstellung der Bindetasche von DHFR (PDB-ID: 4dfr [Bolin *et al.*, 1982]) zusammen mit dem kokristallisierten Inhibitor Methotrexat (Orientierung analog des Bindungsmodus). Die vorhergesagte freie Bindungsenthalpie beträgt -54 (kJ/mol). Das Anker-Pharmakophor für die Erzeugung der Start-Platzierungen besteht aus einer Wasserstoffbrücken-Wechselwirkung mit Asp27.

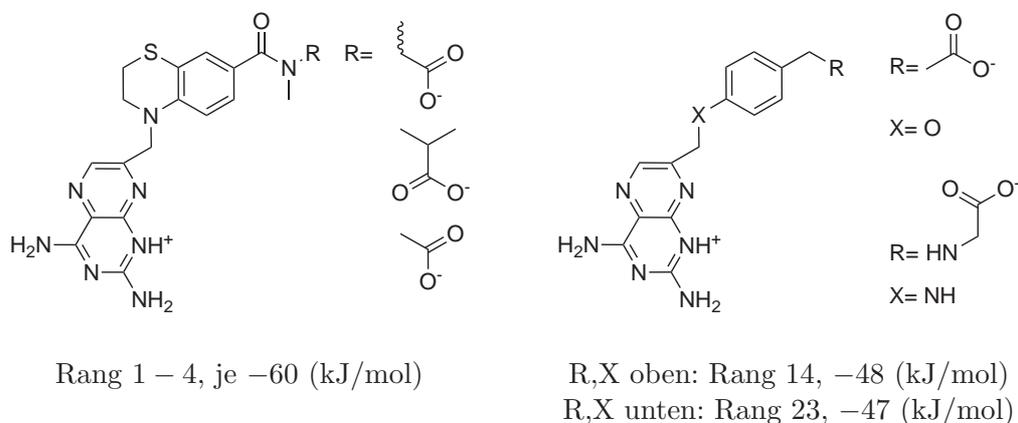


Abbildung 7.2: Repräsentative Ergebnisse für DHFR. Die Moleküle sind analog der vorhergesagten Bindungsmodi orientiert. Zusätzlich sind die Positionen in der finalen Lösungsliste und die dazugehörigen vorhergesagten freien Bindungsenthalpien angegeben. Im Beispiel auf der linken Seite ist das Stereozentrum der ersten R-Gruppe nicht explizit angegeben, da die dazugehörige Platzierungsliste beide Stereoisomere enthält.

tiäres Amin bzw. eine Methylengruppe ersetzt, die teilweise auch als Bestandteil eines Ringsystems auftreten. Diesen Gruppen folgen ein oder zwei Carboxylatgruppen in unterschiedlichen Abständen. Abbildung 7.2 zeigt einige charakteristische Moleküle.

Die Platzierungen aller erzeugten Lösungen entsprechen weitestgehend der von Methotrexat und anderer aktiver Verbindungen. Des Weiteren zeigen die dazugehörigen Moleküle sinnvolle chemische Ersetzungen. Dabei wurden von FlexNovo insbesondere alternative überbrückende Fragmente und zusätzliche Ringschlüsse vorgeschlagen, durch die eine weitere Steigerung der tatsächlichen Bindungsaffinität ermöglicht werden kann, da der Entropieverlust bei der Komplexbildung weniger stark ausfallen sollte. Abbildung 7.10 zeigt zwei weitere repräsentative Moleküle mit dem vorhergesagten Bindungsmodus in der Bindetasche des Rezeptors.

Cyclin-abhängige Kinase 2 Alle Inhibitoren, die an die ATP-Bindetasche des Rezeptors binden, bilden eine oder mehrere Wasserstoffbrücken-Wechselwirkungen mit der *Hinge-Region* im Aktiven Zentrum aus. Dieser Tatsache wurde bei der Formulierung des Anker-Pharmakophors Rechnung getragen, das eine essenzielle Wechselwirkung mit der NH-Gruppe von Leu83 fordert. Die meisten CDK2-Inhibitoren bilden eine weitere Wasserstoffbrücken-Wechselwirkung mit einer der beiden benachbarten Carbonylgruppen aus, die Bestandteil von Glu81 bzw. Leu83 sind. Dies war ebenfalls entsprechender Bestandteil des Anker-Pharmakophors. Eine zweidimensionale Repräsentation des Aktiven Zentrums der verwendeten Kristallstruktur mit den wichtigsten Aminosäuren ist in Abbildung 7.3 enthalten, worin auch der Bindungsmodus eines mikromalaren Inhibitors (entnommen aus PDB-ID: 1h1s [Davies *et al.*, 2002]) angedeutet ist.

Nach der initialen Platzierung erfüllten in etwa 35 Prozent der Fragmente das Anker-Pharmakophor. Die erreichten Scores reichen dabei von kleinen positiven Werten bis hin zu -30 (kJ/mol). Wie bereits bei den DHFR-Ergebnissen zu beobachten war, dominieren einige wenige charakteristische Fragmente die Lösungslisten, wovon die prominentesten in den gezeigten Abbildungen enthalten sind.

Die erzeugten Moleküle sind im Vergleich zu den meisten bekannten Inhibitoren aus der entsprechenden Zusammenstellung relativ groß. Die Ursache dafür liegt in der verwendeten Bewertungsfunktion, die bei der Berechnung des Scores eine gewisse Korrelation mit der Größe der Moleküle zeigt. Dieses Phänomen ist bereits vom Ligand-Docking bekannt und wird im Zusammenhang mit Fragmenträumen aufgrund der Vielzahl an erzeugten Molekülen umso offensichtlicher. Aus diesem Grund bevorzugt FlexNovo tendenziell größere Moleküle, wobei dies durch die Verwendung entsprechender Filterkriterien für physikochemische Eigenschaften einfach und effektiv kontrolliert werden kann.

Strukturell enthalten viele der erzeugten Lösungen chemische Motive von Liganden, die in sehr ähnlichen Positionen in der Bindetasche platziert worden sind. Dazu gehören die Indol-Fragmente und strukturell eng verwandte Motive, die Wechselwirkungen zum Hinge ausbilden sowie verbrückende Harnstoff- und Amid-Gruppen. Abbildung 7.4 zeigt einige der am häufigsten auftretenden Strukturen.

Nach dem Bindungsmodus-Modell von Traxler für Kinase-Inhibitoren [Traxler & Furet, 1999] (vgl. Abb. 7.5) besetzen die erzeugten Moleküle neben der Adenin-Region haupt-

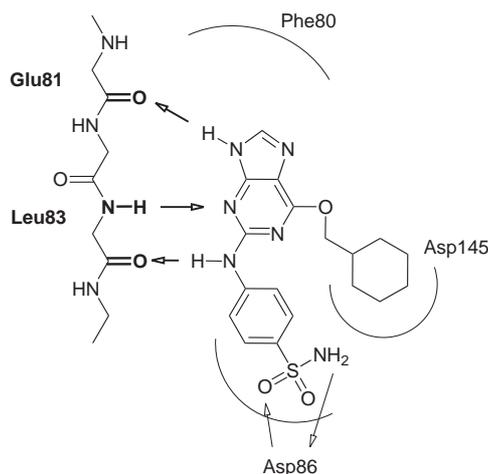


Abbildung 7.3: Schematische Darstellung der ATP-Bindetasche von CDK2 (PDB-ID: 1di8 [Shewchuk *et al.*, 2000]) zusammen mit einem Inhibitor (entnommen aus PDB-ID: 1h1s [Davies *et al.*, 2002]) und analog des Bindungsmodus orientiert, -28 (kJ/mol) vorhergesagte freie Bindungsenthalpie). Das Anker-Pharmakophor für die Erzeugung der Start-Platzierungen besteht aus zwei Wasserstoffbrücken-Wechselwirkungen, wovon die eine mit der NH-Gruppe von Leu83, und die andere mit der Carbonylgruppe von entweder Glu81 oder Leu83 ausgebildet werden muss.

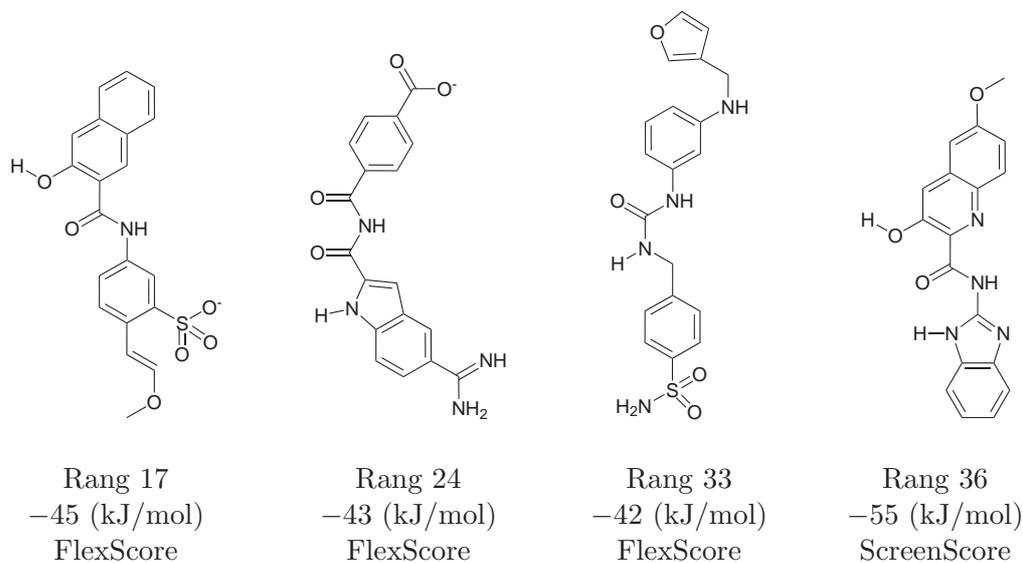


Abbildung 7.4: Repräsentative Ergebnisse für CDK2. Die Moleküle sind analog der vorhergesagten Bindungsmodi orientiert. Zusätzlich sind die Positionen in der finalen Lösungsliste und die dazugehörigen vorhergesagten freien Bindungsenthalpien angegeben.

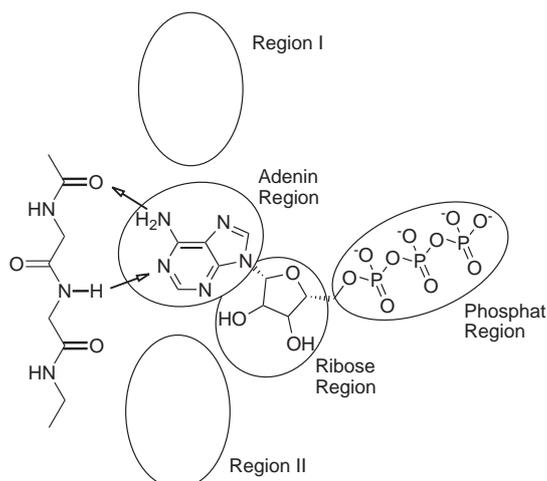


Abbildung 7.5: Bindungsmodus-Modell von Traxler für Kinase-Inhibitoren (Strukturen aus PDB-ID: 1hi1 [Butcher *et al.*, 2001]). Für einen kompetitiven (und selektiven) Inhibitor in der ATP-Bindetasche ist die Ausbildung von Wechselwirkungen in den Regionen I und II entscheidend. Eine höhere Bindungsaffinität kann allein durch Belegung der Phosphat-Region praktisch nicht erreicht werden kann. (Adaptiert aus: [Buijsman, 2004])

sächlich die Regionen *I* und *II*. Demzufolge erweitert FlexNovo die Start-Fragmente bevorzugt entlang des Hinges und nicht senkrecht dazu. Die Phosphat- und Ribose-Regionen sind aus diesem Grund größtenteils unbesetzt. Insgesamt entsprechen die Wechselwirkungsmuster mit der Hinge-Region weitestgehend denen von bekannten Inhibitoren, wobei die meisten erzeugten Moleküle die nahezu exakt planare Geometrie aufweisen, die die meisten aktiven Verbindungen einnehmen.

Abbildung 7.10 zeigt zwei weitere, strukturell unterschiedliche Klassen, wovon das obere Strukturelement der linken Verbindung kommerziell als Naphthol AS erhältlich ist. Aus den Molekülen wird zudem ersichtlich, dass FlexNovo häufig Oxazol-, Thiazol und Imidazolringe als zentrale Strukturelemente verwendet hat, die oft in Kinase-Inhibitoren an ähnlichen Stellen in der Bindetasche des Rezeptors vorhanden sind.

Die erzielten Ergebnisse stellen somit vielversprechende erste Strukturvorschläge dar, die große Ähnlichkeit mit bekannten Inhibitoren aufweisen, strukturell aber dennoch variieren.

Cyclooxygenase-2 Ein weit verbreitetes Strukturelement von COX-2 Inhibitoren ist ein ortho-substituiertes aromatisches Ringsystem, das sich in einer der drei lipophilen Subtaschen des Rezeptors befindet und durch das der Bindungsmodus der Inhibitoren zu einem Großteil bestimmt wird. Demzufolge beinhaltet das Anker-Pharmakophor die Platzierung eines solchen ortho-substituierten Rings in der unter anderem von Leu531, Arg120 und Leu359 gebildeten Subtasche. Abbildung 7.6 zeigt die schematische Darstel-

lung der Bindetasche der verwendeten Kristallstruktur mit den wichtigsten Aminosäuren sowie dem angedeuteten Bindungsmodus des komplexierten Inhibitors SC558.

In diesem Fall konnten nur für etwa 13 Prozent aller Fragmente des FT-Raums und für 20 Prozent aller Fragmente des FTsub-Raums Start-Platzierungen erzeugt werden. Die Scores erreichten dabei in beiden Fällen Werte von bis zu -25 (kJ/mol). Wie auch bei den anderen beide Beispielen dominieren wenige Start-Fragmente die finale Lösungsliste, insbesondere das Motiv eines substituierten Phenyl-Pyrazol-Systems. Der Grund dafür ist der hohe Beitrag der durch die Carboxylatgruppe am Pyrazol-Ring ausgebildeten Wasserstoffbrücken-Wechselwirkungen zu Arg120.

Insgesamt sind die erzeugten Moleküle hydrophiler als die meisten bekannten Inhibitoren. Die Ursache dafür liegt unter anderem in der verwendeten Bewertungsfunktion, die Desolvatationseffekte nur unzureichend berücksichtigt und demnach keine ausreichenden Strafterme für nicht interagierende polare Gruppen enthält. Dieses Phänomen tritt auch im Zusammenhang mit Docking-Experimenten auf. Bei Fragmenträumen ist es allerdings aufgrund der Vielzahl an Kombinationsmöglichkeiten von Fragmenten wesentlich stärker ausgeprägt. Mithilfe des neu eingeführten Platzierungsfilters zur Evaluation der Sättigung der möglichen polaren Wechselwirkungen kann dieses Verhalten zum Teil kompensiert werden.

Obwohl die Verwendung der Carboxylatgruppe im Rahmen eines tatsächlichen Entwurfs einer Leitstruktur für die sehr lipophile Bindetasche von COX-2 aufgrund überwiegend elektrostatischer Kriterien vermieden werden sollte, wurde sie von FlexNovo aus mehreren Gründen ausgewählt. Zum einen benötigt der zugrunde liegende Algorithmus zur Erzeugung der Basis-Platzierungen in der Regel einige wenige gerichtete Wechselwirkungen, um eine niederenergetische Platzierung des Fragments in der Bindetasche berechnen zu können. Zum anderen ist der Score dieser speziellen Gruppe verglichen mit verhältnismäßig lipophilen Start-Fragmenten aufgrund der vielen ausgebildeten Wasserstoffbrücken-Wechselwirkungen wesentlich günstiger. Abbildung 7.7 zeigt solch ein Beispiel.

Mithilfe der integrierten Filterkriterien kann die Verwendung einer solchen Gruppe allerdings auf einfache Weise unterbunden werden. Dafür kann einerseits ein spezifischeres Anker-Pharmakophor definiert werden, in dem nicht nur die Verwendung des entsprechenden Ringsystems gefordert, sondern auch das Auftreten der Carboxylatgruppe ausgeschlossen werden kann. Des Weiteren können entsprechende Fragmente durch Definition geeigneter Substrukturen und Verwendung einer entsprechenden Filterspezifikation von vorne herein von der Erzeugung der Start-Platzierungen ausgeschlossen oder ganz aus dem Fragmentraum entfernt werden.

Abbildung 7.10 zeigt zwei weitere Moleküle, deren vorhergesagter Bindungsmodus nahezu exakt dem der meisten bekannten Inhibitoren entspricht. Beide Lösungen sind auch strukturell eng verwandt mit den aktiven Verbindungen. Der COX-2 Rezeptor stellt für FlexNovo aufgrund seiner geschlossenen Bindetasche und lipophilen Eigenschaften ein besonders anspruchsvolles Design-Szenario dar. Dennoch war der Algorithmus mit dem verwendeten einfachen Anker-Pharmakophor in der Lage, Moleküle zu erzeugen, die die bekannten strukturelle Motive enthalten und alle drei lipophilen Subtaschen des Aktiven Zentrums füllen.

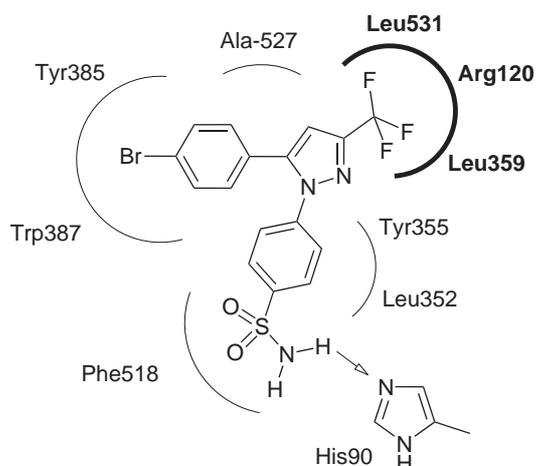
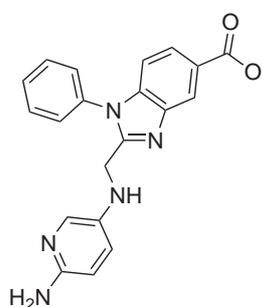


Abbildung 7.6: Schematische Darstellung der Bindetasche von COX-2 (PDB-ID: 6cox [Kurumbail *et al.*, 1996]) zusammen mit dem kokristallisierten Inhibitor SC558 (Orientierung analog des Bindungsmodus). Die vorhergesagte freie Bindungsenthalpie beträgt -16 (kJ/mol). Das Anker-Pharmakophor für die Erzeugung der Start-Orientierung fordert die Platzierung eines *ortho*-substituierten Ringsystems innerhalb einer Kugel von $1,5 \text{ \AA}$ in der Subtasche, die unter anderem von den Aminosäuren Leu531, Arg120 und Leu359 gebildet wird.



Rang 3, -44 (kJ/mol) (ScreenScore)

Abbildung 7.7: Ausgewähltes Ergebnis für COX-2. Das Molekül ist analog des vorhergesagten Bindungsmodus orientiert. Zusätzlich ist die Positionen in der finalen Lösungsliste und die dazugehörige vorhergesagte freie Bindungsenthalpie angegeben.

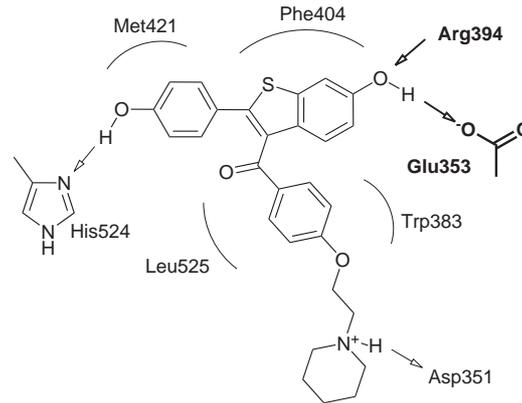


Abbildung 7.8: Schematische Darstellung der Bindetasche von ER (PDB-ID: 1err [Brzozowski *et al.*, 1997]) zusammen mit dem kokristallisierten Inhibitor Raloxifen (Orientierung analog des Bindungsmodus). Die vorhergesagte freie Bindungsenthalpie für Raloxifen beträgt -28 (kJ/mol). Das Anker-Pharmakophor für die Erzeugung der Start-Platzierungen besteht aus einer Wasserstoffbrücken-Wechselwirkung zu Arg394 oder Glu353.

In diesem Zusammenhang sind zwei weitere Charakteristika von Interesse. Zum einen sind Halogene nicht in dem verwendeten Wechselwirkungsmodell enthalten, sodass der Algorithmus Halogen-Bindungseffekte nicht entsprechend berücksichtigen kann. Demnach werden die betroffenen höheren Halogene lediglich durch ihr van-der-Waals-Volumen charakterisiert. Aus diesem Grund sind halogenierte aromatische Systeme praktisch nicht in der Lösungsmenge vertreten. Zum anderen war im Gegensatz zu allen anderen Fallbeispielen die Qualität der Ergebnisse des FTsub-Raums wesentlich schlechter als die des FT-Raums, weshalb die dargestellten Moleküle ausschließlich aus den Ergebnissen des FT-Raums entnommen wurden.

Estrogenrezeptor Die meisten der Agonisten und Antagonisten des Estrogenrezeptors bilden Wasserstoffbrücken zu den Aminosäuren Arg394 und Glu353 aus. Das verwendete Anker-Pharmakophor für die Erzeugung der Start-Platzierungen war weniger strikt. Darin wurde nur die Ausbildung einer der beiden Wasserstoffbrücken-Wechselwirkungen gefordert. Die wichtigsten Aminosäuren sowie ein Umriss der Bindetasche der verwendeten Kristallstruktur sind in Abbildung 7.8 schematisch dargestellt, zusammen mit dem Bindungsmodus des komplexierten Inhibitors Raloxifen.

Aufgrund des relativ unspezifischen Anker-Pharmakophors konnten für ungefähr 63 Prozent aller Fragmente Start-Platzierungen generiert werden. Wie auch bei den anderen Beispielen reichen die erzielten Scores dabei von kleinen positiven Werten bis zu -40 (kJ/mol) für die größten Start-Fragmente. Einige wenige Fragmente dominieren auch in diesem Fall die finale Lösungsliste. Am häufigsten ist dabei ein Pyrimidyl-Hydrazin-Fragment vertreten, das in allen dargestellten Molekülen auftritt. Der vorhergesagte Energiebeitrag für die Hydrazygruppe ist dabei aufgrund der Vielzahl an

7 Strukturbasierte Suche in Fragmenträumen

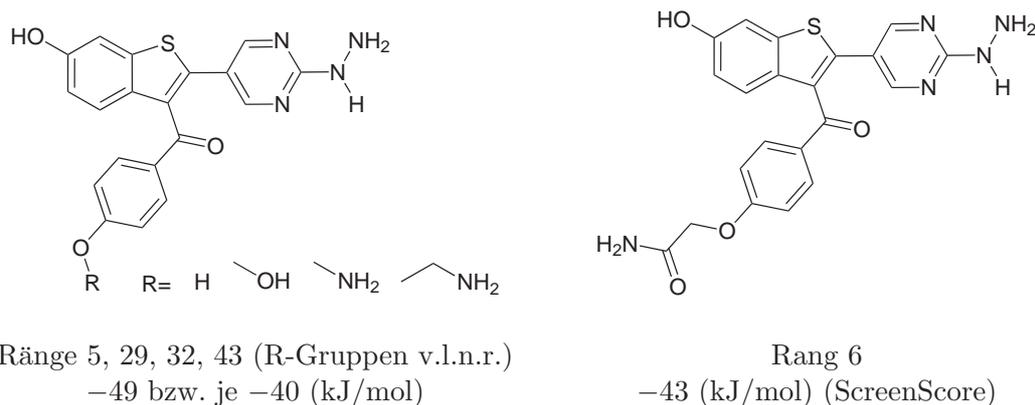


Abbildung 7.9: Repräsentative Ergebnisse für ER. Die Moleküle sind analog der vorhergesagten Bindungsmodi orientiert. Zusätzlich sind die Positionen in der finalen Lösungsliste und die dazugehörigen vorhergesagten freien Bindungsenthalpien angegeben.

ausgebildeten Wasserstoffbrücken-Wechselwirkungen wesentlich höher als für eine einzelne phenolische Hydroxylgruppe, wie sie beispielsweise in Raloxifen auftritt. Diese Beobachtung ist eng verwandt zu der im Zusammenhang mit der Carboxylatgruppe im Fall von COX-2 gemachten, wobei an dieser Stelle anstatt elektrostatischer einige pharmakologische Gründe gegen die tatsächliche Verwendung von Hydrazylgruppen sprechen. Durch die Anwendung entsprechender Filterkriterien und eines strikteren Anker-Pharmakophors kann ein solches Motiv jedoch leicht ausgeschlossen werden.

Steroid-ähnliche Fragmente wurden im Rahmen der durchgeführten Rechnungen nicht berücksichtigt, d.h., sie waren weder Bestandteil des verwendeten Fragmentraums noch konnten sie während des Aufbauprozesses entstehen. Aus diesem Grund war FlexNovo nicht in der Lage, analoge Verbindungen zu den entsprechenden in der zusammengestellten Menge enthaltenen Inhibitoren zu erzeugen. Aufgrund der verwendeten chemischen und algorithmischen Konzepte und Modelle zur Erzeugung und Verknüpfung von Fragmenten können derartige strukturelle Motive nicht analog zu den experimentell üblichen Ringschlussreaktionen gebildet werden. Da diese Motive sich strukturell sehr stark von typischen Leitstrukturen abheben sowie das zentrale strukturelle Element vieler hormoneller Verbindungen darstellen, sind sie im Rahmen der Leitstruktursuche eher uninteressant.

Insgesamt hat FlexNovo die spezifischen strukturellen Motive nicht-steroider Agonisten und Antagonisten gut reproduziert. Alle erzeugten Moleküle enthalten mindestens zwei aromatische Einheiten, die durch eine Einfachbindung miteinander verknüpft sind und den gesamten Bereich der Bindetasche, ausgehend von den Aminosäuren Arg394 und Glu353 bis hin zu His524, ausfüllen. Neben dem Pyrimidinring des Start-Fragments tritt als zweites Ringmotiv häufig ein substituiertes Benzothiophen oder Indol-ähnliche Ringsysteme auf. Dabei unterscheidet sich die Orientierung des Benzothiophens im Vergleich zu Raloxifen, was hauptsächlich auf die Platzierung der Hydrazylgruppe zurückzuführen

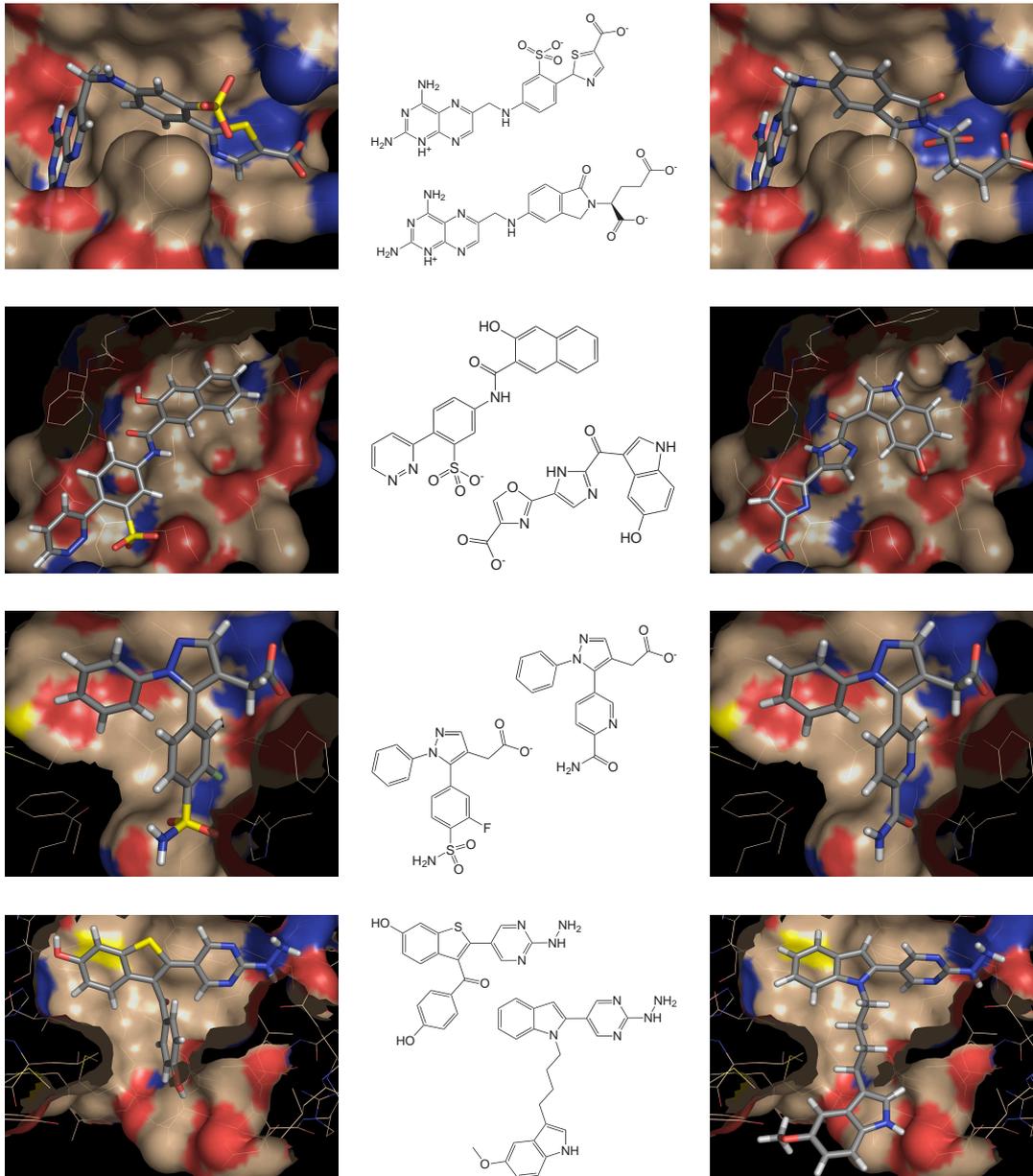


Abbildung 7.10: Repräsentative Ergebnisse der FlexNovo-Rechnungen für DHFR, CDK2, COX-2 und ER (von oben nach unten). Die dreidimensionalen Darstellungen zeigen den vorhergesagten Bindungsmodus der Lösungen zusammen mit der Conolly-Oberfläche der unteren Hälfte der Bindetasche. Oberflächen-Segmente von Heteroatomen sind farbkodiert dargestellt. Die linke Spalte zeigt jeweils ein ausgewähltes Ergebnis des FTsub-Raums, die rechte Spalte eines des FT-Raums (bei COX-2 stammen beide Moleküle aus dem FT-Raum).

ist. Nichtsdestotrotz tritt die spiegelbildliche Orientierung der entsprechenden Substruktur in Raloxifen auch bei Docking-Experimenten auf, da sie gemäß der Bewertungsfunktion energetisch sehr ähnlich sind. Die oben beschriebenen Motive werden häufig durch eine Alkylkette oder einen dritten aromatischen Ring ergänzt. Abbildung 7.9 zeigt einige Beispiele, die die wesentlichen strukturellen Merkmale enthalten.

Die vorhergesagten Bindungsmodi zweier weiterer Moleküle sind in Abbildung 7.10 enthalten. Dabei fällt insbesondere auf, dass FlexNovo einen Indolring außerhalb der eigentlichen Bindetasche platziert hat, dessen NH-Gruppe exakt die gleiche Position hat und dadurch eine Wechselwirkung mit Asp351 ausbildet, wie das auch für die NH-Gruppe des Piperidinrings von Raloxifen der Fall ist. Obwohl sich zumindest die beiden letzten Motive elektronisch und chemisch deutlich unterscheiden, ist es FlexNovo in dieser ersten Studie trotz des großen Suchraums gelungen, aus einer großen Menge an Teillösungen sowohl das Benzothiophen-Ringsystem als auch die meisten anderen wesentlichen strukturellen Merkmale der Antagonisten zu reproduzieren.

7.2.4.2 Ähnlichkeit zu bekannten Inhibitoren

Im Zuge der Evaluation der erzeugten Moleküle wurden neben der oben beschriebenen visuellen Evaluierung zusätzlich automatisierte Ähnlichkeitsanalysen zu den Inhibitoren in den jeweiligen Zusammenstellungen durchgeführt. Der Fokus der Analysen lag dabei auf der Identifikation von Molekülen, die eine hohe topologische Ähnlichkeit zu bekannten Inhibitoren aufweisen.

Aus diesem Grund wurde dafür das FTree-Ähnlichkeitsmaß verwendet [Rarey & Dixon, 1998]. Dabei wurden für jedes Fallbeispiel alle paarweisen Ähnlichkeiten der Inhibitoren aus den Zusammenstellungen zu den Molekülen aus den jeweiligen Lösungslisten berechnet. Für jeden Inhibitor wurde das Molekül mit dem höchsten Ähnlichkeitswert aus der dazugehörigen Ergebnismenge bestimmt. Abbildung 7.11 enthält einige repräsentative Beispiele und zeigt paarweise überlagerte Moleküle und Inhibitoren in ihren vorhergesagten bzw. modellierten Bindungsmodi.

Zunächst muss darauf hingewiesen werden, dass für die COX-2-Ergebnisse im Rahmen dieser Analysen keine Moleküle identifiziert werden konnten, die sich hinreichend von den bereits gezeigten Ergebnissen unterscheiden. Aus diesem Grund enthält Abbildung 7.11 keine Beispiele für COX-2. Die Ursache dafür kann in der sehr kleinen Menge erzeugter Lösungen liegen und steht unter Umständen auch in Zusammenhang mit der relativ kleinen Anzahl sehr ähnlicher aktiver Verbindungen, die für den Vergleich herangezogen wurden.

Bei Betrachtung der Ergebnisse für DHFR fällt auf, dass FlexNovo nicht nur das Pteridin-Ringsystem in der Lösungsmenge reproduziert hat, sondern auch den kleineren identisch substituierten Pyrimidinring, der in den meisten der verwendeten Inhibitoren anzutreffen ist. Aufgrund der sehr spezifischen Wechselwirkungen dieser beiden Motive ist der Bindungsmodus insgesamt sehr ähnlich zu dem der dazugehörigen Inhibitoren.

Die Ergebnisse für CDK2 zeichnen sich insbesondere dadurch aus, dass das Wechselwirkungsmuster mit der Hinge-Region des Aktiven Zentrums von FlexNovo sehr gut reproduziert werden konnte. Obwohl die strukturellen Motive stärker von denen der In-

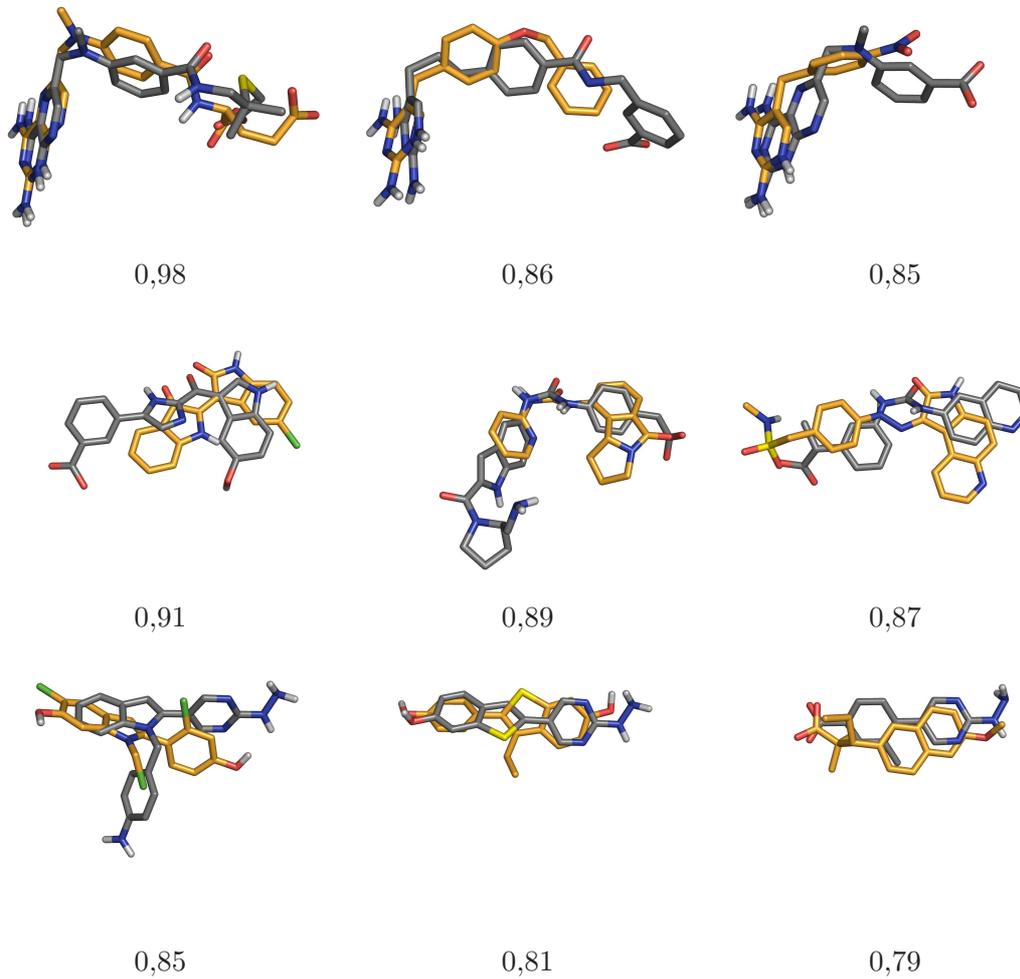


Abbildung 7.11: FTree-Ähnlichkeitsanalysen der FlexNovo-Ergebnisse: Überlagerte Paare von Molekülen der FlexNovo-Ergebnislisten (grau) und bekannten Inhibitoren (orange) für die Rezeptoren DHFR, CDK2, ER (von oben nach unten). Die Geometrien beider Strukturen entsprechen den vorhergesagten Bindungsmodi. Aus Gründen der Übersichtlichkeit wurden alle Wasserstoffatome entfernt, die nicht an Heteroatome gebunden sind. Die Ähnlichkeitswerte sind unterhalb jedes Paares angegeben. Die gezeigten Moleküle sind die Repräsentanten der FlexNovo-Lösungsmenge, die den höchsten FTree-Ähnlichkeitswert zu dem jeweiligen Inhibitor aufweisen.

hibitoren abweichen, enthalten die Moleküle sinnvolle strukturelle Variationen, wovon viele Motive ebenso in anderen aktiven Verbindungen vorkommen.

Auch im Fall des Estrogenrezeptors wird die große topologische Ähnlichkeit der erzeugten Moleküle und der Inhibitoren deutlich. Neben einigen weiteren strukturellen Variationen ist besonders die Konservierung des Bindungsmodus hervorzuheben. Die Ähnlichkeitswerte sind nicht ganz so hoch wie bei den beiden anderen Beispielen. Allerdings sind die erzeugten Moleküle deutlich kleiner, sodass der Effekt auch auf eine geringe Größenabhängigkeit des verwendeten Ähnlichkeitsmaßes zurückzuführen ist.

Insgesamt wurden für die Fallbeispiele durchgängig Moleküle mit hohen bis sehr hohen Ähnlichkeitswerten identifiziert, die in ähnlichen bis identischen Bindungsmodi vorliegen.

7.2.4.3 Profil der physikochemische Eigenschaften

Neben den erzeugten Strukturen selbst sind auch die physikochemischen Eigenschaften der generierten Moleküle von zentralem Interesse. Für die Erzeugung eines physikochemischen Profils der erhaltenen Lösungen wurden für jedes Fallbeispiel zunächst die Lösungsmengen beider Fragmenträume vereinigt. Daraus wurden dann die Verteilungen der Werte für die sechs physikochemischen Eigenschaften bestimmt, die auch in der Filterspezifikation während des Aufbauprozesses verwendet worden sind. Für die Zusammenstellungen der jeweiligen Inhibitoren wurde ein analoges Profil erstellt. Beide Profile wurden auf die Größe des zugrunde liegenden Datensatzes normiert.

Die Verteilungen jeder Eigenschaft für jedes Anwendungsbeispiel werden im Folgenden miteinander verglichen. Abbildung 7.12 enthält die jeweiligen Histogramme, die aus Gründen der Anschaulichkeit als geglättete Liniendiagramme dargestellt sind. Die Ergebnisse für alle Fallbeispiele sind dabei in einer Grafik zusammengefasst.

Anhand der Grafiken können etliche Fälle identifiziert werden, bei denen die von FlexNovo erzeugten Moleküle nahezu identische Verteilungen der dazugehörigen Werte aufweisen wie die zum Vergleich herangezogenen Inhibitoren. Dies trifft insbesondere auf die Verteilungen der $\log P$ Werte für CDK2 und ER, die Verteilungen der Anzahl rotierbarer Bindungen für DHFR, COX-2 und ER sowie die Verteilung der Anzahl der Ringe für COX-2 zu. Bei einigen weiteren Profilen der von FlexNovo erzeugten Lösungsmengen besteht eine große Ähnlichkeit in der Form der Verteilungen zu denen der Inhibitoren, lediglich der dazugehörige Wertebereich ist etwas verschoben.

Besonders hervorzuheben ist in diesem Zusammenhang die Tatsache, dass FlexNovo spezielle Charakteristika der Rezeptoren und der dazugehörigen Inhibitoren reproduzieren konnte. So leiten sich aus den aktiven Verbindungen des Estrogenrezeptors Verteilungen ab, die für das Molekulargewicht den berechneten $\log P$ und die Anzahl rotierbarer Bindungen eine klare Doppelmaximum-Struktur aufweisen. Dies ist darauf zurückzuführen, dass die Zusammenstellung sowohl aus Agonisten als auch aus Antagonisten des Estrogenrezeptors besteht.

Die entsprechende Doppelmaximum-Struktur ist ebenso in den Verteilungen der von FlexNovo erzeugten Moleküle zu erkennen, vor allem bei der Anzahl der rotierbaren Bindungen. Bei höheren Werten des $\log P$ fällt zwar das zweite Maximum signifikant kleiner aus als das bei den Inhibitoren, allerdings liegt die Ursache dafür in den fehlenden

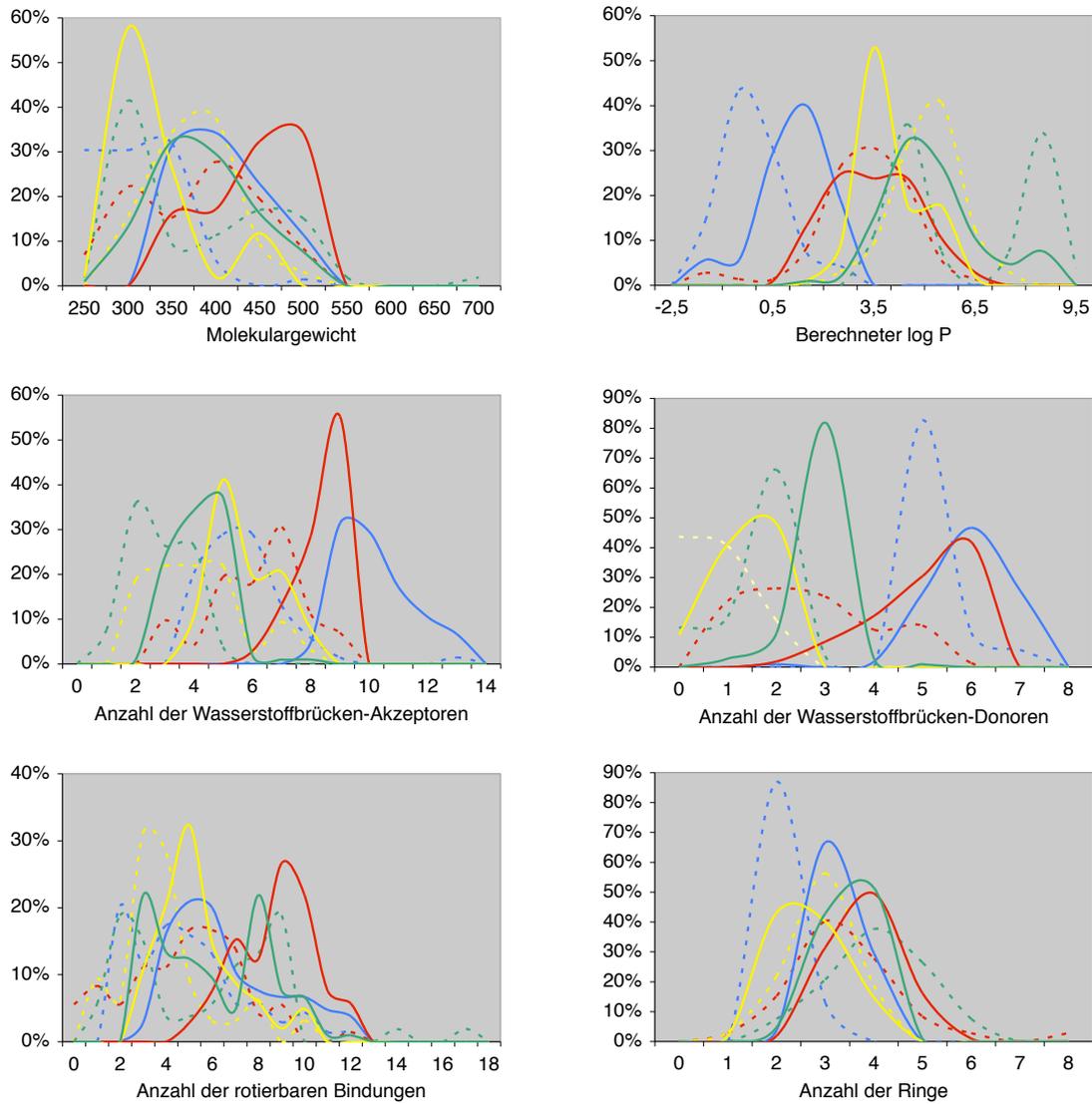


Abbildung 7.12: Verteilungen der physikochemischen Eigenschaften der Ergebnisse (durchgezogene Linien) und den Zusammenstellungen der bekannten Inhibitoren (unterbrochene Linien). Die Histogramme sind aus Gründen der Anschaulichkeit als geglättete Liniendiagramme dargestellt. Jedes einzelne Bild enthält die Verteilungen für die vier Rezeptoren DHFR (blau), CDK2 (rot), COX-2 (gelb) und ER (grün).

Steroid-Strukturen in der erzeugten Lösungsmenge. Bei den Verteilungen der Lösungen für COX-2 ist ein ähnliches Phänomen zu beobachten, allerdings sind die Effekte in diesem Zusammenhang insgesamt weniger stark ausgeprägt.

Zum Teil lassen sich bei der Analyse der Verteilungen einige generelle Charakteristika der Lösungsmengen identifizieren, die bereits bei der Ergebnisdiskussion deutlich wurden. So tendiert FlexNovo in den meisten Fällen dazu, Moleküle zu generieren, die durchschnittlich etwas größer und hydrophiler sind als die vergleichbaren Inhibitoren. Dies lässt sich nicht nur anhand der Verteilung des Molekulargewichts erkennen, sondern auch an denen der Anzahl der Wasserstoffbrücken-Akzeptoren und Donoren.

Generell zeigen alle Grafiken sinnvolle Verteilungen und Wertebereiche für die betrachteten physikochemischen Eigenschaften. Insbesondere konnte durch die Vielzahl verwendeter Filterkriterien erreicht werden, dass FlexNovo trotz der geringen Anzahl erzeugter Lösungen in der Lage war, in den erzeugten Molekülen ein großes Spektrum unterschiedlicher physikochemischer Eigenschaften abzudecken.

7.2.4.4 Diversität der Lösungsmengen

Wie bereits in den vorangegangenen Kapiteln beschrieben, ist die Aufgabe eines de-novo-Design-Verfahrens nicht nur die Erzeugung einer Menge von Molekülen mit möglichst großer Komplementarität zur jeweiligen Bindetasche, sondern ebenso die Realisierung ausreichender struktureller Diversität innerhalb der erzeugten Lösungsmenge. Für die folgenden Analysen wurden – wie bereits zuvor – die vereinigten Ergebnismengen beider Fragmenträume verwendet.

Die Quantifizierung der Diversität der jeweiligen Lösungsmengen erfolgt über die Bestimmung der maximal gemeinsamen (verbundenen) Teilgraphen (MCS) [Raymond & Willet, 2002b] aller enthaltenen Moleküle untereinander, wobei dafür ausschließlich die Schweratome berücksichtigt werden. Mit den so bestimmten Werten wird ein hierarchisches Clustering unter Verwendung eines complete-linkage-Ansatzes durchgeführt. Dabei wird die Mindestgröße des MCS auf zehn Schweratome festgelegt, um zwei Moleküle demselben Cluster bzw. demselben Molekülgerüst zuzuordnen.

Aus den Ergebnissen des Clusterings kann nun die Diversität der Molekülgerüste in den Molekülen der Lösungsmenge abgeleitet werden. Die Anzahl der Cluster für einen gegebenen MCS-Wert entspricht dabei der Anzahl der unterschiedlichen Molekülgerüste in der Lösungsmenge. Die Ergebnisse der Analysen sind in Tabelle 7.2 zusammengefasst. Darin ist die Anzahl der erhaltenen Cluster für drei MCS-Grenzwerte angegeben.

Die Anzahl der Cluster erhöht sich für größere Werte des minimal geforderten MCS. Nach der obigen Definition enthalten die zusammengefassten Ergebnismengen aller Fallbeispiele jeweils mindestens sechs verschiedene strukturelle Klassen, wenn man einen MCS von mindestens zehn Schweratomen fordert, was z.B. dem Pteridin-Ringsystem von Methotrexat entspricht. Erhöht sich der MCS-Wert auf 15, so ergeben sich zwischen 14 und 18 unterschiedliche molekulare Gerüste.

Demzufolge war FlexNovo trotz der relativ kleinen Anzahl an erzeugten Molekülen in der Lage, für die einzelnen Fallbeispiele eine ausreichend hohe Zahl an unterschiedlichen Grundgerüsten zu generieren.

Tabelle 7.2: Diversitätsanalyse der Ergebnisse

	DHFR	CDK2	COX-2	ER
Anzahl Cluster mit MCS \geq 10	8	6	7	7
Anzahl Cluster mit MCS \geq 15	17	14	18	16
Anzahl Cluster mit MCS \geq 20	29	31	46	31

a Die Werte entsprechen der Anzahl unterschiedlicher Cluster, die in der jeweiligen Menge erzeugter Moleküle enthalten sind, wenn eine Mindestgröße des maximal verbundenen gemeinsamen Teilgraphen (MCS) von 10, 15 oder 20 Schweratomen zwischen allen Mitgliedern eines Clusters gefordert wird.

7.2.4.5 Synthetische Zugänglichkeit

Die synthetische Zugänglichkeit der vorhergesagten Moleküle ist ein zentraler Aspekt bei der Beurteilung der Ergebnisqualität und somit eine der wichtigsten Herausforderungen für ein de-novo-Design-Verfahren. Für eine objektive und quantitative Evaluation der synthetischen Zugänglichkeit der Ergebnisse aus den Fallstudien wurde das Programm Sylvia verwendet [Boda *et al.*, 2007].

Die erzeugten Moleküle werden dabei sowohl hinsichtlich ihrer strukturellen Komplexität bewertet, als auch mithilfe eines umfassenden retrosynthetischen Regelwerks zerlegt, und die jeweiligen Fragmente hinsichtlich ihrer Ähnlichkeit zu kommerziell verfügbaren Ausgangsmaterialien beurteilt. Als Ergebnis liefert Sylvia eine Zahl zwischen null und zehn, wobei höhere Werte einer geringeren Zugänglichkeit entsprechen bzw. umgekehrt eine höhere molekulare Komplexität widerspiegeln. Demzufolge entspricht ein Wert von null einem trivialen Molekül wie z.B. Methanol, ein Wert von zehn hingegen einer äußerst komplexen organischen Verbindung wie z.B. einem marinen Naturstoff.

Für die folgenden Analysen wurden – wie bereits zuvor – die vereinigten Ergebnismengen beider Fragmenträume verwendet. Für die entsprechenden Zusammenstellungen der jeweiligen Inhibitoren wurde ein analoges Profil erstellt. Beide Profile wurden auf die Größe des zugrunde liegenden Datensatzes normiert. Abbildung 7.13 enthält die entsprechenden Histogramme.

Zunächst lässt sich anhand der Grafiken erkennen, dass für alle von FlexNovo erzeugten Moleküle Verteilungen für die quantifizierte synthetische Zugänglichkeit ermittelt wurden, die hohe Ähnlichkeit zu denen der bekannten Inhibitoren aufweisen. Insbesondere trifft dies auf die Ergebnisse für CDK2 und COX-2 zu. Generell liegen alle ermittelten Werte im Bereich zwischen 3,5 und 6,0, was für durchschnittliche organische Moleküle erwartungsgemäß der Fall ist.

Interessanterweise ist lediglich für DHFR eine signifikante Abweichung der Werte der Ergebnisse von denen der Inhibitoren zu erkennen. Dieser Unterschied ist hauptsächlich darauf zurückzuführen, dass fast alle von FlexNovo erzeugten Moleküle das substituierte Pteridin-Ringsystem aufweisen, das von der Bewertungsfunktion aufgrund des wesentlich höheren energetischen Beitrags gegenüber dem identisch substituierten Pyrimidin-Ring

7 Strukturbasierte Suche in Fragmenträumen

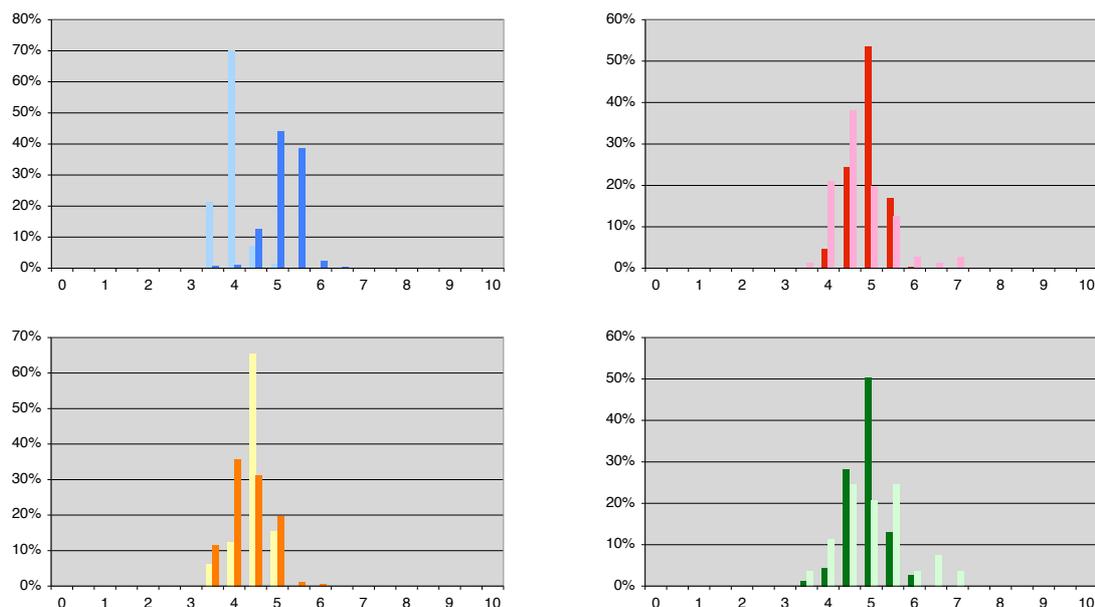


Abbildung 7.13: Verteilungen der Werte der synthetischen Zugänglichkeit für die FlexNovo-Ergebnisse (dunkel) und den Zusammenstellungen der bekannten Inhibitoren (hell). Die Histogramme zeigen die Verteilungen für die vier Rezeptoren DHFR (blau), CDK2 (rot), COX-2 (gelb) und ER (grün).

bevorzugt wurde. Im Gegensatz dazu weist die Zusammenstellung der Inhibitoren fast ausschließlich den entsprechenden kleineren Pyrimidin-Ring auf. Da in den von FlexNovo erzeugten Moleküle außerdem häufig Fragmente verwendet wurden, die im Vergleich zu dem Großteil der Inhibitoren zusätzliche Ringschlüsse enthalten, ist eine Abweichung in Höhe von ein bis zwei Einheiten bezüglich des Sylvania-Scores in diesem Zusammenhang zu erwarten und leicht nachvollziehbar.

Im Fall des Estrogenrezeptors treten für die Steroidstrukturen in der Zusammenstellung der Inhibitoren, wie bereits bei den vorherigen Ergebnisanalysen, einige Besonderheiten auf. Diese Strukturen werden von Sylvania aufgrund der Größe des Ringsystems und der Vielzahl an Stereozentren tendenziell mit höheren Scores bewertet. In der entsprechenden Verteilung der Werte ist dies entsprechend zu erkennen. Da solche Strukturen in den Berechnungen mit FlexNovo nicht berücksichtigt bzw. nicht erzeugt worden sind, können entsprechend erhöhte Werte bei den Molekülen der Ergebnisliste nicht beobachtet werden.

Insgesamt sind unter Berücksichtigung der mit Sylvania erhaltenen Werte alle erzeugten Moleküle synthetisch einfach bis gut zugänglich. Obwohl die jeweilige Strategie zur Herstellung einer speziellen chemischen Verbindung im Einzelfall erarbeitet werden muss, zeigen die erhaltenen Ergebnisse deutlich, dass FlexNovo für unterschiedlichen Szenarien und Targets aus dem verwendeten Fragmentraum chemisch sinnvolle und insbesondere

realistische Moleküle erzeugt hat. Somit konnte FlexNovo neben den vielen weiteren Teilaspekten und Anforderungen an ein de-novo-Design-Verfahren auch die chemische Machbarkeit der vorhergesagten Verbindungen weitestgehend gewährleisten.

7.2.4.6 Generelle Aspekte

Aus der Analyse der Ergebnisse für die vier Fallbeispiele können einige allgemeine Schlüsse gezogen werden. Die von FlexNovo vorhergesagten Bindungsmodi zeichnen sich insbesondere durch die gute Geometrie der Wasserstoffbrücken-Wechselwirkungsmuster aus, was auch für die Docking-Ergebnisse von FlexX zu beobachten ist. Des Weiteren können mithilfe der physikochemischen Filterkriterien sowie der zusätzlichen geometrischen Kriterien für Platzierungen einige gravierende Nachteile der verwendeten Bewertungsfunktion kompensiert werden. Aufgrund der Vielzahl an möglichen Lösungen treten diese Effekte in Zusammenhang mit Fragmenträumen umso stärker in Erscheinung. Diese Kriterien führen aber unabhängig von der Steigerung der Qualität der Ergebnisse auch zur Reduktion der benötigten Rechenzeit. Darauf wird in den Kapiteln 7.2.4.7 und 7.3 näher eingegangen.

In der Regel wird die Lösungsmenge von einigen wenigen strukturellen Motiven dominiert. Dies trifft nicht nur auf die letztendlich resultierenden Moleküle sondern auch auf die erzeugten Zwischenlösungen zu. Aufgrund der Größe des zu durchsuchenden Lösungsraums sind die Unterschiede in den mit der Bewertungsfunktion abgeschätzten Energiebeiträgen der Zwischenlösungen mitunter sehr gering. Auch die Moleküle der finalen Lösungsmenge liegen typischerweise energetisch recht dicht zusammen. Die integrierten Diversitätskriterien stellen ein effektives Mittel dar, um diese Effekte zu kompensieren und eine hinreichende strukturelle Diversität auf allen Ebenen des Aufbauprozesses zu gewährleisten.

FlexNovo profitiert von der Definition möglichst strikter Randbedingungen aufgrund des großen Suchraums in noch größerem Maße, als das bereits bei FlexX der Fall ist. Das trifft sowohl auf die Formulierung der physikochemischen Eigenschaften als auch in besonderer Weise auf die geometrischen Kriterien der Bindetasche zu. Allerdings können zu eng formulierte Bedingungen auch dazu führen, dass möglicherweise interessante Lösungen verloren gehen, wie exemplarisch anhand des Beispiels in Kapitel 7.2.6 gezeigt wird.

7.2.4.7 Statistische Informationen

Die oben beschriebenen Rechnungen für die vier Fallbeispiele wurden auf einem Workstation-Computer mit einem Intel Xeon Prozessor mit 3,0 GHz sowie 2 GB Arbeitsspeicher durchgeführt. Abgesehen von den unterschiedlichen Werten zur Steuerung der Diversität wurden für alle Rechnungen dieselben algorithmischen Parameter verwendet.

Tabelle 7.3 enthält die wichtigsten statistischen Informationen. Die einzelnen Kenngrößen wurden dabei aus allen durchgeführten Experimenten ermittelt, weshalb die Spannen der jeweiligen Werte recht groß sind. Der Einfluss der algorithmischen Parameter auf das

Tabelle 7.3: Statistische Informationen zu den Rechnungen

		FT-Raum	FTsub-Raum
Anzahl enthaltener Fragmente		16301	4350
Präprozessierung (initiales Docking)	[h]	15,2 – 18,5	4,3 – 7,1
Anzahl der evaluierten Teillösungen	[Tsd.]	1202 – 1521	350 – 478
Anzahl der platzierten Teillösungen	[Tsd.]	28 – 62	24 – 62
FlexNovo-Gesamtrechenzeit	[h]	7,0 – 15,5	5,0 – 10,0
FlexNovo-Gesamtspeicherbedarf	[MB]	750	200

Laufzeitverhalten und die Zusammensetzung der Gesamtlaufzeit wird in Kapitel 7.3 beschrieben.

Aus Gründen der Vollständigkeit sind in Tabelle 7.3 auch die Zeiten für die Präprozessierung der Fragmente enthalten. Dabei handelt es sich jedoch um einmalige sequenzielle Docking-Rechnungen, die typischerweise für jede Kombination aus Fragmentraum, Bewertungsfunktion und Rezeptor durchgeführt werden. Die dafür benötigte Rechenzeit hängt in erster Linie von der Anzahl und durchschnittlichen Größe der Fragmente des Fragmentraums ab.

Generell resultiert die Menge des benötigten Arbeitsspeichers sowie die Anzahl der evaluierten Teillösungen bei gleichen algorithmischen Parametern hauptsächlich aus der Größe des verwendeten Fragmentraums. In der Regel erfordert eine Rechnung einige Hundert MB Hauptspeicher. Die Anzahl der erzeugten Zwischenlösungen variiert von mehreren Hunderttausend bis hin zu wenigen Millionen Fragmenten. Die Gesamtrechenzeit hingegen resultiert hauptsächlich aus der Anzahl an tatsächlich platzierten Zwischenlösungen sowie in wesentlich geringeren Maße aus der Menge an evaluierten Teillösungen. Dabei spielt auch die Form des Aktiven Zentrums eine entscheidende Rolle, d.h., für eher kleine, geschlossene Bindetaschen ist die Rechenzeit geringer als für große, offene Bindetaschen. Je strikter die Randbedingungen insgesamt sind, desto weniger Rechenzeit wird vom Algorithmus benötigt. In der Regel kann eine Rechnung in einigen Stunden abgeschlossen werden, bei Abdeckung eines sehr großen Teils des Suchraums werden allerdings mehrere Tage (auf einem einzelnen Prozessor) benötigt.

Für den großen Unterschied zwischen der Anzahl der evaluierten und der Menge an tatsächlich platzierten Teillösungen gibt es zwei Ursachen, die beide auf Kriterien beruhen, die bereits vor der tatsächlichen Erzeugung einer Teillösung überprüft werden können. Zum einen kann ein Großteil potenzieller Fragmente nach Abschätzung des maximal erreichbaren Energiebeitrages aufgrund des k -Greedy-Kriteriums verworfen werden. Zum anderen brauchen viele Teillösungen überhaupt nicht erzeugt zu werden, da sie die physikochemischen Filterkriterien verletzen. Beide Kriterien tragen signifikant zur Reduktion der Rechenzeit bei, wie aus den Werten in Tabelle 7.3 hervorgeht.

Die Wahl der algorithmischen Parameter, d.h., die Beschränkung der Menge an Zwischenlösungen auf 100 sowie die Berücksichtigung von bis zu 60 Platzierungen pro Fragment stellte sich als ein guter Kompromiss zwischen Laufzeit und Qualität der Ergeb-

Tabelle 7.4: Daten und Referenzen zu Thrombin

Thrombin	
PDB-ID der Kristallstruktur	1k22 [Dullweber <i>et al.</i> , 2001]
Anzahl der Inhibitoren	144 [Stahl & Böhm, 1998]
Molekulargewicht	150 – 650
Anzahl Ringe	1 – 6
Anzahl rotierbarer Bindungen	0 – 15
Anzahl H-Brücken Akzeptoren	2 – 12
Anzahl H-Brücken Donoren	1 – 6
Berechneter logP	1.0 – 6.0

nisse heraus. Kleinere Werte für die Anzahl der Zwischenlösungen führten in der Regel zu einer zu geringen Abdeckung des Suchraums und entsprechend schlechteren Ergebnissen, größere hingegen nicht zur signifikanten Verbesserung der erhaltenen Moleküle. Die Reduktion der Menge an Platzierungen führte oft dazu, dass für eigentlich gute Zwischenlösungen in späteren Anbausritten vom Algorithmus keine Platzierungen mehr gefunden wurden. Umgekehrt führte die Erhöhung jedoch nicht zu einer signifikanten Verbesserung der Resultate. Dieser Punkt wird in der detaillierten Analyse des Laufzeitverhaltens in Kapitel 7.3 noch einmal aufgegriffen.

7.2.5 Berücksichtigung einer komplexen Pharmakophor-Definition

Bei den bisher diskutierten Ergebnissen wurde ausschließlich ein Anker-Pharmakophor für die Auswahl der Start-Fragmente und Erzeugung der Start-Platzierungen verwendet. Der Algorithmus von FlexNovo erlaubt jedoch insbesondere auch die Verwendung einer komplexen Pharmakophor-Definition während des inkrementellen Aufbauprozesses von Molekülen.

Ein Großteil der im Zusammenhang der Fallstudien vorgestellten Ergebnisse erfüllt neben dem jeweiligen Anker-Pharmakophor bereits die überwiegende Anzahl potenzieller weiterer Einzelbedingungen einer umfassenderen Pharmakophor-Definition. Aus diesem Grund wurde speziell für die Validierung der Pharmakophor-Funktionalität von FlexNovo ein wesentlich anspruchsvolleres Anwendungsbeispiel ausgewählt.

Dabei handelt es sich um das Enzym Thrombin, das eine zentrale Rolle bei der Blutgerinnung inne hat und somit ebenfalls von großer pharmakologischer Bedeutung ist. Die verwendete Kristallstruktur sowie die physikochemischen Eigenschaften der Zusammensetzung bekannter Inhibitoren sind in Tabelle 7.4 zusammengefasst. Die Daten wurden entsprechend der in Kapitel 7.2.2 beschriebenen Vorgehensweise aufbereitet.

Aus den Inhibitoren wurde mit dem neu erstellten BRICS-Regelwerk analog der in Kapitel 6.1.2 beschriebenen Vorgehensweise ein fokussierter *Thrombin-Raum* erzeugt, der für die folgende Validierung verwendet wurde. Dieser besteht aus 146 Fragmenten und ist, wie auch die BRICS-Fragmenträume, verbunden und prinzipiell unendlich groß.

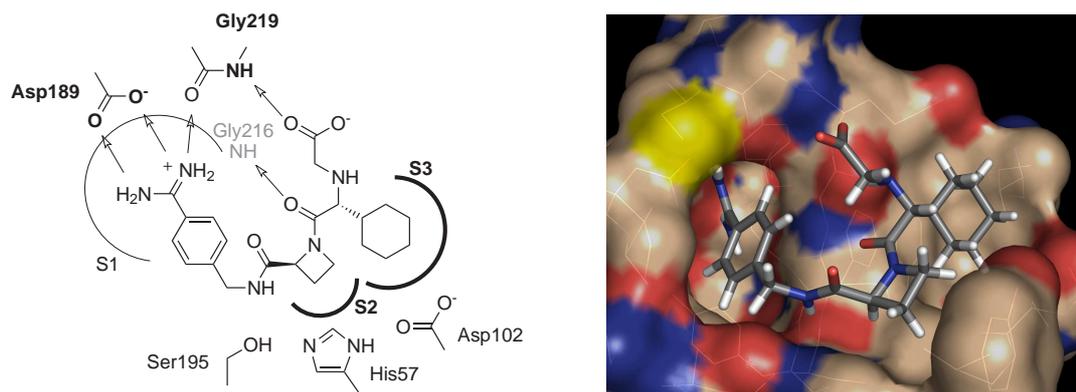


Abbildung 7.14: Schematische Darstellung der Bindetasche von Thrombin (PDB-ID: 1k22 [Dullweber *et al.*, 2001]) zusammen mit dem kokristallisierten Inhibitor Melagatran (Orientierung analog des Bindungsmodus). Die charakteristischen Subtaschen S1, S2 und S3 sind entsprechend gekennzeichnet. Das Anker-Pharmakophor für die Erzeugung der Start-Platzierungen besteht aus der Platzierung einer Benzamidinium-Gruppe in der S1 Subtasche, die zwei Wasserstoffbrücken-Wechselwirkung mit Asp189 ausbildet. Das Pharmakophor während des Aufbauprozesses besteht zusätzlich aus der Platzierung jeweils eines Ringkohlenstoff-Atoms innerhalb der S2 und der S3 Subtasche sowie einer Wasserstoffbrücken-Wechselwirkung zum NH-Donor von Gly219.

Die theoretische Anzahl an zugänglichen Molekülen liegt jenseits von 10^8 , weshalb eine vollständige Enumeration innerhalb der angegebenen physikochemischen Bereiche im Gegensatz zu dem in Kapitel 6.2.2 vorgestellten Fragmentraum nicht möglich ist.

Die Bindetasche von Thrombin weist mehrere spezifische hydrophobe Subtaschen auf, ist abgesehen von der S1 Subtasche sehr flach und erfordert verhältnismäßig große Liganden, die zudem eine verzweigte Struktur aufweisen. Diese Tatsache lässt sich jedoch nicht direkt bei der Auswahl der Start-Fragmente berücksichtigen, wie das etwa für den Estrogenrezeptors oder Cyclooxygenase 2 der Fall ist. Im Fall von Thrombin kann die topologisch wichtige Verzweigung erst nach der Platzierung eines entsprechenden Fragmentes in der S2 Tasche eingefügt werden. Dies wird bei Betrachtung der Bindetasche der verwendeten Kristallstruktur und des Bindungsmodus des komplexierten Inhibitors Melagatran deutlich. Abbildung 7.14 enthält eine dreidimensionale Repräsentation des Aktiven Zentrums sowie eine schematische Darstellung mit den wichtigsten Aminosäuren, in der außerdem die Einzelbedingungen des verwendeten Pharmakophors hervorgehoben sind.

Die Benzamidinium-Gruppe innerhalb der S1 Subtasche des Aktiven Zentrums von Thrombin ist ein weit verbreitetes Strukturmerkmal unter bekannten Inhibitoren, weshalb in der Definition des Anker-Pharmakophors die Verwendung eines solchen Mo-

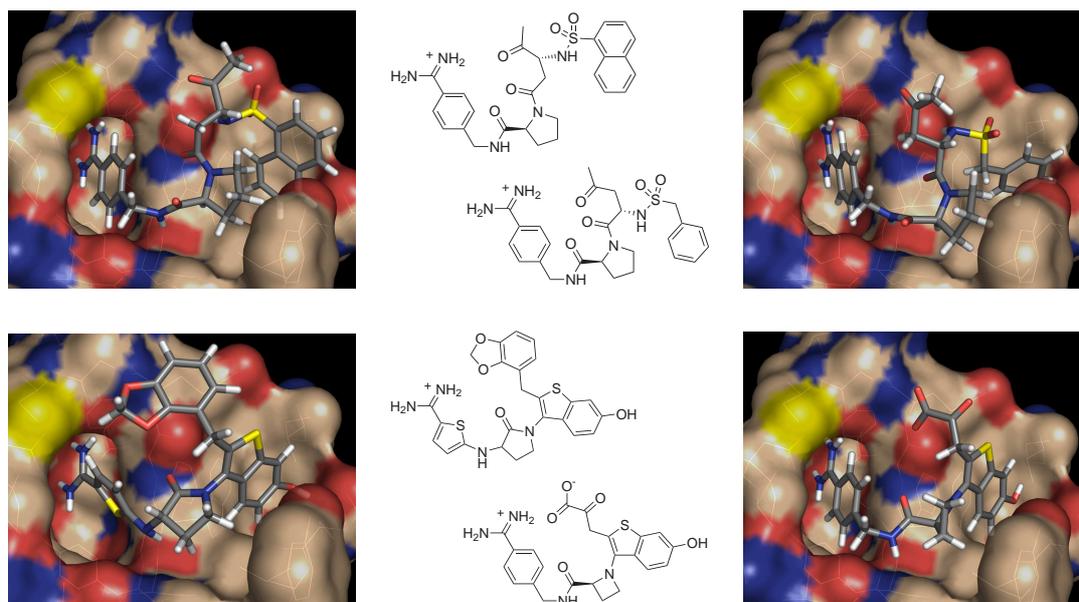


Abbildung 7.15: Repräsentative Ergebnisse der Rechnungen für Thrombin. Die dreidimensionalen Darstellungen zeigen den vorhergesagten Bindungsmodus der jeweiligen Lösung zusammen mit der Conolly-Oberfläche der Binde-tasche. Oberflächen-Segmente von Heteroatomen sind farbkodiert dargestellt. Die linke Spalte zeigt jeweils ausgewählte Ergebnisse für die FlexScore Bewertungsfunktion, die rechte Spalte für die ScreenScore Bewertungsfunktion.

tivs zur Ausbildung von zwei Wasserstoffbrücken zu Asp189 gefordert wurde. 14 Fragmente des Thrombin-Raums konnten damit erfolgreich platziert werden. Neben dem Benzamidinium-Anker sind auch meta-Chlorphenyl-Motive experimentell als spezifisch bindende Fragmente identifiziert worden [Hartshorn *et al.*, 2005]. Die analoge Verwendung dieser Strukturen hat allerdings einen entscheidenden Nachteil, der auch schon im Zusammenhang mit der Erzeugung der Start-Platzierungen bei COX-2 diskutiert worden ist. Aufgrund fehlender Wasserstoffbrücken-Wechselwirkungsmöglichkeiten hat der Platzierungsalgorithmus bei diesen Fragmenten Schwierigkeiten, vielversprechende niederenergetische Orientierungen zu erzeugen. Aufgrund der letztendlich geringen Relevanz für diese spezielle Anwendungsstudie wurde auf die Verwendung von meta-Chlorphenyl-Ankern verzichtet. Allerdings können diese leicht durch manuelle Platzierung oder im Zuge einer nachträglichen Optimierung der erhaltenen Strukturen eingeführt werden.

Die anschließenden Rechnungen wurden mit Standard-Einstellungen durchgeführt. Lediglich die Anzahl der Anbauzyklen wurde auf drei erhöht, um die strukturellen und geometrischen Besonderheiten entsprechend zu berücksichtigen. Das Pharmakophor wurde für den Aufbauprozess um weitere Schlüsselbedingungen erweitert (siehe Abb. 7.14).

Aufgrund der komplexen Pharmakophor-Spezifikation sind die Berechnungen sehr aufwendig, weshalb die Verwendung eines wesentlichen größeren Fragmentraums, wie z.B. BRICS_4k oder FTsub, in diesem Fall praktisch nicht möglich ist (erwartete/geschätzte Rechenzeit \gg ein Monat).

In diesem Zusammenhang soll außerdem erwähnt werden, dass nur 18 der 144 Inhibitoren das bewusst sehr strikt formulierte Pharmakophor erfüllen. Dementsprechend ist die Anzahl der letztendlich erzeugten Moleküle erwartungsgemäß niedrig. So wurden bei den Berechnungen mit der FlexScore Bewertungsfunktion 22, mit der ScreenScore Funktion 27 Lösungen erzeugt, die sich zum Teil strukturell sehr ähnlich sind. Abbildung 7.15 zeigt jeweils zwei repräsentative Beispiele, in denen die am häufigsten verwendeten chemischen Motive enthalten sind.

Insgesamt zeigen die Ergebnisse deutlich, dass FlexNovo mithilfe der umfassenden Pharmakophor-Definition die Topologie typischer Thrombin-Inhibitoren insgesamt erfolgreich rekonstruieren konnte und für die jeweiligen Subtaschen sinnvolle chemische Motive ausgewählt hat. So wurde in den meisten Fällen in der S2 Subtasche ein Pyrrolidin- (vgl. Abb. 7.15 oben) oder Azetidin-Ring (vgl. Abb. 7.15 unten rechts) platziert und über eine Amidbindung mit dem Anker-Fragment in der S1 Subtasche verknüpft. Im Fall des Thiophenyl-Startfragments wurde alternativ ein sekundäres verbrückendes Amin in Verbindung mit einem Lactam-Motiv verwendet (vgl. Abb. 7.15 unten links).

Bei den Ergebnissen der oberen Reihe von Abbildung 7.15 hat der Algorithmus im zweiten Anbauzyklus zunächst die Wasserstoffbrücken-Wechselwirkung mit dem NH-Donor von Gly219 ausgebildet. Dies wurde durch den Anbau einer verzweigten Alkylkette realisiert, die über eine Amidbindung mit dem Stickstoffatom des Heterozyklus verbunden wurde. Dies ist auch bei vielen Inhibitoren der Fall. Im dritten und letzten Anbauzyklus hat der Algorithmus unterschiedliche aromatische Systeme über eine verbrückende Sulfonamid-Funktion angebaut, um die S3 Subtasche auszufüllen und der entsprechenden Pharmakophor-Bedingung gerecht zu werden. Anstelle der Carboxylatgruppe ist in der Nähe von Gly219 eine Acetylgruppe platziert. Die darin enthaltene Methylgruppe resultiert aus der Substitution des nach Abschluss des dritten Anbauzyklus dort befindlichen offenen Link-Atoms. Da dieses aus einer Amidgruppe und nicht aus einem Ester hervorgegangen ist, erfolgt die Substitution entsprechend, was jedoch leicht in der Spezifikation der terminalen Gruppen des Fragmentraums angepasst werden kann.

Bei den Molekülen der unteren Reihe von Abbildung 7.15 hat der Algorithmus zuerst das aromatische Ringsystem in der S3-Tasche platziert. Dafür wurde ein Benzothiophen-Fragment direkt mit den Stickstoff des Heterozyklus verbunden. Im folgenden Anbauschritt wurde dann die Wasserstoffbrücke zu Gly219 ausgebildet. Dafür hat der Algorithmus in den gezeigten Beispielen zum einen die Ketosäuregruppe verwendet und zum anderen ein Acetal-Fragment. Letzteres ist an dieser Stelle nicht optimal positioniert und befindet sich bei Thrombin-Inhibitoren typischerweise in der S3 Subtasche. Eine entsprechende Einschränkung war jedoch nicht Bestandteil der Rechnungen, sodass für ein reales Szenario eine Substitution des Acetals, z.B. mit einem Carboxylat- oder Tetrazol-Fragment, in Betracht gezogen werden sollte.

Obwohl die gezeigten Strukturen trotz der umfangreichen Randbedingungen Opti-

mierungspotenzial aufweisen, sind die Ergebnisse insgesamt überzeugend. Ferner unterstreicht dies erneut, dass es unrealistisch ist zu erwarten, dass eine einzelne vom Algorithmus erzeugte Lösung alle Randbedingungen nahezu optimal erfüllt. Unabhängig davon konnte der Algorithmus alle fünf Einzelbedingungen des komplexen Pharmakophors erfüllen und dadurch alle wesentlichen Merkmale von Thrombin-Inhibitoren reproduzieren. Weiterhin ist es allen gezeigten Strukturen aufgrund der ausgewählten chemischen Motive möglich, neben den in der Pharmakophor-Definition geforderten Wechselwirkungen, zusätzlich auch die bei Melagatran vorhandene Wasserstoffbrücken-Wechselwirkung mit dem NH-Donor von Gly216 auszubilden. Damit sind alle wesentlichen Wechselwirkungen abgedeckt.

In diesem Anwendungsbeispiel war FlexNovo ohne die Verwendung eines vergleichbaren Pharmakophors nicht in der Lage, zufriedenstellende Ergebnisse zu produzieren. Die erhaltenen Ergebnisse unterstreichen deshalb eindeutig die Fähigkeit des Algorithmus, unter Verwendung komplexer Randbedingungen insgesamt überzeugende Moleküle vorherzusagen.

7.2.6 Serendipity

Wie bereits zu Beginn dieser Arbeit erwähnt, spielt der Zufall bei der Entwicklung von Medikamenten und der Auffindung von vielversprechenden Leitstrukturen eine herausragende Rolle. Aus diesem Grund soll an dieser Stelle ein Ergebnis vorgestellt werden, das tatsächlich zufällig im Rahmen der Ergebnisanalysen entdeckt worden ist.

Abbildung 7.16 zeigt das Ergebnis einer Rechnung mit dem FT-Fragmentraum für CDK2, das mit einer der ersten Versionen von FlexNovo erzielt worden ist, in der noch keine Filterkriterien für physikochemische Eigenschaften enthalten, die algorithmischen Parameter jedoch dieselben waren.

Das gezeigte Molekül verletzt die meisten der für die oben beschriebenen Fallstudien verwendeten Filterkriterien. Gleichzeitig entspricht ein Großteil der Struktur sowie der vorhergesagte Bindungsmodus dem von Staurosporin. Bei letzterem handelt es sich um ein ursprünglich aus einem Bakterium isolierten Naturstoff, dessen biologische Wirkungen vom Antipilzmittel bis hin zu blutdrucksenkenden Effekten reicht [Rüegg & Burgess, 1989]. Hauptsächlich wurde Staurosporin jedoch als hochaffiner aber nicht-selektiver Kinase-Inhibitor untersucht.

In dem gezeigten Beispiel liegt Staurosporin zwar in einer anderen Kristallstruktur (PDB-ID: 1qpd [Zhu *et al.*, 1999]) als das von FlexNovo erzeugte Ergebnis vor, es handelt sich aber bei beiden Proteinen um CDK2. Nichtsdestotrotz ist es in Anbetracht der Größe des zugrunde liegenden Suchraums erstaunlich, dass das gezeigte Molekül Bestandteil der finalen Lösungsliste war. Obwohl es zunächst nicht als potenzielle Leitstruktur in Frage zu kommen scheint, hat es gleichzeitig große strukturelle Ähnlichkeit mit dem Inhibitor CEP-1347 von Cephalon und Lundbeck [Saporito *et al.*, 2002], das zur Behandlung der Parkinson'schen Krankheit verwendet werden sollte. Inzwischen sind die klinischen Studien in Phase II/III aufgrund schlechter Wirkung jedoch eingestellt worden, allerdings nicht aufgrund toxischer Eigenschaften.

Obwohl die Verwendung einer derart komplexen und großen Leitstruktur sicherlich

7 Strukturbasierte Suche in Fragmenträumen

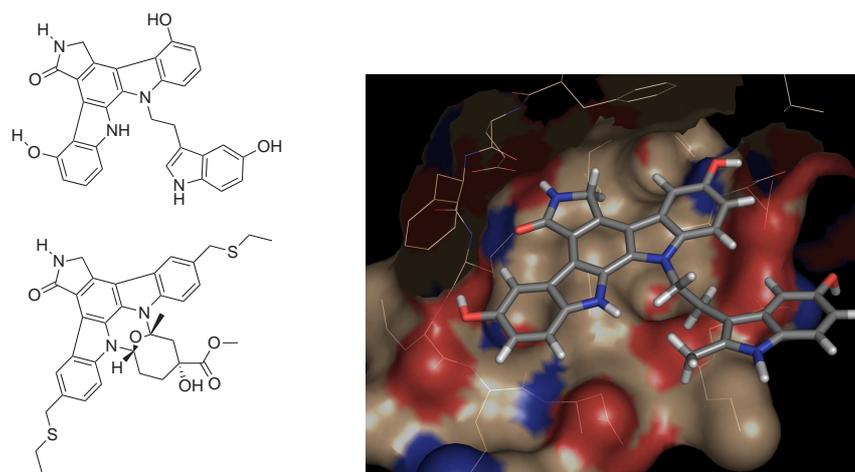


Abbildung 7.16: *Serendipity* im strukturbasierten molekularen Design. Links: Das oben dargestellte Molekül resultiert aus einer der ersten Rechnungen mit FlexNovo (Rang 43 mit -61 (kJ/mol) vorhergesagter freier Bindungsenthalpie). Bei der unten dargestellten Verbindung handelt es sich um CEP-1347 von Cephalon und Lundbeck [Saporito *et al.*, 2002], die zur Behandlung der Parkinson'schen Erkrankung verwendet werden sollte. Rechts: Vorhergesagter Bindungsmodus des von FlexNovo erzeugten Moleküls. Die untere Hälfte des Aktiven Zentrum von CDK2 (PDB-ID: 1di8 [Shewchuk *et al.*, 2000]) ist mit der Conolly-Oberfläche dargestellt, die Oberflächenanteile von Heteroatomen sind farbkodiert.

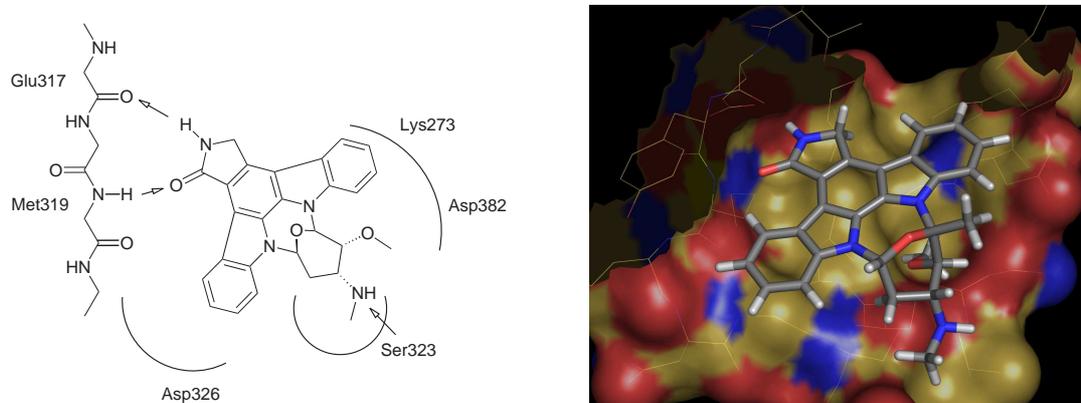


Abbildung 7.17: Darstellung der Bindetasche und des Bindungsmodus von Staurosporin (PDB-ID: 1qpd [Zhu *et al.*, 1999]). Die vorhergesagte freie Bindungsenthalpie beträgt -28 (kJ/mol). Die untere Hälfte des Aktiven Zentrum ist mit der Conolly-Oberfläche dargestellt, die Oberflächenanteile von Heteroatomen sind farbkodiert.

nicht erstrebenswert ist, verdeutlicht dieses Beispiel dennoch eindrucksvoll, dass durch zu strikte Filterkriterien potenziell vielversprechende Lösungen schnell verloren gehen können bzw. erst gar nicht erzeugt werden. Die verwendeten physikochemischen und geometrischen Bedingungen für die Auswahl der Start-Fragmente und die Erzeugung der Zwischenlösungen, sowie die Spezifikation der Einzelbedingungen für das Pharmakophor sind deshalb mit Vorsicht zu verwenden.

Unabhängig davon ist eine starke Fokussierung des Lösungsraum dennoch wünschenswert, da die Menge potenzieller Moleküle selbst bei Verwendung strikter Filterkriterien noch sehr groß ist.

7.3 Laufzeitverhalten

Die Untersuchungen des Laufzeitverhaltens von FlexNovo wurden auf einem Workstation-Computer mit einem Intel Xeon Prozessor mit 3,0 GHz sowie 2 GB Arbeitsspeicher durchgeführt. Die Berechnungen wurden exemplarisch am Beispiel der Dihydrofolat-Reduktase unter Verwendung der in Tabelle 7.1 beschriebenen Eingabedaten und der in Kapitel 7.1 vorgestellten Parametrisierung durchgeführt. Für die Berechnungen wurden sowohl die FlexScore als auch die ScreenScore Bewertungsfunktion verwendet. Im Gegensatz zu den oben diskutierten Fallstudien kamen in diesem Zusammenhang die Fragmenträume BRICS_4k und BRICS_9k zum Einsatz.

Neben der Analyse des Speicherbedarfs und der Gesamtlaufzeit des Programms wurden insbesondere die Laufzeit-Anteile der einzelnen Teilberechnungen des Aufbaualgorithmus bestimmt. Die Werte wurden in Abhängigkeit von der Größe des Fragmentraums, der Menge an Teillösungen, die für den nächsten Anbauschritt berücksichtigt werden, der Diversität der (Teil-)Ergebnisse sowie der Verwendung von Pharmakophor-Randbedingungen während des Aufbauprozesses ermittelt. Für letzteres wurde das in Abbildung 4.8 vorgestellte Pharmakophor herangezogen. Des Weiteren wurden Statistiken über die jeweilige Anzahl der prozessierten, gefilterten und tatsächlich generierten Teillösungen und Platzierungen erstellt.

Bevor die Ergebnisse im Detail diskutiert werden, sollen einige generelle Charakteristika vorweggenommen werden. So hat die verwendete Bewertungsfunktion erwartungsgemäß einen entscheidenden Einfluss auf die Laufzeit des Algorithmus. Allerdings entsteht dadurch lediglich eine Verschiebung zu höheren bzw. niedrigen Gesamtlaufzeiten. Die relativen Anteile der einzelnen Teilberechnungen ändern sich dadurch nicht signifikant. Aus diesem Grund sind im Folgenden nur die Ergebnisse der FlexScore Bewertungsfunktion dargestellt. Bei den durchgeführten Experimenten nahmen Berechnungen mit der ScreenScore Funktion durchschnittlich etwa 30 Prozent mehr Zeit in Anspruch.

Die Parameter zur Steuerung der Diversität der Liste der Teillösungen sowie der finalen Ergebnisse haben nur einen sehr geringen Einfluss auf das Laufzeitverhalten und den Speicherbedarf. Eine höhere Diversität führt typischerweise dazu, dass eine größere Anzahl an Teillösungen prozessiert wird, wodurch die Gesamtlaufzeit geringfügig steigt. Dies tritt jedoch erst bei sehr großen Werten für die Anzahl der berücksichtigten Teillösungen auf ($k_F \gg 200$) und die Unterschiede liegen durchweg deutlich im einstelligen

Prozentbereich. Aus diesem Grund sind jeweils nur die Ergebnisse mit den Standardwerten für die Diversität dargestellt.

Abbildung 7.18 zeigt die aufsummierten Laufzeiten der Teilberechnungen des Aufbaualgorithmus für die Fragmenträume BRICS_4k und BRICS_9k in Abhängigkeit von der Größe der berücksichtigten Menge an Teillösungen sowie der Verwendung eines Pharmakophors. Anhand der Grafiken lässt sich zunächst erkennen, dass mit einer Verdopplung der Menge an Teillösungen in etwa eine Verdopplung der Laufzeit einher geht. Dies gilt sowohl für Berechnungen mit als auch ohne Pharmakophor. Weiterhin resultiert die Verdopplung der Anzahl der Fragmente des Fragmentraums ebenso in einer etwa doppelt so hohen Gesamtlaufzeit. Diese setzt sich dabei aus der Erzeugung der Fragmente, der Initialisierung, dem Transfer von Teilplatzierungen, dem inkrementellen Aufbau, gegebenenfalls aus den Pharmakophor-Berechnungen sowie einem Postprocessing-Anteil zusammen, der hauptsächlich aus der Substitution der offenen Link-Atome und dem anschließenden inkrementellen Aufbau der Komponenten besteht. Alle weiteren Teilberechnungen können in diesem Zusammenhang vernachlässigt werden.

Bei der Verwendung eines Pharmakophors beträgt der Anteil der Pharmakophor-Berechnungen durchschnittlich etwa 40 Prozent an der Gesamtlaufzeit, bei kleinen Werten für k_F liegt er etwas höher. Für den inkrementellen Aufbau beträgt der Anteil bei Verwendung des BRICS_4k Fragmentraums etwa 30 Prozent für kleine Werte für k_F und sinkt auf etwa 20 Prozent für große Werte von k_F . Bei BRICS_9k liegt er durchweg bei 20 Prozent und darunter. Umgekehrt steigt die benötigte anteilige Rechenzeit für die Initialisierung von etwa 20 Prozent auf etwa 35 Prozent. Der Rechenanteil für das Postprocessing beträgt zwischen drei und sechs Prozent. Der übrige Anteil der Laufzeit setzt sich aus dem Transfer der Teilplatzierungen sowie der Fragmenterzeugung zusammen. Die Anteile der beiden letzten Positionen betragen jedoch maximal zwei bzw. ein Prozent. Die Laufzeitanteile für alle weiteren Funktionen, unter anderem die Anwendung der physikochemischen Filter und die Vorverarbeitung der Daten, liegen deutlich unter 0,1 Prozent und sind in diesem Zusammenhang zu vernachlässigen.

Sofern kein Pharmakophor verwendet wird, ist der prozentuale Anteil des inkrementellen Aufbaus an der Gesamtlaufzeit wesentlich höher. Beim BRICS_4k Fragmentraum liegt er bei knapp 70 Prozent für kleine Werte von k_F und sinkt auf etwa 50 Prozent für große Werte von k_F . Bei BRICS_9k reicht die Spanne von 50 Prozent bis etwa 40 Prozent. Umgekehrt steigt auch hier der Anteil der Initialisierung von 25 auf 35 Prozent für BRICS_4k bzw. von 35 auf 50 Prozent für BRICS_9k. Auch der Anteil des Postprocessing ist mit fünf bis sieben Prozent (BRICS_4k) bzw. sieben bis neun Prozent (BRICS_9k) deutlich höher als bei den Rechnungen mit Pharmakophor. Die Anteile der übrigen Teilberechnungen liegen in derselben Größenordnung wie bei den Berechnungen mit Pharmakophor, d.h., der Transfer der Teilplatzierungen sowie die Fragmenterzeugung benötigen zwei bzw. ein Prozent und alle weiteren Teilberechnungen sind zu vernachlässigen.

Das beobachtete Verhalten lässt sich primär darauf zurückführen, dass bei größeren Werten für k_F entsprechend mehr Teillösungen erzeugt und prozessiert werden. Dieser Zusammenhang wird insbesondere bei Betrachtung der Statistiken über die Anzahl der Fragmente deutlich. Die entsprechenden Daten sind in den Abbildung 7.19 für bei-

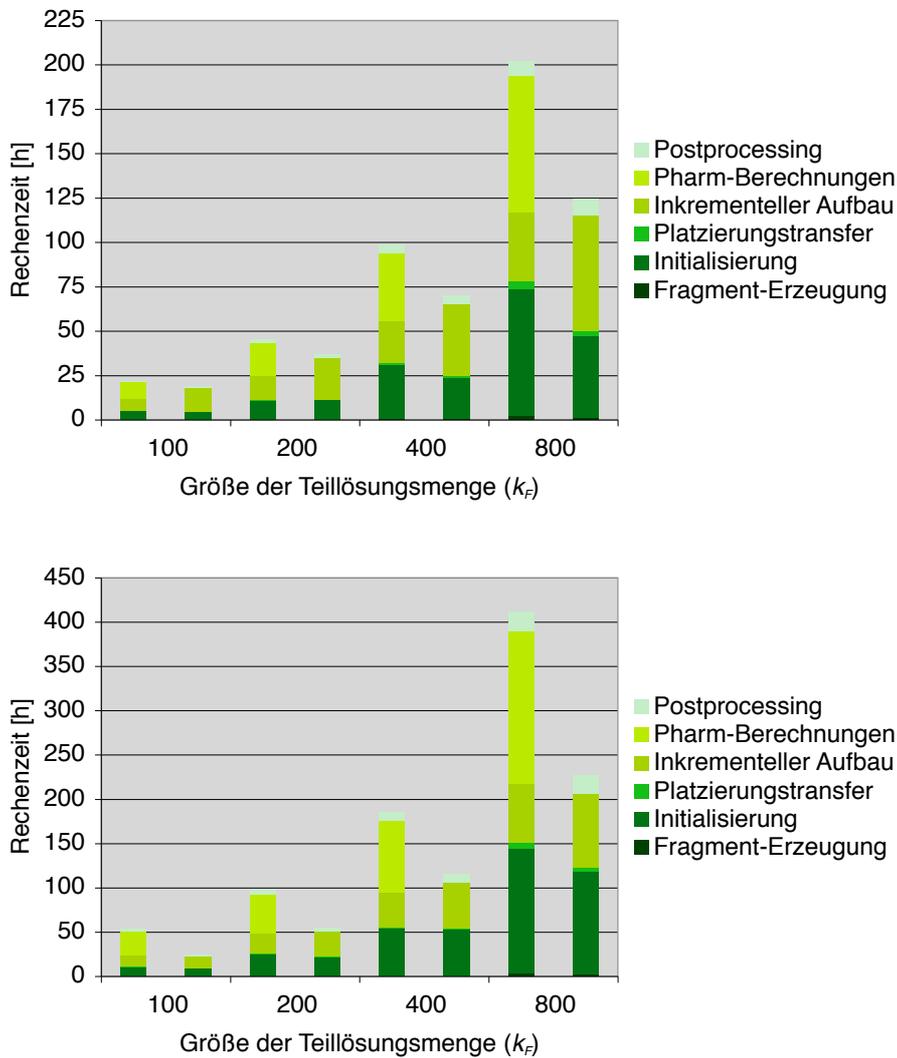


Abbildung 7.18: Laufzeitanteile der Teilberechnungen des Aufbaualgorithmus für die Fragmenträume BRICS_4k (oben) und BRICS_9k (unten) in Abhängigkeit von der Menge der berücksichtigten Teillösungen (k_F). Bei gleichem Wert für k_F zeigt die jeweils linke Säule die Anteile bei Verwendung eines Pharmakophors während des Aufbauprozesses, die jeweils rechte Säule entspricht der identischen Rechnung ohne Pharmakophor.

7 Strukturbasierte Suche in Fragmenträumen

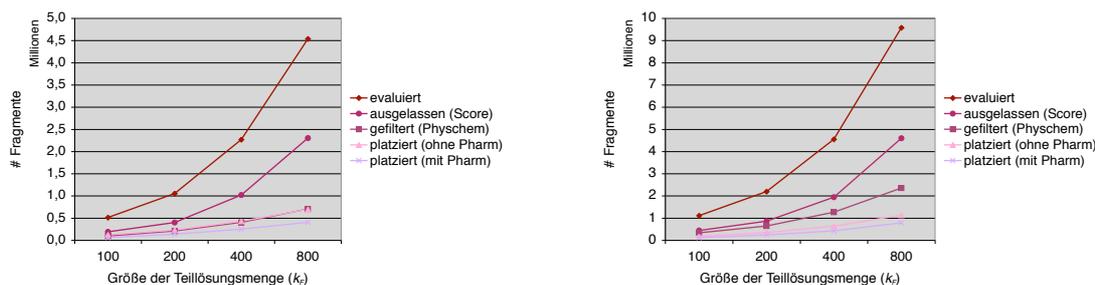


Abbildung 7.19: Fragment-Statistik des Aufbaualgorithmus für die BRICS_4k (links) und BRICS_9k (rechts) Fragmenträume in Abhängigkeit von der Menge der berücksichtigten Teillösungen.

de Fragmenträume dargestellt. Darin sind die Mengen an evaluierten, aufgrund des zu geringen Scores verworfenen, mithilfe der physikochemischen Eigenschaften gefilterten sowie tatsächlich platzierten Fragmente aufgeführt, wobei bei letzteren zusätzlich zwischen Berechnungen mit und ohne Pharmakophor unterschieden wird.

Generell spiegelt sich darin das bereits oben beschriebene Skalierungsverhalten wider, wonach eine Verdoppelung der Größe der verwendeten Teillösungen bzw. der Anzahl der Fragmente des Fragmentraums in etwa zu einer Verdopplung der jeweiligen Mengen an prozessierten Fragmenten führt. Ein signifikanter Unterschied in den Ergebnissen der Rechnungen mit bzw. ohne Pharmakophor tritt bei der Anzahl tatsächlich platzierter Fragmente auf, wobei der jeweilige Anteil ohne Berücksichtigung eines Pharmakophors um 50 bis 100 Prozent höher liegt. Ein weiterer signifikanter Unterschied lässt sich beim Vergleich der relativen Anteile der Kategorien zwischen den beiden Fragmenträumen erkennen. Entgegen des generellen Trends ist hier eine Verdreifachung der Anzahl mithilfe der physikochemischen Eigenschaften gefilterten Fragmente festzustellen. Die wesentlich größere Anzahl platzierter Fragmente im Fall der Rechnungen ohne Pharmakophor ist letztendlich für den höheren relativen wie auch absoluten Anteil des inkrementellen Aufbaus an der Gesamtlaufzeit verantwortlich.

Die Statistiken für die Platzierungen zeigen ein entsprechendes Verhalten. Die Grafiken in Abbildung 7.19 unterstreichen deutlich, dass sich durch Verwendung eines Pharmakophors die Anzahl der evaluierten wie auch der tatsächlich annotierten Platzierungen drastisch senken lässt. Dies drückt sich nur zum Teil in der Anzahl tatsächlich durch den Pharm-Filter verworfener Platzierungen aus, sondern vielmehr in der Anzahl der Platzierungen, die aufgrund dessen in den folgenden Anbausritten für Komponenten und Fragmente nicht mehr erzeugt werden können bzw. müssen. In diesem Zusammenhang wird außerdem deutlich, dass die rein geometrischen Platzierungsfiler das fehlende Pharmakophor zumindest statistisch teilweise zu „kompensieren“ scheinen.

Die große Differenz zwischen der Anzahl evaluiert und annotierter Platzierungen kommt dadurch zustande, dass erstere als Summe aller während des inkrementellen Aufbaus der Komponenten erzeugter Platzierungen ermittelt werden, wohingegen letztere die Anzahl an Platzierungen repräsentiert, die nach mehreren Clusterschritten sowie

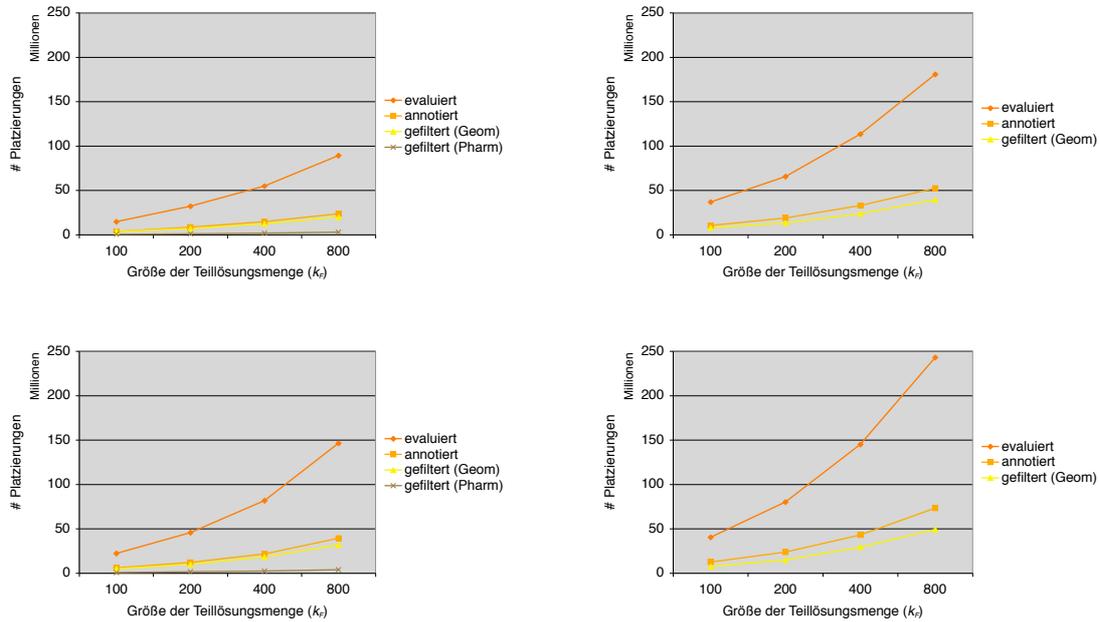


Abbildung 7.20: Platzierungsstatistik des Aufbaualgorithmus für die BRICS_4k (oben) und BRICS_9k (unten) Fragmenträume in Abhängigkeit von der Menge der berücksichtigten Teillösungen. Die linke Spalte zeigt die Anteile bei Verwendung eines Pharmakophors während des Aufbauprozesses, die rechte Spalte entspricht der identischen Rechnung ohne Pharmakophor.

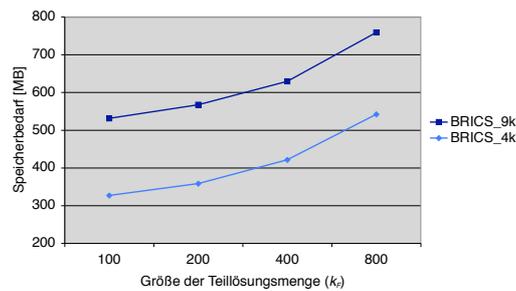


Abbildung 7.21: Speicherbedarf für die BRICS_9k und BRICS_4k Fragmenträume in Abhängigkeit von der Menge der berücksichtigten Teillösungen.

eventuell zusätzlicher Begrenzung der resultierenden Platzierungsliste tatsächlich an den Fragmenten annotiert worden ist.

Die Anzahl der tatsächlich an den Teillösungen gespeicherten Platzierungen hat dabei nur einen sehr geringen Einfluss auf die Gesamtlaufzeit sowie den Speicherbedarf. Der Standardwert von maximal 60 unterschiedlichen Platzierungen hat sich in diesem Zusammenhang als ideal herausgestellt. Bei einem wesentlich größeren Maximalwert ist kein signifikanter Einfluss auf die Qualität der Ergebnisse festzustellen. Umgekehrt tragen wesentlich kleinere Maximalwerte nicht zu einer signifikanten Verbesserung des Laufzeitverhaltens und des Speicherbedarfs bei, vielmehr tritt eine merkliche Verschlechterung der Ergebnisse ein. Dieses Ergebnis steht im Einklang mit dem bereits in Kapitel 7.2.4.7 diskutierten Verhalten.

Der benötigte Speicherbedarf des Aufbaualgorithmus hängt praktisch ausschließlich vom verwendeten Fragmentraum sowie der Menge an Teillösungen ab, die für den nächsten Anbauzyklus verwendet werden. Abbildung 7.21 enthält die dazugehörigen Werte für beide Fragmenträume. Demnach ermittelt sich der Gesamtbedarf aus einem Grundwert, der hauptsächlich von der Größe des Fragmentraums abhängt und einem variablen Beitrag, der entsprechend mit der Anzahl der berücksichtigten Teillösungen skaliert. Für eine typische Rechnung werden demnach einige Hundert MB Hauptspeicher benötigt, wovon etwa 70 MB auf das Programm selbst entfallen.

Erstaunlicherweise hat die deutliche Erhöhung der Menge an betrachteten Teillösungen keinen Einfluss auf die ersten Moleküle in der finalen Ergebnisliste. So war insbesondere das Molekül auf dem ersten Platz der finalen Ergebnisliste für alle Werte von k_F und alle Diversitätsparameter dasselbe. Auch für die Moleküle auf den nächsten fünf Rängen war dies der Fall. Eine Veränderung in der Zusammensetzung der Lösungsliste für größere Werte für k_F kann erst ab den folgenden Rängen beobachtet werden. Dabei tritt nur eine sehr geringe Verschiebung hin zu besseren Scores auf, d.h., der Wert des Moleküls auf dem letzten Rang verbessert sich maximal um einen Betrag, der in etwa zehn Prozent des Scores des ersten Rangs entspricht. Die Ursachen für dieses Phänomen sind allerdings mit Sicherheit auch in den verwendeten Bewertungsfunktionen zu suchen. Nichtsdestotrotz wird das k -Greedy-Kriterium demnach vom Algorithmus nahezu optimal ausgenutzt.

8 Zusammenfassung und Ausblick

Die vorliegende Arbeit befasst sich mit der Entwicklung eines Software-Werkzeugs für strukturbasiertes de-novo-Design sowie der Erzeugung und Modellierung chemischer Fragmenträume für den Einsatz im rationalen Wirkstoffentwurf. Das resultierende computergestützte Verfahren wird anschließend zur Vorhersage potenziell bioaktiver Verbindungen im Rahmen unterschiedlicher Anwendungsszenarien verwendet. Die daraus gewonnenen Erkenntnisse werden zur Entwicklung eines neuen Regelwerks zur Erzeugung und Modellierung chemischer Fragmenträume herangezogen, die hinsichtlich ihrer Vorhersagekraft mit bestehenden Verfahren verglichen und zur Identifikation wichtiger Kriterien für den fragmentbasierten Wirkstoffentwurf benutzt werden.

Das im Rahmen dieser Arbeit entstandene de-novo-Design-Verfahren FlexNovo [Degen & Rarey, 2006] wurde auf Basis der Programmbibliothek des molekularen Docking-Programms FlexX [Rarey *et al.*, 1996a] und der Erweiterung zur Behandlung von Pharmakophor-Randbedingungen [Hindle *et al.*, 2002] entwickelt. Dabei wurden insbesondere die zugrunde liegenden physikochemischen Modelle zur Behandlung molekularer Flexibilität (s. Kap. 4.1.2) und intermolekularer Wechselwirkungen (s. Kap. 4.1.3) adaptiert. Die Modelle wurden dabei derart modifiziert, dass die Besonderheiten von Fragmenten berücksichtigt werden können (s. Kap. 4.3.4), was in erster Linie die Behandlung der Anknüpfungsstellen (*Link-Atome*) betrifft. In diesem Zusammenhang wurde ein Phasenmodell für die *Initialisierung* entwickelt, d.h., für die Annotation der relevanten Informationen an den molekularen Graphen, wodurch die effiziente Behandlung großer Fragmenträume ermöglicht wird (s. Kap. 4.3.2). Des Weiteren wurde die Behandlung von Pharmakophor-Bedingungen modularisiert und flexibilisiert (s. Kap. 4.3.5), und damit die Grundvoraussetzung für die Berücksichtigung umfangreicher Pharmakophor-Definitionen während des inkrementellen Aufbaus von Molekülen in der Bindetasche geschaffen (s. Kap. 5.3).

Für die Platzierung von Fragmenten in der Bindetasche eines Proteins wurden die Algorithmen von FlexX [Rarey *et al.*, 1996b, 1997] zur Platzierung kleiner molekularer Verbindungen so angepasst, dass diese nicht nur für einzelne Liganden sondern auch für eine große Anzahl von Fragmenten unter Berücksichtigung von deren Besonderheiten verwendet werden können (s. Kap. 5.2.3). Auf dieser Basis wurde ein *k*-Greedy-Algorithmus zum sequenziellen Aufbau von Molekülen aus Fragmenten in der Bindetasche eines Proteins entwickelt. Der Algorithmus zeichnet sich dadurch aus, dass neben den Randbedingungen des Aktiven Zentrums etliche weitere Kriterien direkt in den Aufbauprozess integriert sind. Dazu zählen wichtige physikochemische Eigenschaften, die in Form von Filterkriterien definiert werden können (s. Kap. 4.3.3). Des Weiteren sind mehrere Kriterien für die Realisierung struktureller Diversität integriert, die die Anzahl der jeweiligen „atomaren“

Fragmente in jedem Anbauzyklus begrenzen sowie Beschränkungen für die Ähnlichkeit der Moleküle der finalen Lösungsliste beinhalten (s. Kap. 5.3.5). Zusätzlich sind neben den bereits in FlexX integrierten geometrischen Filterfunktionen für Platzierungen [Stahl & Böhm, 1998] zwei weitere entwickelt worden, die bekannte Schwachstellen der verwendeten Bewertungsfunktionen zum Teil kompensieren sollen (s. Kap. 5.3.4.5). Nicht zuletzt wurde die Verwendung von komplexen Pharmakophor-Bedingungen in den *de-novo*-Design-Algorithmus integriert. Dabei müssen die Informationen über erfüllte, noch erfüllbare und nicht mehr zu erfüllende Einzelbedingungen der Pharmakophor-Definition für jede Teillösung individuell verwaltet werden, was eine gewisse Flexibilität hinsichtlich der Reihenfolge der Erfüllung der Einzelbedingungen mit sich bringt (s. Kap. 5.3.4.2). Durch die Abschätzung der maximalen energetischen Beiträge aller „atomaren“ Fragmente und entsprechender Prozessierung der jeweiligen Teillösungslisten kann die verwendete Heuristik effizient ausgenutzt werden.

FlexNovo wurde für die Vorhersage potenziell bioaktiver Moleküle in Verbindung mit unterschiedlichen Proteinen und Fragmenträumen verwendet. Dabei sollten verschiedene Aspekte in typischen Anwendungsszenarien untersucht werden, die sich auch in den verwendeten Randbedingungen und Filterkriterien widerspiegeln. Die erhaltenen Ergebnisse wurden möglichst umfassend charakterisiert. Zunächst erfolgte die Beurteilung der erzeugten Moleküle hinsichtlich ihrer Plausibilität und Komplementarität zu der jeweiligen Proteinbindetasche. Dabei zeigte sich, dass FlexNovo für alle Anwendungsbeispiele in der Lage war, trotz einer verhältnismäßig geringen Menge an Ergebnissen, sinnvolle Moleküle zu erzeugen, die strukturell gleichermaßen verwandte wie alternative Elemente zu bekannten Inhibitoren zeigten. Gleichzeitig wurden für alle Moleküle niederenergetische Platzierungen erzeugt, wobei der Bindungsmodus dem der natürlichen Liganden sehr ähnlich ist und dadurch alle wichtigen Wechselwirkungen in der Bindetasche erfüllt werden konnten (s. Kap. 7.2.4). Die automatischen Ähnlichkeitsanalysen der Ergebnisse zu gesonderten Zusammenstellungen von Inhibitoren zeigten sowohl die Konservierung spezifischer struktureller Motive als auch die Verwendung sinnvoller bioisosterer Ersetzungen unter weitgehender Beibehaltung der Platzierungsgeometrie (s. Kap. 7.2.4.2). Die Charakterisierung der kompletten Lösungsmengen hinsichtlich ihrer physikochemischen Eigenschaften ergab, dass die Verteilungen wichtiger physikochemischer Größen in allen Anwendungsszenarien große Ähnlichkeit zu den analogen Verteilungen bekannter Inhibitoren aufweisen. In vielen Fällen konnten trotz der sehr geringen Anzahl erzeugter Moleküle spezifische Charakteristika der verschiedenen Targets reproduziert werden (s. Kap. 7.2.4.3). Die Analyse der inhärenten strukturellen Diversität der einzelnen Lösungsmengen zeigte, dass innerhalb der 100 erzeugten Moleküle zwischen 14 und 18 unterschiedliche molekulare Gerüste (*Scaffolds*) enthalten sind, wobei zwei Strukturen dabei mindestens 15 Schweratome gemeinsam haben mussten, um dem identischen Scaffold zugeordnet zu werden (s. Kap. 7.2.4.4). Die ermittelten Werte für die synthetische Zugänglichkeit liegen im Bereich von vier bis sechs Einheiten bezüglich der Bewertung durch Sylvania [Boda *et al.*, 2007] und repräsentieren somit Werte, die typischen niedermolekularen Verbindungen im Rahmen des Wirkstoffentwurfs entsprechen. Dies zeigt sich auch bei den analogen Verteilungen der Werte für die Zusammenstellungen der Inhibitoren, die demzufolge ein sehr ähnliches Profil aufweisen (s. Kap. 7.2.4.5). Bereits

durch die Verwendung der Anker-Pharmakophor Bedingungen konnten von FlexNovo in vier der fünf Anwendungsbeispielen Moleküle erzeugt werden, die alle wesentlichen (nicht nur die explizit geforderten) Wechselwirkungen in den jeweiligen Bindetaschen erfüllen und gleichzeitig eine hohe strukturelle und physikochemische Komplementarität aufweisen (s. Kap. 7.2.4.6). Das Potenzial, das sich durch die Berücksichtigung komplexer Pharmakophor-Definitionen eröffnet, wird insbesondere im Fall von Thrombin deutlich, wofür FlexNovo mit der alleinigen Verwendung eines Anker-Pharmakophors keine zufriedenstellenden Lösungen erzeugen konnte. In Verbindung mit einem fokussierten Fragmentraum und der Spezifikation von Pharmakophor-Einzelbedingungen für alle drei Subtaschen des Aktiven Zentrums konnte FlexNovo viel versprechende Moleküle erzeugen, die bezüglich ihrer physikochemischen Eigenschaften sowie des vorhergesagten Bindungsmodus große Ähnlichkeit zu bekannten Inhibitoren aufweisen (s. Kap. 7.2.5). Durch die Verwendung umfangreicher Pharmakophor-Spezifikationen ändert sich das Laufzeitverhalten von FlexNovo zwar deutlich, jedoch kann keine Veränderung der allgemeinen Skalierungseigenschaften festgestellt werden. Generell hängt sowohl die benötigte Rechenzeit wie auch der Speicherbedarf lediglich von der Größe des verwendeten Fragmentraums ab (s. Kap. 7.2.4.7). Daneben hat die Menge der Teillösungen, die für den nächsten Anbauschnitt berücksichtigt werden, einen signifikanten Einfluss auf die Laufzeit. Es zeigt sich, dass diese, wie auch die Mengen an evaluierten, verworfenen und tatsächlich erzeugten Teillösungen bzw. Platzierungen, in linearer Weise von der Anzahl der berücksichtigten Teillösungen abhängen (s. Kap. 7.3). Dabei tragen die verwendeten physikochemischen Filterkriterien sowie die energetische Abschätzung der „atomaren“ Fragmente signifikant zur Reduktion der Laufzeit bei. Ferner zeigt die Analyse der anteiligen Laufzeiten der einzelnen Teilberechnungen des Aufbaualgorithmus, dass sich die Gesamtlaufzeit zum Großteil aus der Initialisierung der Teillösungen, dem inkrementellen Aufbau der Komponenten sowie gegebenenfalls aus den Pharmakophor-Berechnungen zusammensetzt. Durch die Wiederverwertung der Informationen über Teilplatzierungen sowie der Einführung des Phasenmodells für die Initialisierung konnte somit ein zeitlich effizienter und gut skalierender Algorithmus für den inkrementellen Aufbau von Molekülen in der Bindetasche eines Proteins realisiert werden (s. Kap. 7.3).

Insgesamt wurde mit FlexNovo ein umfangreiches, robustes und modulares Software-Werkzeug entwickelt, das viele wichtige und zueinander komplementäre Randbedingungen des strukturbasierten de-novo-Designs integriert und miteinander kombiniert [Stahl *et al.*, 2002]. Dadurch erlaubt FlexNovo eine breite Einsatzfähigkeit sowie die Integration in unterschiedliche Arbeitsabläufe. Dabei trägt die in Anlehnung an FlexX entwickelte Benutzungsschnittstelle – auf die erst in Anhang A dieser Arbeit eingegangen wird – sowie die ausführliche und umfassende Dokumentation entscheidend zur Akzeptanz und Anwendbarkeit von FlexNovo bei. So hat FlexNovo in vielen unterschiedlichen Anwendungsstudien neben qualitativ hochwertigen Ergebnissen seine Robustheit und Anwendbarkeit unter Beweis stellen können [Zaliani, 2007].

Die Ergebnisse von FlexNovo haben gezeigt, dass nicht nur die Randbedingungen sondern insbesondere auch die verwendeten Fragmente einen entscheidenden Einfluss auf die Qualität der Ergebnisse haben. Diese Aspekte wurden im Rahmen der Untersuchungen

fokussierter Fragmenträume analysiert. In diesem Zusammenhang wurden Fragmente aus Inhibitoren 40 unterschiedlicher Target-Klassen erzeugt und darauf basierende Fragmenträume definiert (s. Kap. 6.1.4). Die Analyse der grundlegenden Eigenschaften dieser Fragmenträume hat zunächst ergeben, dass die meisten zwar theoretisch unendlich groß sind, sich die tatsächliche Anzahl zugänglicher Moleküle jedoch deutlich unterscheidet (s. Kap. 6.2.2). Die Ursache dafür ist nur zum Teil in der Menge der dem jeweiligen Raum zugrunde liegenden Fragmente zu suchen. Vielmehr handelt es sich um das Zusammenspiel von Fragment-Topologie, Verteilung der Link-Atome und Regelwerk für die Verknüpfung der unterschiedlichen chemischen Motive. Aufschlussreicher ist in diesem Zusammenhang die Bestimmung der Anzahl an Molekülen, die sich aus den jeweiligen Räumen erzeugen lassen, unter Berücksichtigung von Bereichen physikochemischer Eigenschaften, die insbesondere im Rahmen der Leitstruktursuche von Interesse sind [Vieth *et al.*, 2004; Vieth & Sutherland, 2006]. Dazu wurde ein Software-Werkzeug basierend auf einer Enumerationsprozedur angewandt, das alle Moleküle eines Fragmentraums erzeugt, deren physikochemische Eigenschaften innerhalb definierter Wertebereiche liegen [Pärn *et al.*, 2007]. Die Resultate zeigen, dass ein Großteil der fokussierten Fragmenträume unter diesen Bedingungen vollständig enumeriert werden konnte. Dabei sind große Unterschiede im Verhältnis der tatsächlich erzeugten Moleküle zu der theoretisch möglichen Anzahl an Verbindungen zu beobachten. Darauf bezogen liegt der Prozentsatz der generierten Moleküle zwischen 0,1 Prozent und etwa 40 Prozent. Die tatsächliche Anzahl an Molekülen unterscheidet sich zwischen den Klassen ebenfalls um einige Größenordnungen und reicht von wenigen Dutzend bis zu etwa einer halben Million. Eine einfache Regel, ob ein gegebener Fragmentraum prinzipiell enumerierbar ist oder nicht, konnte dabei nicht abgeleitet werden. Allerdings liegt bei den verwendeten Fragmenträumen die Obergrenze bei etwa 10^8 theoretisch zugänglicher Moleküle. Die weitere Charakterisierung der Ergebnisse hinsichtlich ihrer Ähnlichkeit zu den ursprünglich für die Erstellung der Fragmenträume verwendeten Inhibitoren zeigt, dass auch dort große Unterschiede zwischen den Klassen vorliegen. Die kleinen Fragmenträume zeigen dabei eine deutlich stärkere Verschiebung hin zu sehr hohen Ähnlichkeitswerten als dies für die großen Fragmenträume der Fall ist, bei denen etliche Zehntausend bis Hunderttausend Moleküle enumeriert wurden. Insgesamt zeigt sich durch diese Untersuchungen, dass die zugrunde liegenden Fragmente einen entscheidenden Einfluss auf die Menge an zugänglichen Molekülen eines Fragmentraums und deren Ähnlichkeit zu bekannten Inhibitoren haben. Das verwendete Regelwerk zur Ableitung von Fragmenten sowie für die Kompatibilität der jeweiligen chemischen Motive spielt dabei ebenfalls eine wichtige Rolle.

Die bis dahin gesammelten Erfahrungen im Umgang mit Fragmenten und der dazugehörigen Fragmenträume waren der Anlass, ein neues und verbessertes Regelwerk zu entwickeln, um daraus vielseitige und leistungsfähige Fragmenträume zu generieren. Das in diesem Zusammenhang neu entwickelte BRICS-Modell erweitert die Anzahl der explizit modellierten chemischen Umgebungen um wichtige strukturelle Motive und verwendet dabei gleichzeitig eine wesentlich präzisere Definition der zu fragmentierenden Substrukturen [Degen *et al.*, 2008]. Neben diesen Aspekten wurden explizite bioisostere Ersetzungen modelliert, spezifische Substitutionsmuster beibehalten und außerdem

zwischen in alpha-Position zum Link-Atom aktivierten und nicht-aktivierten heterozyklischen Ringsystemen unterschieden (s. Kap. 6.1.3). Zu Vergleichszwecken wurde das RECAP-Modell [Lewell *et al.*, 1998] innerhalb derselben Shredding-Umgebung reimplementiert. Beide Regelwerke wurden zur Ableitung von Fragmenten aus dem 'drug-like' Subset der ZINC-Datenbank und dem World Drug Index (WDI) verwendet. Dabei zeigt sich, dass mithilfe von BRICS ein um zehn Prozent höherer Anteil an Molekülen aus dem jeweiligen Katalog zerlegt werden kann. Gleichzeitig werden auch weniger terminale Fragmente erzeugt, d.h. Fragmente mit nur einem Link-Atom, was sich auch geringfügig in den Verteilungen der Molekulargewichte niederschlägt. Die Shredding-Ergebnisse wurden zur Erzeugung von Fragmenträumen unterschiedlicher Größe verwendet. Dazu wurden zunächst strukturell identische Fragmente in den Shredding-Ergebnissen der ZINC-Datenbank und des WDI ermittelt. Für BRICS ergaben sich dabei 4800 Fragmente und damit 16 Prozent mehr als bei RECAP (s. Kap. 6.1.6). Aus beiden Schnittmengen wurden zwei Fragmenträume erzeugt, BRICS_4k und RECAP_4k, wobei letzterer als Referenz verwendet wurde. Die übrigen Fragmente der ZINC-Datenbank wurden hinsichtlich ihrer Ähnlichkeit zu den WDI-Fragmenten charakterisiert, woraus zwei weitere Fragmenträume, BRICS_9k und BRICS_20k, mit ungefähr 9300 bzw. 23000 Fragmenten abgeleitet wurden. Die Leistungsfähigkeit aller Fragmenträume wurde in umfangreichen Validierungsexperimenten untersucht, in denen Moleküle aus drei komplementären, großen und diversen Anfragedatensätzen unter Verwendung der jeweiligen Fragmenträume rekonstruiert werden sollten. Dafür wurden Moleküle aus der ZINC-Datenbank, dem WDI und PubChem verwendet, wobei es sich bei letzterem um einen echten Testdatensatz handelt, da keine der darin vorkommenden Verbindungen für die Erzeugung der Fragmenträume verwendet wurde. Die Ergebnisse zeigen deutlich, dass mit BRICS_4k gegenüber RECAP_4k eine wesentlich größere Menge an Anfragemolekülen bezüglich aller Datensätze reproduziert werden konnte. Dabei konnte die Leistungsfähigkeit für den WDI mehr als verdoppelt, für ZINC und PubChem mehr als verdreifacht werden. Mit den größeren Fragmenträumen BRICS_9k und BRICS_20k konnten die Resultate weiter deutlich verbessert werden, insbesondere in Bezug auf die Anfragemoleküle aus ZINC und PubChem. Auch für den WDI konnte eine signifikante Verbesserung durch die Verwendung von Fragmenten aus kommerziell erhältlichen Verbindungen erzielt werden. Insgesamt konnten demnach für alle Testdatensätze mehr als 20 Prozent aller Anfragemoleküle rekonstruiert werden (s. Kap. 6.2.3). Die erstellten BRICS-Fragmenträume zeichnen sich dabei durch eine breite Anwendbarkeit aus und die zugrunde liegenden Fragmente repräsentieren sowohl Inhibitoren aus dem WDI als auch kommerziell sowie öffentlich zugängliche Moleküle wesentlich besser, als das mit bisherigen Ansätzen möglich ist.

Grenzen des de-novo-Design-Verfahrens

Unabhängig von der erfolgreichen Anwendung von FlexNovo im Rahmen dieser Arbeit hat der Einsatz automatischer de-novo-Design-Verfahren prinzipielle Grenzen, von denen die wichtigsten hier kurz zusammenfassend dargestellt werden. Durch Verwendung strukturell wohl-definierter Fragmente und eines umfangreichen Regelwerks zu deren Verknüp-

fung können viele wichtige Aspekte im Rahmen der Leitstruktursuche und -optimierung berücksichtigt werden. Dennoch lassen sich einige erst auf Basis des gesamten Moleküls beurteilen.

So kann die letztendliche Synthetisierbarkeit einer Verbindung nicht allein durch die Verwendung von aus existierenden Molekülen abgeleiteten Fragmenten garantiert werden. Diese Entscheidung kann nur im Einzelfall und anhand einer umfassenden retrosynthetischen Analyse getroffen werden, die neben strategischen Bindungen auch mögliche Edukte und deren Verfügbarkeit für die Synthese berücksichtigt [Corey & Cheng, 1995]. Des Weiteren können pharmakodynamische und pharmakokinetische Gesichtspunkte wie Absorption, Verteilung, Metabolisierung, Ausscheidung und Toxizität in noch geringerem Maße auf Fragmentebene behandelt werden. Selbst für einen gegebenen Satz von Molekülen ist die Abschätzung dieser Aspekte nur eingeschränkt und mit großem Rechenaufwand möglich [van de Waterbeemd & Gifford, 2003; Norinder & Bergstrom, 2006; Gola *et al.*, 2006].

Der Entwurf von *Prodrugs*, d.h. von Medikamenten, aus denen erst nach Metabolisierung im Organismus der eigentliche Wirkstoff gebildet wird, ist im fragmentbasierten molekularen Design nur eingeschränkt möglich. Mithilfe dieser Strategie lassen sich eventuelle Probleme der Bioverfügbarkeit, der vorzeitigen Verstoffwechslung oder bei der Überwindung der Blut-Hirn-Schranke adressieren. Derartige Konzepte können gegebenenfalls direkt bei der Modellierung des Fragmentraums berücksichtigt werden, indem spezielle strukturelle Motive verwendet werden. Allerdings ist eine Vorhersage pharmakologischer Eigenschaften mit großen Unsicherheiten behaftet (s.o.).

Die Identifikation nicht-kompetitiver Antagonisten, d.h. von Inhibitoren, die nicht an die Bindungsstelle des Agonisten sondern beispielsweise an eine allosterische Position binden, ist mit einem strukturbasierten de-novo-Design-Verfahren – wenn überhaupt – nur eingeschränkt möglich. In der Regel erfordern die Rechnungen eine sehr strikte Definition der Randbedingungen, was sich insbesondere auch auf die Spezifikation der Bindetasche des Proteins bezieht. Unter Berücksichtigung von Pharmakophor-Bedingungen ist es jedoch für den Algorithmus praktisch unmöglich, alternative Bindungsstellen zu identifizieren. Allerdings können bei Verzicht auf die Verwendung eines Pharmakophors und Berücksichtigung eines großen Teils der tatsächlichen Proteinstruktur allosterische Bindungsmodi prinzipiell vorhergesagt werden.

Zur Bewertung der physikochemischen Komplementarität zwischen Protein und Molekül bzw. für die Abschätzung der freien Bindungsenthalpie werden in der Regel relativ einfache Bewertungsfunktionen verwendet [Schneider & Fechner, 2005], die tatsächliche Bindungsaffinitäten in der Regel nur mit unzureichender Genauigkeit berechnen können. Bei der Anwendung des de-novo-Design-Verfahrens sind entsprechende Effekte aufgrund der großen Anzahl potenzieller Teillösungen umso stärker ausgeprägt. Dadurch ist die Differenzierung der einzelnen Teillösung entsprechend schwierig und die Wahrscheinlichkeit, dass der Algorithmus im Rahmen der Optimierung vermeintliche lokalen Minima findet ist verhältnismäßig hoch [Stahl *et al.*, 2002].

Die induzierte Anpassung der Geometrie der Bindetasche durch die Anlagerung eines Moleküls (*induced fit*) kann momentan nicht berücksichtigt werden. Allerdings spielt eine solche Veränderung nicht bei allen Proteinen eine Rolle und der Proteinflexibilität

sind ebenso natürlich Grenzen gesetzt. In der Regel gehen derartige Effekte zu Lasten der Spezifität der Wechselwirkungen und der Bindungsaffinität [Böhm *et al.*, 2002]. Sie spielen also für das de-novo-Design eher eine untergeordnete Rolle.

Schlussfolgerungen für den praktischen Einsatz von de-novo-Design-Verfahren

Für den praktischen Einsatz von de-novo-Design-Verfahren im Rahmen des rationalen strukturbasierten Wirkstoffentwurfs sollten alle spezifischen Informationen berücksichtigt werden, die über das jeweilige Target vorhanden sind. Dies schließt neben der Bindetasche potenziell auch Informationen über obligatorisch auszubildende Wechselwirkungen sowie physikochemische Eigenschaften mit ein. Mit FlexNovo wurde ein Software-Werkzeug entwickelt, das die Verwendung verschiedenster komplementärer Randbedingungen im Rahmen des de-novo-Designs ermöglicht.

Die Spezifikation der Bindetasche sollte dabei so genau wie möglich erfolgen, ohne jedoch Möglichkeiten für sinnvolle strukturelle Variationen oder Ergänzungen von vorne herein auszuschließen. Sofern die Bindetasche Wassermoleküle beinhaltet, sollte deren Rolle in Bezug auf die Ausbildung und Vermittlung gerichteter Wechselwirkungen kritisch untersucht werden. Einerseits kann es für die Suche nach alternativen Antagonisten von Vorteil sein, Wassermoleküle aus der Bindetasche zu „verdrängen“. Andererseits können Wassermoleküle aber auch als Mediatoren für wichtige Wechselwirkungen fungieren und sollten beim Entwurf neuer Verbindungen entsprechend berücksichtigt werden. Schließlich gibt es Wassermoleküle, die feste Bestandteile der Bindetasche sind, wie das bei der Dihydrofolat-Reduktase der Fall ist. Diese sollten unbedingt beibehalten werden, da sie auch bei der Bindung von Liganden nicht verdrängt werden.

Eine kritische und umfassende Evaluation ist auch bei der Formulierung der Pharmakophor-Bedingungen notwendig. Sofern diese aus bekannten Inhibitoren abgeleitet werden, sind die Einzelbedingungen daraufhin zu überprüfen, ob sie Teil eines speziellen Wechselwirkungsmusters sind, das sich ausschließlich auf eine Serie von Inhibitoren bezieht, oder ob sie generischen Charakter haben und damit potenziell für die Identifikation neuer chemischer Verbindungen geeignet sind. Unabhängig von der Datengrundlage, aus denen die Pharmakophor-Bedingungen abgeleitet worden sind, ist bei Verknüpfung mehrerer Einzelbedingungen darauf zu achten, dass dadurch eine hinreichende Einschränkung des Lösungsraums erzielt wird, ohne jedoch mögliche strukturelle Variationen *a priori* zu stark zu fokussieren. Dies trifft neben der Formulierung des Anker-Pharmakophors insbesondere auf die Spezifikation des Pharmakophors für den Aufbauprozess zu (s. Kap. 7.2.5). Dabei ist außerdem zu beachten, dass sehr komplexe Pharmakophor-Spezifikationen die benötigte Rechenzeit um ein Vielfaches erhöhen können (s. Kap. 7.3).

Die Verwendung von Filterkriterien für physikochemische Eigenschaften während des Aufbauprozesses ist dringend zu empfehlen, da sie gleichermaßen den Lösungsraum einschränken und die Erzeugung von Molekülen mit gänzlich unerwünschten Eigenschaften verhindern (s. Kap. 7.2.4.6). Bei Ableitung der Wertebereiche für physikochemische Eigenschaften aus Zusammenstellungen von Inhibitoren ist in besonderer Weise auf die strukturelle Zusammensetzung der verwendeten Bibliothek zu achten. Sollte diese, wie

im Fall des Estrogenrezeptors, sowohl Agonisten als auch Antagonisten enthalten, ergeben sich unter Umständen sehr große Wertebereiche, die sich nicht für die Erzeugung einer fokussierten Ergebnismenge eignen, sodass auch dort potenzielle Agonisten und Antagonisten enthalten sind (s. Kap. 7.2.4.3). Haben die Inhibitoren jedoch eine sehr hohe Ähnlichkeit zueinander und zeigen wenig strukturelle Variation, wie im Fall der Cyclooxygenase-2, so sind die resultierenden Filterkriterien gegebenenfalls zu strikt und die Ergebnismenge zeigt ebenfalls eine geringe Diversität und nur wenig alternative Strukturvorschläge (s. Kap. 7.2.4.1). Die Wahl der Filterkriterien hängt somit in entscheidender Weise von dem Anwendungsszenario und den gewünschten Ergebnissen ab.

Dies trifft analog auch auf die Datengrundlage, d.h., auf den verwendeten Fragmentraum zu. Prinzipiell ist zunächst die Entscheidung zu treffen, ob ein generischer Fragmentraum, z.B. BRICS_4k, oder ein fokussierter Fragmentraum verwendet werden soll. Ersterer hat dabei den Vorteil, dass er viele unterschiedliche Motive enthält und sich dadurch eine relativ hohe strukturelle Diversität und breite Anwendbarkeit ergibt (s. Kap. 6.2.3.2). Viele Gründe sprechen aber für die Verwendung eines, beispielsweise aus BRICS abgeleiteten, fokussierten Fragmentraums. Abhängig von dem verwendeten Target und den gewünschten Ergebnissen sollten durch Anwendung von physikochemischen Filterkriterien und SMARTS-Ausdrücken spezifische strukturelle Motive angereichert und für ein spezielles Target eher untypische oder ungeeignete Substrukturen ausgeschlossen werden. In gleicher Weise gilt dies auch für das verwendete Regelwerk zur Verknüpfung von Fragmenten. Dieses sollte ebenfalls hinsichtlich für ein spezifisches Target sinnvollen und ungeeigneten Regeln untersucht und gegebenenfalls angepasst werden. Auch die verwendeten Randbedingungen können gegen die Verwendung großer generischer Fragmenträume sprechen. So kann eine relativ große Bindetasche in Verbindung mit einer komplexen Pharmakophor-Definition dazu führen, dass die Verwendung eines Fragmentraums wie z.B. BRICS_4k praktisch nicht sinnvoll ist (s. Kap. 7.2.5). Umgekehrt kann eine zu starke Einschränkung hinsichtlich der Anzahl der verwendeten Fragmente und der strukturellen Motive kontraproduktiv sein, weshalb die Ableitung von Fragmenten aus einer kleinen Menge an Inhibitoren kaum neue Strukturvorschläge ergeben wird. Generell ist deshalb die prinzipielle Verwendung einer Menge generischer Fragmente zur Realisierung typischer bzw. gewünschter bioisosterer Ersetzungen zu empfehlen. Neben prinzipiellen Aspekten zur Auswahl von Fragmenten hat insbesondere die Wahl der Startfragmente einen entscheidenden Einfluss auf die erzeugten Ergebnisse. Im Fall der Cyclooxygenase-2 lassen sich beispielsweise bereits wesentliche strukturelle Kriterien potenzieller Inhibitoren durch die Platzierung geeigneter Startfragmente realisieren. Allerdings sollten auch in diesem Fall die vom Algorithmus vorgeschlagenen Fragmente kritisch hinsichtlich weiterer spezifischer (pharmakologischer) Eigenschaften überprüft und gegebenenfalls verworfen werden (s. Kap. 7.2.4.1).

In Abhängigkeit von dem verwendeten Protein sowie der Eigenschaften der Fragmente des Fragmentraums ist die Anpassung der algorithmischen Parameter notwendig. Die Topologie der zu erzeugenden Moleküle spielt dabei eine entscheidende Rolle. Aus der Betrachtung komplexierter Liganden oder bekannter Inhibitoren sollte die typische Anzahl strategischer Bindungen bezüglich des Regelwerks zur Erzeugung der Fragmente des

zugrunde liegenden Fragmentraums ermittelt werden. Typischerweise sind zwei Anbauzyklen ausreichend, aber in bestimmten Fällen kann eine Erhöhung auf drei Fragment-Anbauschritte zwingend notwendig sein, um die gewünschten Randbedingungen erfüllen zu können, wie am Beispiel von Thrombin ersichtlich wird (s. Kap. 7.2.5).

In speziellen Fällen kann ein schrittweises Vorgehen während des Aufbauprozesses eines de-novo-Design-Verfahrens sinnvoll sein. Diese Möglichkeit wurde bei der Entwicklung von FlexNovo bereits zu Beginn explizit berücksichtigt (s. Kap. 5.3.1). Dadurch können Schwachstellen bei der Bewertung von Zwischenlösungen bzw. Auswahl struktureller Motive durch manuelle Inspektion und Nachoptimierung kompensiert werden. Allerdings erfordert dies eine gezielte Anpassung der algorithmischen Parameter und ist insgesamt wesentlich rechenintensiver. Allerdings ermöglicht eine derartige Strategie die Einbeziehung von Wissen, mit dessen Hilfe eine sehr spezifische Vorhersage potenziell aktiver Verbindungen ermöglicht wird.

Erweiterungen bestehender Ansätze

Die vorliegende Arbeit belegt, dass der erfolgreiche Einsatz von de-novo-Design-Verfahren im strukturbasierten Wirkstoffentwurf möglich ist, sofern eine Vielzahl komplementärer Randbedingungen verwendet wird. Durch die Entwicklung eines integrierten und modularen Software-Werkzeugs im Rahmen dieser Arbeit ist das Potenzial eines de-novo-Design-Ansatzes sicherlich noch nicht erschöpft. Anhand der erzielten Ergebnisse lassen sich einige Schlüsselaspekte identifizieren, die als Ansatzpunkte für mögliche Erweiterungen des bestehenden Verfahrens in Betracht kommen.

Der mit Abstand wichtigste Punkt betrifft die verwendete Bewertungsfunktion zur Abschätzung der freien Bindungsenthalpien, von der die Ergebnisse maßgeblich abhängen (s. Kap. 4.1.4). Durch eine präzisere Bewertung könnte nicht nur eine bessere Diversifizierung der Teillösungen erreicht werden, sondern auch eine bessere Korrelation der vorhergesagten freien Bindungsenthalpien der Ergebnisse mit tatsächlichen experimentell bestimmten Affinitäten erreicht werden. Denkbar wäre auch die Verwendung zweier unterschiedlicher Funktionen, wovon die eine schnell zu berechnen sein müsste und für die Erzeugung der Platzierungen und die andere zur Nachoptimierung der Platzierungen und tatsächlichen Abschätzung der Affinität verwendet würde. Für letzteres kommen beispielsweise DrugScore [Gohlke *et al.*, 2000] oder HYDE [Reulecke *et al.*, 2008] in Frage.

Ein weiterer zentraler Aspekt beim de-novo-Design ist die synthetische Zugänglichkeit der erzeugten Moleküle. Als Erweiterung zum Modell der binären Kompatibilität für stark diversifizierte chemische Umgebungen kann die Berücksichtigung physikochemischer Parameter der jeweiligen Motive hinzugezogen werden. Dadurch ließen sich Aspekte berücksichtigen, die nicht nur die nähere chemische Umgebung des Link-Atoms betreffen, sondern Eigenschaften des gesamten Fragments reflektieren, wozu beispielsweise Polarisierbarkeit und Dipolmoment zählen. Auch eine präzisere Steuerung der erzeugten Stereoisomere bei der Verknüpfung von Fragmenten wäre wünschenswert, um beispielsweise die Konfiguration chiraler Bausteine konservieren zu können (s. Kap. 4.3.4).

Nicht zuletzt ist die Berücksichtigung der Flexibilität der Proteinstruktur eine große

Herausforderung. Obwohl im Rahmen des strukturbasierten Designs nur in seltenen Fällen die Apo-Struktur eines Proteins verwendet wird, d.h. die Konformation des Proteins ohne Ligand, so könnten doch Effekte wie der *induced fit* [Koshland & Neet, 1968] damit wesentlich besser berücksichtigt werden. In einem ersten Schritt könnte die Flexibilität der Seitenketten der Aminosäuren integriert werden, wie das beispielsweise in der *Ensemble*-Erweiterung [Claußen *et al.*, 2001] des molekularen Docking-Programms FlexX realisiert ist.

Die umgesetzte Implementierung – auf die erst im Anhang dieser Arbeit im Detail eingegangen wird (s. Anh. B), die hier aber dennoch kurz angesprochen werden soll – ermöglicht dabei insbesondere eine einfache und direkte Erweiterbarkeit der bestehenden Funktionalität. Besondere Bedeutung kommt in diesem Zusammenhang der im Rahmen dieser Arbeit entstandenen Bibliothek zur Behandlung chemischer Fragmenträume und physikochemischer Informationen zu. Bereits mehrere Projekte im Bereich chemischer Fragmenträume bedienen sich erfolgreich dieser Bibliothek und tragen zu deren Erweiterung und Verbesserung bei. Somit können neue Konzepte durch die Möglichkeit einer einfachen und direkten Integration schnell umgesetzt und die damit erzielten Ergebnisse validiert werden.

Anhang

A Die Benutzungsschnittstelle von FlexNovo

FlexNovo basiert auf der FlexX-Programmbibliothek. Viele grundlegende Einstellungen, Parameter und Befehle sind deshalb gemeinsam mit denen von FlexX und werden an dieser Stelle nicht im Detail erläutert. Die entsprechenden Informationen können dem FlexNovo bzw. dem FlexX-Benutzerhandbuch entnommen werden [Degen & Rarey, 2007; Rarey, 2008].

In diesem Kapitel sind die neu hinzugekommenen sowie die für FlexNovo wichtigen Einstellungen, Befehle und Parameter zusammengestellt. Bevor diese jedoch einzeln detailliert beschrieben werden, soll die grundlegende Benutzung von FlexNovo anhand eines Tutoriums erläutert werden.

Die folgenden Passagen stammen aus dem FlexNovo-Benutzerhandbuch und sind deshalb in englischer Sprache verfasst.

A.1 Einführung in FlexNovo

This is meant to be a quick (but not dirty ;-)) introduction to de novo design with FlexNovo. FlexNovo is extremely flexible and configurable, and you will learn in the later sections where to tune to improve your results. Here we would like to give you a rough outline of the first steps to take to get the program running, acquaint you with the most important commands and show you the very basics of the build-up procedure. We assume you are familiar with basic operating system commands.

Configuration

The most important thing to do before you start is to provide FlexNovo with a config.dat file in the working directory. This file tells FlexNovo the location of files that it needs at runtime and that are important for correct execution, e.g. template files for the receptor amino acids etc., helper programs such as a graphic visualizer (e.g. FlexV, our free 3D graphics program supplied with FlexNovo) or a text editor of your choice.

Create a working directory where no valuable data can be destroyed and copy the example config.dat file (located in <flexnovo_installation_dir>/example/config.dat) into this directory. Open the config.dat file with a text editor of your choice. The file is separated into 9 subsections, each one marked with a '@':

```
@LICENSE_FILES
@ROOTDIR
@DIRECTORIES
```

```
@STATIC_DATA
@PROGRAMS
@FLAGS
@ID_STRINGS
@PARALLEL
@ALIASES
```

In this tutorial, we only need to tinker with @ROOTDIR; we assume you have already entered the information about the license file in @LICENSE_FILES.

In order for FlexNovo to find the important files for error-free execution, the @ROOTDIR should point to the directory where you installed the software. Let's assume that this is your home directory, /home/user. In this case, @ROOTDIR should be /home/user/flexnovo.

FlexNovo should now find the location of static_data, objects etc.

In this tutorial we use FlexV as a molecular viewer, therefore you must make sure that under @PROGRAMS the path points to its executable. Alternatively, you can create a link in the <flexnovo_installation_dir>/bin to wherever you have installed FlexV, the path to FlexV would then be the same as to FlexNovo.

You should now have a readily configured config.dat for this tutorial.

Please note: FlexNovo does not come with a 3D generator, so the path in @PROGRAMS following RCGENERATOR, 3DGENERATOR and CONV_SMILES is not configured. You should point these three variables to the location of your own 3D generator (e.g. CORINA, CONFORD, CONCORD etc.) and SMILES conversion tool respectively. Without this information, FlexNovo cannot use the ring torsion feature and is not able to convert SMILES strings.

When you type flexnovo in your working directory now, FlexNovo will say HI, and the command prompt will be waiting for your input:

```
-----
                                F l e x N o v o
ZBH - Center for                Structure-Based Searching in Fragment Spaces
Bioinformatics
University of Hamburg          Version:  1.4.2 Beta   (25.02.08)
Bundesstrasse 43               Modules:  [PVM] [PHARM]
20146 Hamburg
Germany                         Authors:  Joerg Degen, Matthias Rarey

BioSolveIT GmbH                Copyright: ZBH, University of Hamburg, Germany
An der Ziegelei 75              BioSolveIT GmbH, Sankt Augustin, Germany
53757 St. Augustin             Contact:  flexnovo@zbh.uni-hamburg.de
Germany                        www.zbh.uni-hamburg.de/flexnovo
-----

For information about additional contributors and copyright notes
please consult the user guide or type 'help about'.

>> Running on <your_computer> (Linux...) with N processor(s).
>> FlexNovo configuration file 'your_project_dir/config.dat' loaded.
```

```

>> FlexNovo_base license check (BioSolveIT keys): succeeded.
>> Licensed modules: FlexNovo [PVM] [PHARM]
>> PVM status: USE_PVM_FEATURE is set to 0, running sequential.
>> SETTINGS = '<your_home_dir>/flexnovo/static_data/flexnovo_settings.dat' loaded.
>> CHEMPAR = '<your_home_dir>/flexnovo/static_data/chempar.dat' loaded.
>> CONTYPE = '<your_home_dir>/flexnovo/static_data/contype.dat' loaded.
>> GEOMETRY = '<your_home_dir>/flexnovo/static_data/geometry.dat' loaded.
>> AMINO = '<your_home_dir>/flexnovo/static_data/amino.dat' loaded.
>> CHARGES = '<your_home_dir>/flexnovo/static_data/amino_pcharges.dat' loaded.
>> FCHARGES = '<your_home_dir>/flexnovo/static_data/fcharges.dat' loaded.
>> DELOC = '<your_home_dir>/flexnovo/static_data/delocalized.dat' loaded.
>> CONTACT = '<your_home_dir>/flexnovo/static_data/contact.dat' loaded.
>> TORSION = '<your_home_dir>/flexnovo/static_data/torsion_standard.dat' loaded.
>> GAUSSIAN = '<your_home_dir>/flexnovo/static_data/gaussian.dat' loaded.
>> LOGP = '<your_home_dir>/flexnovo/static_data/logp.dat' loaded.
>> GRAPHIC = '<your_home_dir>/flexnovo/static_data/graphic.dat' loaded.
Process time used: 0.67 s. Current process size: 73852 kB.
FLEXNOVO>

```

For this tutorial introduction, we will first load a ligand, then a receptor. Normally, the ligand is obsolete when performing calculations with fragment spaces, but since we do not yet have the information about which atoms make up the active site, we will take the ligand as it is in the reference pdb file, and create a sphere of 6.0 Å radius around every atom of the ligand.

Subsequently, all protein atoms encountered within this sphere are considered active site atoms. If we answer 'yes' to the question `...complete amino acids ...?` (see below), then the set of already selected active site atoms is augmented. We also chose the remaining amino acid atoms of the ones that have already partially been selected by the sphere. In consequence of all this, it is necessary to start with a ligand.

After having defined the active site, we will load a simple fragment space and perform some basic operations with it. Then we will place fragments in the active site of the receptor and use these as start placements for the build-up calculation later on.

The ligand

In this context, we will load a ligand and assign reference coordinates to it in order to be able to define the active site later on.

Let us read in the ligand structures now: Type `ligand`. FlexNovo changes to the ligand submenu:

```

FLEXNOVO> ligand
FLEXNOVO/LIGAND>

```

Now read in the ligand structure as a `mol2` file. Depending on your settings (level of verbosity etc.) you will receive some warnings, errors or nothing at all.

Another thing to remember: Unlike other programs, FlexNovo does not need partial charges. For scoring, FlexNovo needs *formal* charges. This is why you should get rid of all partial charges before reading in your ligand or fragment space respectively. Unlike FlexX Release 2, the TRANSFORM functionality is not available in FlexNovo at the moment.

A Die Benutzungsschnittstelle von FlexNovo

```
FLEXNOVO/LIGAND> read ligand_minimized.mol2
>> Attempting to index multi-mol file..
>> File '4dfr_min' contains 1 compounds.
>> Number of ring systems: 2
>> SYBYL atom types already assigned. Checking assignment.
>> Atom type assignment step 1: Determining hybridization of atoms.
>> Atom type assignment step 2: Assigning non-aromatic atom types.
>> Atom type assignment step 3: Assigning aromatic atom types.
>> Ringsystem 1, ring 2 is aromatic
>> Ringsystem 2, ring 1 is aromatic
>> Automatic assignment of formal charges
>> Converting localized to delocalized systems
>> Ring 1: conformations computed by CORINA.
>> Ring 2: conformations computed by CORINA.
>> Potential number of conformations: 3.919104e+07
>> Number of components: 11
>> Ligand 'mtx' loaded from file './example/lig/ligand_minimized.mol2'.
    Current process size: 73988 kB
FLEXNOVO/LIGAND>
```

Next, we will read the same ligand as a reference. As mentioned above, this is necessary for FlexNovo to calculate RMSD values and for generation of the active site.

```
FLEXNOVO/LIGAND> readref ligand_as_in_pdb.mol2
>> Set reference coordinates by separate mol2 file
    Ignore hydrogen atoms [y] :
>> Ligand reference coordinates loaded from file './example/lig/ligand_as_in_pdb.mol2'.
    Current process size: 73922 kB
FLEXNOVO/LIGAND>
```

As you can see, FlexNovo ignores the hydrogens by default. If you type [n], FlexNovo will protonate the reference as well (not recommended).

If you would like to visualize the ligand(s), you may type **draw fix** (*fix* stands for fixed coordinates), and/or **draw ref** (*ref* stands for reference), press Return, followed by **display** or the shortcut **go** to make FlexV pop up. Click on the eye in FlexV to center the molecule.

```
FLEXNOVO/LIGAND> draw fix
>> Ligand drawn to graphics object 5
FLEXNOVO/LIGAND>
```

```
FLEXNOVO/LIGAND> draw ref
>> Ligand drawn to graphics object 3
    Current process size: 74124 kB
FLEXNOVO/LIGAND> display
FLEXNOVO/LIGAND>
```

You can see that the two molecules — though being the same compounds — do not overlap. The ligand loaded second is still missing its surrounding protein, but we will take care of that in a minute.

The receptor

Generating an rdf file To make the initial work with FlexNovo easier, a default `rdf` file is included in the `static_data` directory as a template which is directly used if a `pdb` file is read as a receptor instead of an `rdf` file. The receptor description file contains the location of the protein `pdb` file and provides FlexNovo with all the details about an active site (the pocket) — if it exists. This procedure can be used to get first results quickly, and to get a first impression for example if a new complex structure is added to the Brookhaven Protein database. To get information about the `pdb` file, you can use the command `pdbinfo` again. After obtaining this information, we can now actually read in the receptor.

```
FLEXNOVO/LIGAND> receptor
FLEXNOVO/RECEPTOR> read 4dfr_hoh.pdb
>> PDB file given, using generic rdf file
>> Interactive determination of active site
  Selection radius (Å) [6.5] : 6
  Select always complete amino acids [y] : n
>> 125 atoms active in site.
  ILE A   5 (ile ); ALA A   6 (ala ); ALA A   7 (ala ); ASP A  27 (asp- );
  LEU A  28 (leu ); ALA A  29 (ala ); TRP A  30 (trp ); PHE A  31 (phe );
  LYS A  32 (lys+ ); THR A  35 (thr ); LEU A  36 (leu ); THR A  46 (thr );
  SER A  49 (ser ); ILE A  50 (ile ); ARG A  52 (arg+ ); LEU A  54 (leu );
  PRO A  55 (pro ); ARG A  57 (arg+ ); VAL A  93 (val ); ILE A  94 (ile );
  GLY A  95 (gly ); TYR A 100 (tyr ); TYR A 111 (tyr ); LEU A 112 (leu );
  THR A 113 (thr ); ALA B 145 (ala );
>> Compute surface atoms, please wait ...
  Interaction points (Level 3 2 1):  965 ( 400 186 379)
  Unbound receptor SAS: 15588.289 Å2 lipophilic SAS: 8184.434 Å2
>> Number of active site atoms: 125
>> Receptor loaded from '4dfr_hoh.pdb'.
  Process time used: 1.30 s.   Current process size: 78992 kB
FLEXNOVO/RECEPTOR>
```

With the help of the reference ligand, we enquote cut out an active site definition from the surrounding `pdb` file. FlexNovo asks within which radius of the reference ligand it should create the active site. 6.5 Å is the default value that we can safely accept in the example. The next question FlexNovo asks is whether it should choose complete amino acids or not. Unless you have a specific reason not to choose complete amino acids, just type `return here`. Depending of the level of verbosity, FlexNovo outputs information about the number of active site atoms, the SAS (solvent accessible surface) and lipophilic SAS. Once this is done, we can write the receptor, pocket and surface information to their respective files. This is done with the `write` command.

```
FLEXNOVO/RECEPTOR> write a 4dfr_hoh_poc65.pdb n
>> Receptor written to file './example/receptor/site/4dfr_poc65.pdb'.
FLEXNOVO/RECEPTOR> write s 4dfr_hoh_surf.sdf
>> Receptor written to file './example/receptor/site/4dfr_surf.sdf'.
FLEXNOVO/RECEPTOR>
```

`a` stands for (a)ctive site and `s` for (s)urface. You can get this menu also by simply typing `write` and then `<Return>`, or use `help write`.

After we have written the files, we can adapt the `rdf` file to our special needs. We advise the user in production to **carefully inspect** the files, since this is where all information about the active site is given and modified etc. Also, protonation-dependent rules may be given here.

You may want to use `draw` and `go` for piping the necessary visualization data to FlexV (it may be in the background, so bring it to the front and move your mouse across the canvas).

```
FLEXNOVO/RECEPTOR> draw
>> Receptor drawn to graphics object 4
    Current process size: 79256 kB
FLEXNOVO/RECEPTOR> display
FLEXNOVO/RECEPTOR>
```

It should be visible that your reference ligand enquotesits nicely in the pocket. We are now done with both the ligand and the receptor and may proceed to the more important stages of the computation. Beforehand the ligand information can be deleted.

```
FLEXNOVO/RECEPTOR> ligand
FLEXNOVO/LIGAND> delete
FLEXNOVO/LIGAND>
```

The pharmacophore

As you will see later on, the placements of the starting fragments may be quite a **sensitive point in the buildup procedure!** Therefore the use of pharmacophore type constraints for generating the initial fragment poses is highly recommended, if possible.

Read the pharmacophore definition from the corresponding example file:

```
FLEXNOVO/LIGAND> pharm
FLEXNOVO/PHARM> read 4dfr_anchor
    INTERACTION: (ess)
        h_acc on atom OD1 ASP 27 A direction 1
>> All constraints must be fulfilled.
FLEXNOVO/PHARM>
```

The fragment space

To change to the fragspace menu type `fragspace`:

```
FLEXNOVO/PHARM> fragspace
FLEXNOVO/FRAGSPACE>
```

Now read in the fragspace definition file. Depending on your settings you will receive some warnings, errors or nothing at all.

```
FLEXNOVO/FRAGSPACE> read 4dfr
    Fragment file: 4dfr/mtx_fmin.mol2           :      3 fragment(s) found.
>> Fragment space loaded from file 4dfr:
    Link types found: 3 [ R1 R2 R4 ]
```

```

Fragment files found: 1 #fragments found: 3
>> Links found:
Link:   R1   R2   R4
#   :    1    2    1
>> Fragment space loaded from file '4dfr.fsf'.
>> Current pharmacophore settings used
FLEXNOVO/FRAGSPACE>

```

We can get some additional information by using the command `info`.

```

FLEXNOVO/FRAGSPACE> info
>> Fragment Space:
Filename: 4dfr.fsf
# link types :      3
  links :      R1   R2   R4
# links :      1    2    1
total   : 4

Link compatibility matrix:
   R1  R2  R4
R1  .  x  x
R2  x  .  x
R4  x  x  .

Terminal group information:
link group      aatom  bond  blen  tor
R1  4dfr/r1.mol2  *      1  1.280  *
R2  4dfr/r2.mol2  *      1  1.009  *
R4  4dfr/r4.mol2  *      1  1.090  *

Connection group information:
link1 link2 aatom1 aatom2 bond  blen  tor
R1    R2   C.3   N.p13  am  1.355 180.0
R1    R4   C.2   C.3    1  1.507  *
R2    R4   C.2   N.p13  1  1.465  *

Total # fragments      : 3
4dfr/mtx_fmin.mol2    : 3

Link histogram:
#links : 0 1 2 3 4 5 6 7 8 9 10
#frags : 0 2 1 0 0 0 0 0 0 0 0

fragment space size:
max #fragments : 1 2 3 4 5
#molecules    : 3.00e+00 8.00e+00 1.70e+01 2.90e+01 3.50e+01
Comp 1: R1 R2 R4 : infinite
FLEXNOVO/FRAGSPACE>

```

Before starting to work with a 'new' fragment space, the fragments contained should be **carefully inspected**, in particular with respect to the geometry, charges, sybyl atom and bond types and the rules for the link compatibility and terminal group definitions in the input file.

First, we will start the internal FlexNovo initialization functions that assign additional data to the fragments loaded from the input file(s). Type `init` to start the initialization.

```
FLEXNOVO/FRAGSPACE> init
```

A Die Benutzungsschnittstelle von FlexNovo

```
Initialization mode (a= all, s= selection) [a] :
>> Initializing mtx1
>> Number of ring systems: 1
>> SYBYL atom types already assigned. Checking assignment.
>> Atom type assignment step 1: Determining hybridization of atoms.
>> Atom type assignment step 2: Assigning non-aromatic atom types.
>> Atom type assignment step 3: Assigning aromatic atom types.
>> Ringsystem 1, ring 2 is aromatic
>> Automatic assignment of formal charges
>> Converting localized to delocalized systems
>> Ring 1: conformations computed by CORINA.
>> Potential number of conformations: 1.200000e+01
>> Number of components: 2
>> Initializing mtx2
>> Number of ring systems: 0
>> SYBYL atom types already assigned. Checking assignment.
>> Atom type assignment step 1: Determining hybridization of atoms.
>> Atom type assignment step 2: Assigning non-aromatic atom types.
>> Atom type assignment step 3: Assigning aromatic atom types.
>> Automatic assignment of formal charges
>> Converting localized to delocalized systems
>> Potential number of conformations: 1.008000e+04
>> Number of components: 6
>> Initializing mtx3
>> Number of ring systems: 1
>> SYBYL atom types already assigned. Checking assignment.
>> Atom type assignment step 1: Determining hybridization of atoms.
>> Atom type assignment step 2: Assigning non-aromatic atom types.
>> Atom type assignment step 3: Assigning aromatic atom types.
>> Ringsystem 1, ring 1 is aromatic
>> Automatic assignment of formal charges
>> Converting localized to delocalized systems
>> Ring 1: conformations computed by CORINA.
>> Potential number of conformations: 1.200000e+01
>> Number of components: 3
FLEXNOVO/FRAGSPACE>
```

As there are no warnings or errors shown, the fragment data seems to be ok.

Please note: The initialization procedure used depends on the flags that are either set in the `config.dat` file or modified by hand!

In a second step, we will check if there are fragments that might undergo configurational changes with respect to the configuration of Nitrogen atoms upon fragment connection or link substitution. This is important since each fragment is represented in one (input) conformation only and, therefore, each (pseudo) stereo center in one configuration only. Type `check` to start the process.

```
FLEXNOVO/FRAGSPACE> check
>> Checking for potential configurational changes of Nitrogen atoms
>> Fragment mtx1
>> Fragment mtx2
>> WARNING: Mismatching geometry of atom and connection rule found:
1|R2 -am- 2|N( N.p13|SP2 ) in ring: 0
  link1 link2 aatom1 aatom2 bond blen tor
    R2   R4   C.2   N.p13   1   1.465  *
                                     ~
>> WARNING: Mismatching geometry of atom and terminal fragment connection found:
1|R2 -am- 2|N( N.p13|SP2 ) in ring: 0
```

```

link  group          aatom  bond  blen  tor
R2   [H]             *      1   1.009  *

```

>> Fragment mtx3
FLEXNOVO/FRAGSPACE>

From this output we can see that there is a Nitrogen atom adjacent to a link atom in the fragment mtx2. This Nitrogen has planar geometry but there is a fragment connection and a link substitution rule that imply a configurational change. Although this may not cause any problems for other link connection rules that, as a consequence, are not listed above, it may lead to a bad geometry when connecting two fragments by applying this particular rule above. For the **input coordinates** of the fragments, this can be accounted for by setting the parameter <USE_GEOM_CORRECTION> in the chempar.dat file to 1. However, **this cannot be used during the build-up calculation and, as a consequence, may lead to bad geometries!**

As you can see, even for this small tutorial example, there is a non unique mapping of fragment input geometries and connection or link substitution rules respectively. It is of importance that this should be accounted for in the best and most convenient way possible, for example, by introducing separate link types for planar and tetraetrical Nitrogen configurations.

Please note: The example fragment space used in this tutorial was deliberately generated to produce the above warnings in order to point out the importance of a consistent handling of fragment geometries. For the sample fragment space included in this release, the issue mentioned above has already been taken care of.

As a last step, you can also inspected the fragments visually and may type `mdraw` followed by `display` or the shortcut `go` to make FlexVpop up. Click on the eye in FlexV to center the fragment. It may be necessary to do this for each fragment individually if the input coordinates were not 'centered' beforehand.

```

FLEXNOVO/FRAGSPACE> mdraw
Drawing mode (a= all, s= selection) [a] :
>> Fragment mtx1 drawn to /tmp/flexv_tmp_16.gdf
>> Fragment mtx2 drawn to /tmp/flexv_tmp_16.gdf
>> Fragment mtx3 drawn to /tmp/flexv_tmp_16.gdf
FLEXNOVO/FRAGSPACE> display
FLEXNOVO/FRAGSPACE>

```

Working with fragment spaces

Generate a selection of all fragments by typing `select` and taking default values for all the selection criteria.

```

FLEXNOVO/FRAGSPACE> select
Fragment id range (format:a b) [not_def] :
File id range (format:a b) [not_def] :
Mol id range (format:a b) [not_def] :
Nof atoms range (format:a b) [not_def] :
Nof links range (format:a b) [not_def] :
Links: 0:R1 1:R2 2:R4

```

A Die Benutzungsschnittstelle von FlexNovo

```
Required links (format: a b ...) [not_def] :
Fragment name substring [] :
>> 3 hits found.
FLEXNOVO/FRAGSPACE>
```

Type `flist` to print a list of all fragments in the selection on the screen.

```
FLEXNOVO/FRAGSPACE> flist
Format <0:header, 1:line> <0,1> [1] : 0
  No   | Fragment name      | file  mol | #L | Links
-----|-----|-----|-----|-----|-----
FLEXNOVO/FRAGSPACE> flist
Format <0:header, 1:line> <0,1> [1] : 1
Print mode (a= all, s= selection) [a] : s
  0 | mtx1      | 0     0 | 1 | R4
  1 | mtx2      | 0     1 | 1 | R2
  2 | mtx3      | 0     2 | 2 | R1 R2
FLEXNOVO/FRAGSPACE>
```

Type `filter` to change to the filter submenu. Then load the filter definition from the file `prefilter.fsf`.

```
FLEXNOVO/FRAGSPACE> filter
FLEXNOVO/FILTER> read prefilter.fsf
>> Filter ruleset:
  Filename: prefilter.fsf
  rule violation tolerance : 0

  molecular weight   range: 100 - 250
  # of rings         range: undefined
  # of rotatable bonds range: undefined
  # of h_bond acceptor range: undefined
  # of h_bond donor   range: undefined
  # of stereo centers range: undefined
  polar SAS         range: undefined
  calculated logP    range: undefined
  molar refractivity range: undefined
  exclusion SMARTS   : undefined
  inclusion SMARTS   : undefined
>> Filter ruleset loaded from file 'prefilter.fsf'.
FLEXNOVO/FILTER>
```

Apply the filter rules above to the fragment selection in order to make sure that there isn't any fragment with a molecular weight above 250 Daltons in it.

```
FLEXNOVO/FILTER> apply
  Filter mode (a= all, s= selection) [a] : s
>> Checking for initialization state and fragment descriptors
  Initializing if necessary...
  Calculating descriptors if necessary...
>> Applying filter rules
>> 3 fragments remaining in selection
FLEXNOVO/FILTER>
```

In our example case all three fragments remain in the selection. Now it is time to compute starting orientations for each fragment.

Placing fragments

Type `placing` to change to the respective submenu.

```
FLEXNOVO/FILTER> placing
FLEXNOVO/PLACING>
```

The standard FlexX docking algorithm is used for placing all fragments of the selection in the active site of the protein. For usage later on, all placement information can be written to files. We will use the standard directory for the example. As mentioned earlier, it is much more appropriate to do the calculation under pharmacophore type constraints in order to get more reasonable starting positions. Since the pharmacophoric constraints are already loaded we can start straight away:

```
FLEXNOVO/PLACING> place
  Placement mode (a= all, s= selection) [a] : s
  Placement algorithm (3= triangle, 2= line) [3] :
  Storing placement info to disk [n] :
  Storing solutions to disk [n] : y
  Coding transformation matrices [y] :
  PDF Path (default directory: your_project_dir/predict/) :
  Keeping solutions in memory [n] :
  >> No receptor triangle hash table, generate one ...
  >> Placing mtx1
  [...]
FLEXNOVO/PLACING>
```

We will do the above calculation additionally for all fragments of the fragment space. Please note that this calculation should be done without any pharmacophore type constraints in order to get as much placement information for as much fragments as possible. Therefore, the pharmacophore constraints have to be deleted again. Please note that this information is used for score estimation only (see `INIT` command later in the tutorial). Therefore we will not save the placements themselves, but write the score of each fragment to a file only.

```
FLEXNOVO/PLACING> pharm
FLEXNOVO/PHARM> delete
FLEXNOVO/PHARM> placing
FLEXNOVO/PLACING> place
  Placement mode (a= all, s= selection) [a] :
  Placement algorithm (3= triangle, 2= line) [3] :
  Storing placement info to disk [n] : y
  Filename(use explicit suffix) (default directory: your_project_dir/predict/) [scores] : scores.txt
  Table format [n] :
  Output table <0:standard, 1:rms, 2:rank, 3:acc rms, 4:rms history, 5:frag score, 6:screen,
  7:score> <0,7> [7] : 7
  Storing solutions to disk [n] :
  Keeping solutions in memory [n] :
  >> Placing mtx1
  [...]
FLEXNOVO/PLACING>
```

All fragments could be successfully placed and the corresponding score information was written to text file. Now the preprocessing phase is completed and we can start working with the build-up algorithm. We want to see a little bit more information

during the calculation, therefore we set the verbosity level to a higher value. However, **it is not recommended to set the verbosity to a value higher than 4 for larger calculations!**

```
FLEXNOVO/PLACING> set verbosity 4
>> Data may be inconsistent (RELOAD?).
FLEXNOVO/PLACING>
```

The build-up procedure

We will use all fragments of the example fragment space as starting fragments for the buildup algorithm. Therefore, we first have to load the corresponding placement information in order to have starting orientations. We will use the ones previously generated under pharmacophore type constraints:

```
FLEXNOVO/PLACING> load
  Load placements for (a= all, s= selection) [a] : s
  PDF Path (default directory: your_project_dir/predict/) :
>> Loading up to 60 solutions per fragment
>> Placements loaded from file your_project_dir/predict/mtx1.pdf
>> 1 fragment placement(s) loaded.
FLEXNOVO/PLACING>
```

In order to have a good score evaluation for all fragments, the `init` command should be used to estimate the best scores for each fragment, if these were computed earlier. This data will be taken from the `scores.txt` file.

```
FLEXNOVO/PLACING> init
  Filename(use explicit suffix) (default directory: your_project_dir/predict/) scores.txt
  mtx1 : -24.943
  mtx2 : -22.813
  mtx3 : -11.101
>> 3 fragment placement scores loaded from file scores.txt
FLEXNOVO/PLACING>
```

There are probably different filter criteria necessary for the build-up process than the ones which were used during the preprocessing phase. Therefore, the filter definitions should be replaced.

```
FLEXNOVO/PLACING> filter
FLEXNOVO/FILTER> read filter_buildup.fsf
>> Filter ruleset:
  Filename: filter_buildup.fsf
  rule violation tolerance : 0

  molecular weight   range: 100 - 460
  # of rings         range: 0 - 4
  # of rotatable bonds range: 0 - 12
  # of h_bond acceptor range: 0 - 13
  # of h_bond donor  range: 0 - 7
  # of stereo centers range: undefined
  polar SAS          range: undefined
  calculated logP    range: -2.5 - 2.0
  molar refractivity range: undefined
  exclusion SMARTS   : undefined
  inclusion SMARTS   : undefined
>> Filter ruleset loaded from file 'filter_buildup.fsf'.
FLEXNOVO/FILTER>
```

Next, we will load the pharmacophore specification that will be used during the buildup algorithm.

```
FLEXNOVO/FILTER> pharm
FLEXNOVO/PHARM> read 4dfr_anchor
  INTERACTION: (ess)
    h_acc on atom  OD1  ASP  27  A direction 1
  INTERACTION: (opt)
    h_acc on atom  OD2  ASP  27  A direction 1
  INTERACTION: (ess)
    h_acc on atom   0   ILE   5  A
  INTERACTION: (ess)
phenyl_center on atom  CG   PHE  31  A direction 1
  INTERACTION: (opt)
    h_don on atom  NH2   ARG  57  A
>> Partial match only: min 1, max 2 matches required
FLEXNOVO/PHARM>
```

Now we can start the build-up algorithm. The starting fragments and the starting placements used for each fragment are of crucial importance for the algorithm. For the tutorial introduction we will use standard parameters for the computation.

```
FLEXNOVO/PHARM> buildup
FLEXNOVO/BUILDUP> greedy
  Take start fragments from (a= all(placed), s= selection, p= partial results) [a] : s
>> Create initial fragment queue
  initial queue size: 1
>> Clearing fragment space of unused fragment placement data
-----
>> Greedy sequential buildup
  max. # of extension cycles      : 2
  max. # of fragments per cycle   : 100
  max. # of buildup solutions     : 100
  max. # of placements per molecule : 60
  max. # of placements per solution : 10
  max. # of common fragments(results): 1
  max. # of each fragment per cycle : 10
  score estimation mode           : 1
  link resolving mode             : 1
-----
>> Entering extension cycle: 1
-----
>> Expanding fragment : mtx1
  dock entry with | no. |norm. E.|total E.| rms|#M | overlap (ave.)
-----
  min. norm. energy |  1| -31.743| -24.943| inf| 26| 0.14 ( 0.02)
  min. energy      |  1| -31.743| -24.943| inf| 26| 0.14 ( 0.02)
  min. rms         |  1| -31.743| -24.943| inf| 26| 0.14 ( 0.02)
  no solution with rms < 1.5
  Deriving 'reference' pharmacophore information
  Constraints fully satisfied: 1/3(ess) 1/2(opt)
-----
>> Building : mtx1-(R4|14) - (R1|12)-mtx3
  Generating 'local' pharmacophore information
  Master list contains 2838 candidate counter group sets (162 rejected)
  dock entry with | no. |norm. E.|total E.| rms|#M | overlap (ave.)
-----
  min. norm. energy |  1| -41.850| -32.250| 2.21| 34| 0.74 ( 0.09)
  min. energy      |  1| -41.850| -32.250| 2.21| 34| 0.74 ( 0.09)
  min. rms         | 380| -26.469| -16.869| 1.21| 32| 0.63 ( 0.08)
```

A Die Benutzungsschnittstelle von FlexNovo

```
first with rms<1.5| 70| -35.392| -25.792| 1.34| 26| 2.35 ( 0.34)
Number of solutions 380 (before) 60 (after)
-----
>> Building : mtx1-(R4|14) - (R2|1)-mtx2
Generating 'local' pharmacophore information
Master list contains 1478 candidate counter group sets (34 rejected)
No solution found.
-----
>> Building : mtx1-(R4|14) - (R2|1)-mtx3
Generating 'local' pharmacophore information
Master list contains 2838 candidate counter group sets (162 rejected)
dock entry with | no. |norm. E.|total E.| rms|#M| overlap (ave.)
-----
min. norm. energy | 1| -42.259| -32.659| 2.01| 35| 2.42 ( 0.27)
min. energy | 1| -42.259| -32.659| 2.01| 35| 2.42 ( 0.27)
min. rms | 329| -30.943| -21.343| 1.04| 34| 1.13 ( 0.20)
first with rms<1.5| 30| -36.018| -26.418| 1.47| 31| 0.62 ( 0.06)
Number of solutions 337 (before) 60 (after)
-----
Summary of extension cycle 1:
-----
# of fragments evaluated in total : 3
# of fragments placed successfully : 2
# of fragments filtered (props) : 0
# of fragments filtered (pharm) : 0
# of fragments skipped (score) : 0
# of fragments skipped (error) : 0
# of placements evaluated : 1237
# of placements generated : 717
# of placements filtered (geom) : 254
# of placements filtered (pharm) : 20 ( 0| 20| 0)
-----
>> Entering extension cycle 2:
-----
>> Expanding fragment : mtx1-mtx3
dock entry with | no. |norm. E.|total E.| rms|#M| overlap (ave.)
-----
min. norm. energy | 1| -42.259| -32.659| 2.01| 35| 2.42 ( 0.27)
min. energy | 1| -42.259| -32.659| 2.01| 35| 2.42 ( 0.27)
min. rms | 33| -35.894| -26.294| 1.29| 25| 0.77 ( 0.07)
first with rms<1.5| 30| -36.018| -26.418| 1.47| 31| 0.62 ( 0.06)
Deriving 'reference' pharmacophore information
Constraints fully satisfied: 2/3(ess) 1/2(opt)
-----
>> Building : mtx1-mtx3-(R1|32) - (R2|1)-mtx2
Generating 'local' pharmacophore information
Master list contains 6468 candidate counter group sets (5628 rejected)
dock entry with | no. |norm. E.|total E.| rms|#M| overlap (ave.)
-----
min. norm. energy | 1| -70.200| -53.600| 2.55| 46| 1.27 ( 0.13)
min. energy | 1| -70.200| -53.600| 2.55| 46| 1.27 ( 0.13)
min. rms | 28| -62.268| -45.668| 2.13| 43| 1.24 ( 0.12)
no solution with rms < 1.5
Number of solutions 48
Checking placements against 'global' pharm filter.
Master list contains 6468 candidate counter group sets (5628 rejected)
0 placements destroyed, remaining: 48
-----
>> Building : mtx1-mtx3-(R1|32) - (R2|1)-mtx3
Generating 'local' pharmacophore information
Master list contains 2948 candidate counter group sets (3927 rejected)
dock entry with | no. |norm. E.|total E.| rms|#M| overlap (ave.)
```

```

-----
min. norm. energy | 1| -42.442| -31.442| 2.00| 40| 2.49 ( 0.38)
min. energy       | 1| -42.442| -31.442| 2.00| 40| 2.49 ( 0.38)
min. rms          | 41| -32.139| -21.139| 1.75| 45| 2.50 ( 0.56)
no solution with rms < 1.5
Number of solutions 45
Resolving open link atom(s)
Checking placements against 'global' pharm filter.
Master list contains 3828 candidate counter group sets (4422 rejected)
dock entry with  | no. |norm. E.|total E.| rms|#M | overlap (ave.)
-----
min. norm. energy | 1| -43.164| -32.164| 1.80| 52| 2.53 ( 0.36)
min. energy       | 1| -43.164| -32.164| 1.80| 52| 2.53 ( 0.36)
min. rms          | 4| -38.374| -27.374| 0.35| 47| 2.12 ( 0.34)
first with rms<1.5| 3| -39.241| -28.241| 1.15| 53| 2.37 ( 0.41)
Number of solutions 5
-----
>> Building : mtx1-mtx3-(R1|32) - (R4|14)-mtx1
Molecular weight limit exceeded: 485.5 (100-460)
Number of rings limit exceeded: 5 (0-4)
Hydrogen bond acceptors limit exceeded: 14 (0-13)
Hydrogen bond donors limit exceeded: 10 (0-7)
-----
>> Finalizing fragment: mtx1-mtx3
Resolving open link atom(s)
Checking placements against 'global' pharm filter.
Master list contains 1608 candidate counter group sets (2142 rejected)
dock entry with  | no. |norm. E.|total E.| rms|#M | overlap (ave.)
-----
min. norm. energy | 1| -46.170| -36.570| 2.00| 38| 2.49 ( 0.28)
min. energy       | 1| -46.170| -36.570| 2.00| 38| 2.49 ( 0.28)
min. rms          | 4| -38.922| -29.322| 1.77| 40| 1.55 ( 0.16)
no solution with rms < 1.5
Number of solutions 5
-----
>> Expanding fragment : mtx1-mtx3
dock entry with  | no. |norm. E.|total E.| rms|#M | overlap (ave.)
-----
min. norm. energy | 1| -41.850| -32.250| 2.21| 34| 0.74 ( 0.09)
min. energy       | 1| -41.850| -32.250| 2.21| 34| 0.74 ( 0.09)
min. rms          | 26| -38.613| -29.013| 1.52| 34| 1.62 ( 0.13)
no solution with rms < 1.5
Deriving 'reference' pharmacophore information
Constraints fully satisfied: 2/3(ess) 1/2(opt)
-----
>> Building : mtx1-mtx3-(R2|22) - (R1|12)-mtx3
Generating 'local' pharmacophore information
Master list contains 2860 candidate counter group sets (4015 rejected)
dock entry with  | no. |norm. E.|total E.| rms|#M | overlap (ave.)
-----
min. norm. energy | 1| -45.570| -34.570| 2.14| 37| 2.11 ( 0.19)
min. energy       | 1| -45.570| -34.570| 2.14| 37| 2.11 ( 0.19)
min. rms          | 147| -33.436| -22.436| 0.31| 38| 1.80 ( 0.32)
first with rms<1.5| 94| -38.395| -27.395| 1.29| 38| 1.73 ( 0.22)
Number of solutions 222 (before) 60 (after)
Resolving open link atom(s)
Checking placements against 'global' pharm filter.
Master list contains 5962 candidate counter group sets (5918 rejected)
dock entry with  | no. |norm. E.|total E.| rms|#M | overlap (ave.)
-----
min. norm. energy | 1| -45.570| -34.570| 4.57| 37| 2.11 ( 0.19)
min. energy       | 1| -45.570| -34.570| 4.57| 37| 2.11 ( 0.19)

```

A Die Benutzungsschnittstelle von FlexNovo

```
min. rms      | 18| -40.188| -29.188| 0.24| 39| 1.62 ( 0.11)
first with rms<1.5| 18| -40.188| -29.188| 0.24| 39| 1.62 ( 0.11)
Number of solutions 20
-----
>> Building :  mtx1-mtx3-(R2|22) - (R4|14)-mtx1
Molecular weight limit exceeded: 485.5 (100-460)
Number of rings limit exceeded: 5 (0-4)
Hydrogen bond acceptors limit exceeded: 14 (0-13)
Hydrogen bond donors limit exceeded: 10 (0-7)
-----
>> Finalizing fragment:  mtx1-mtx3
Resolving open link atom(s)
Checking placements against 'global' pharm filter.
Master list contains 2328 candidate counter group sets (2856 rejected)
dock entry with  | no. |norm. E.|total E.| rms|#M | overlap (ave.)
-----
min. norm. energy |  1| -41.804| -32.204| 1.54| 34| 0.74 ( 0.09)
min. energy      |  1| -41.804| -32.204| 1.54| 34| 0.74 ( 0.09)
min. rms         | 11| -37.629| -28.029| 0.55| 34| 2.25 ( 0.23)
first with rms<1.5|  2| -41.166| -31.566| 1.32| 33| 0.74 ( 0.11)
Number of solutions 16
-----
Summary of extension cycle 2:
-----
# of fragments evaluated in total :      5
# of fragments placed successfully :     3
# of fragments filtered (props) :       2
# of fragments filtered (pharm) :       0
# of fragments skipped (score) :        0
# of fragments skipped (error) :        0
# of placements evaluated :            1485
# of placements generated :            361
# of placements filtered (geom) :       28
# of placements filtered (pharm) :     265 ( 242| 21| 2)
Process time used:  4.72 s.
FLEXNOVO/BUILDUP>
```

After a few seconds, the build-up calculation ends with a number of solutions constructed and ranked according to the scoring function. Print a list of the solutions on the screen using the command `listsol`:

```
FLEXNOVO/BUILDUP> listsol
number of solutions show <1,1> [1] :
FLEXNOVO BUILDUP SOLUTIONS
-----+-----+-----+-----+
|Rank| Molecule Name                | Score|#poses|
-----+-----+-----+-----+
|  1| mtx1-mtx3-mtx2                |-55.600|  10 |
-----+-----+-----+-----+
FLEXNOVO/BUILDUP>
```

Visualize the best solution with its best scoring placements according to scoring rank together with the receptor in FlexV:

```
FLEXNOVO/BUILDUP> mdraw
Drawing mode (a= all, s= selection) [a] :
Fragment selection <1-1> : 1
Placement selection <1-10> : all
Draw in multiple sliders [y] :
```

```
>> Fragment mtx1-mtx3-mtx2: placement 10 drawn to /tmp/flexv_tmp_4.gdf
>> Fragment mtx1-mtx3-mtx2: 10 placements drawn to graphics object 15.
Process time used: 0.10 s.
FLEXNOVO/BUILDUP> display
FLEXNOVO/BUILDUP>
```

Bring FlexV to the front and see how well you did!

Outputting information

You should by now be curious enough to see how good or bad it worked in detail. To this end, we would encourage you to use the built-in help function.

If you simply press **Return** at any time at an ordinary menu prompt, a list of commands available at this stage will be output.

Since we would like to get information about a special command called `info` under the buildup menu, we simply type `help info` at the buildup menu prompt:

```
FLEXNOVO/BUILDUP> help info
Listing main statistical build-up information (INFO)
[Syntax:] INFO
[Description:] Displays some statistical and timing information of the build-up
               calculation on the screen.
[Requirements:] Build-up results must be computed.
FLEXNOVO/BUILDUP>
```

This is what the output looks like. If you would like to see a summary and some general information about the calculation you just performed, type `INFO`, which outputs the information in table format:

```
FLEXNOVO/BUILDUP> info
GREEDY BUILDUP INFO
Receptor      : 4dfr_hoh
Fragospace    : 4dfr
-----
Number of solutions : 1
-----
Highest Ranking Solution
Molecule     : mtx1-mtx3-mtx2
Energy        : -53.600 kJ/mol

FlexNovo build-up history
-----
Extension cycle |          1          2
-----
# fragments evaluated |          3          5
# fragments placed   |          2          3
# fragments filtered (props) |          0          2
# fragments filtered (pharm) |          0          0
# fragments skipped (score) |          0          0
# fragments skipped (error) |          0          0
# placements evaluated |        1237        1485
# placements generated |          717        361
# placements filtered (geom) |         254         28
# placements filtered (pharm) |          20         265
-----
```

```

Runtime (elapsed process time)
-----
Preprocessing           : 0.00 s
Filtering (properties)  : 0.00 s
Fragment build-up      : 0.01 s
Fragment initialization : 0.23 s
Transferring solutions  : 0.01 s
Incremental construction : 3.63 s
Pharmacophore processing : 0.33 s
Postprocessing          : 0.46 s
FLEXNOVO/BUILDUP>

```

Please note: Although several fragments were actually successfully placed during the computation, the final solution list contains only one molecule. This is due to the diversity parameters for the calculation. The parameter <BUILDUP_RESULTS_DIVERSITY> is by default set to 1. This means that every molecule in the final solution list must have a maximum of one initial (input) fragment in common with every other solution. This leads to only one final solution in this case.

Additionally you can have a look the properties the solutions generated. Print a list of the corresponding values using the command `solinfo`:

```

FLEXNOVO/BUILDUP> SOLINFO
  Format <0:header, 1:line> <0,1> [1] : 0
Name                               | size      | topology          | interactions      [...]
|      +--- atoms| +----- rot. bonds| +----- type [...]
|      |      mass| | +----- R/S E/Z| | | | don a
|      |      | | | | +--- rings| | | | acc a
|      |      | | | | */comp| | | |
FLEXNOVO/BUILDUP> SOLINFO
  Format <0:header, 1:line> <0,1> [1] : 1
  Info mode (a= all, s= selection) [a] :
  mtx1-mtx3-mtx2                    | 33 453.44 | 8 0 1 3 2 1 11 | 14 10 6 6 13 | [...]
FLEXNOVO/BUILDUP>

```

Play around with the commands to get used to the procedures behind FlexNovo's help function

You should now have obtained an initial overview of the steps to take for constructing molecules out of a tiny fragment space for one protein. For more complex applications a script is usually more appropriate. Please refer to the index or the table of contents to get into more detail. Thank you for going through the tutorial!

Oh, BTW: You exit FlexNovo by saying `quit` or — if you dislike being asked if you are sure — by a simple `x`. And of course you can delete all files generated in the `predict` directory afterwards.

```

FLEXNOVO/BUILDUP> quit
  Are you sure you want to quit FLEXNOVO <y,n> [y] : y
  >> Releasing user data.
  >> Releasing system data.
  >> Bye!

```

A.2 FlexNovo-Einstellungen

In der zentralen Konfigurationsdatei `config.dat` sind die Pfade zu externen Programmen, den verwendeten Parameter-Dateien sowie Steuerungsparameter angegeben. Die für FlexNovo wichtigen Einstellungen werden im Folgenden erläutert.

A.2.1 Konfiguration

Abweichend von den (FlexX) Standard-Einstellungen müssen für FlexNovo einige spezifische Anpassungen in der Konfigurationsdatei vorgenommen werden, die im folgenden zusammengestellt sind.

Eine vollständige Konfigurationsdatei inklusive Dokumentation ist in Anhang A.4.1 zu finden.

```
@STATIC_DATA
SETTINGS      static_data/flexnovo_settings.dat

@PROGRAMS
CONV_SMILES   bin_Linux/corina -i t=smiles,dummies -o t=mol2,m2l -dwh

@FLAGS
# User interface
VERBOSITY     3
SIZE_LIMIT    0

# fragment data preparation
TEST_ATOM_TYPES  0
ADD_HYDROGENS    0
ASSIGN_FORMAL_CHARGES  1
ASSIGN_DELOCALIZED  1
KEEP_RCGEN_FILES  0
STEREO_MODE      7
SECONDARY_TORSION_MODE  0

#application FRAGSPACE
PDF_COMPRESSION  1
#end_application
```

A.2.2 Algorithmische Parameter

Die Einstellungen der Parameter für FlexNovo werden in einer gesonderten Datei abgelegt, typischerweise `flexnovo_settings.dat`. Darin finden sich die Parameter für den FlexNovo-Aufbaualgorithmus und die Einstellungen für die Filterfunktionen, die Platzierungen hinsichtlich zusätzlicher geometrischer Kriterien bewerten.

A.2.2.1 Programm-Parameter

Name (type): <NOF_BUILDUP_RESULTS> (integer)

Description: Number of solutions created by the buildup process. All solutions are kept in memory, so the amount of memory used can get very high depending on the value of this parameter.

Default value: 100

Reasonable range: 10 – 1000

Name (type): <NOF_BUILDUP_PLACEMENTS> (integer)

Description: Maximal number of placements kept per buildup solution. If equals -1 all solutions are kept.

Default value: 10

Reasonable range: -1, 10 – 100

Important notes: This parameter has to be set to higher values in case the buildup process is performed in a stepwise manner, meaning doing separate calculations for each extension cycle. Otherwise the number of start orientations is probably not sufficient for the buildup algorithm.

Name (type): <BUILDUP_EXTENSION_CYCLES> (integer)

Description: Number of extension cycles performed by the buildup process. Each molecule in the solutions list therefore consists of up to <BUILDUP_EXTENSION_CYCLES> + 1 fragments.

Default value: 2

Reasonable range: 1 – 10

Name (type): <BUILDUP_QUEUE_SIZE> (integer)

Description: Number of fragments in the queue for each extension cycle. The fragments used for the next extension cycle are all kept in memory, so the amount of memory used can get very high depending on the value of this parameter.

Default value: 100

Reasonable range: 1 – 200

Name (type): <BUILDUP_RESULTS_DIVERSITY> (integer)

Description: An efficient way to increase the diversity of the list of resulting compounds is to limit the number of fragments in common between two solutions. A positive value k means two solutions have a maximum of k fragments in common. A negative value of $-k$ means that two solutions differ by at least k fragments. If set to 0, no filtering is performed.

Default value: 1 (must have a maximum of one fragment in common)

Reasonable range: -10 – 10

Name (type): <BUILDUP_QUEUE_DIVERSITY> (integer)

Description: In addition to limiting the number of common fragments in the resulting molecule list, the number of maximum occurrences of each fragment in the queue for each extension cycle can be limited. If set to 0, no filtering is performed.

Default value: 10

Reasonable range: 0 – 100

Name (type): <BUILDUP_SCORE_ESTIMATION> (integer)

Description: During the buildup process, the score of each fragment in each extension cycle is estimated with respect to the scores of all other placed fragments. If equals 1, the fragment is discarded without doing any calculation if its estimated score is too low with respect to the already placed fragments. If set to 0, no evaluation is performed.

Default value: 1

Possible values: 0,1

Name (type): <BUILDUP_RESOLVE_LINKS> (integer)

Description: If equals 1, all non substituted link atoms in all molecules which are part of the final solutions list during the buildup process are replaced by the terminal group specified in the input file. Due to geometrical reasons, a solution already found can be discarded after replacing the link atom with its terminal group. Therefore these solutions are kept if the parameter is set to 2. If it is set to 0, no substitution is performed at all.

Default value: 1

Possible values: 0,1,2

Important notes: This parameter has to be set to 0 in case the buildup procedure is performed in a stepwise manner, meaning doing separate calculations for each extension cycle. Otherwise all compounds in the solutions list will not have any unsubstituted link atoms.

Name (type): <NOF_PLACEMENTS> (integer)

Description: Maximal number of placements kept per fragment/molecule. If equals -1 all solutions are kept.

Default value: 60

Reasonable range: -1, 10 – 500

Name (type): <USE_GEOM_CORRECTION> (integer)

Description: Due to the fact that, typically, each fragment is represented by one conformation only, a Nitrogen atom will either have a planar or a tetrahedral geometry. In the aliphatic case, this geometry might change upon the connection of fragments or the substitution of links with their terminal group. The geometry

can change either from planar to tetraetrical or the other way round. This depends on the rules specified in the `fsf` file. Note, that the geometry cannot be corrected at all if the Nitrogen is part of a ringsystem since this implies changes in the ringconformation.

In order to check for the compatibility of fragment geometries and the connection rules specified use the `CHECK` function (see section A.3.1.3) in the menu `FRAGSPACE`.

Default value: 0

Possible values: 0,1

Important notes: **The geometry correction works for the fragment input coordinates only!** This means that in case a partial solution of the build-up calculation is extended with another fragment, and the Nitrogen configuration changes upon this connection, the corresponding placements are not corrected. In case you obtain bad Nitrogen configurations in solution list molecules of the build-up calculation, this is a result of non-unique fragment geometries and connection rules. **Please note that this flag is still experimental!**

A.2.2.2 Filter-Funktionen für Platzierungen

Name (type): <FILTER_POLARITY> (integer)

Description: The polarity filter checks the fragment placements and perhaps erases some solutions. All placements resulting in close contacts of non-interacting equally charged polar groups will be discarded. If <FILTER_POLARITY> equals 1, the filter will be applied.

Default value: 1

Possible values: 0,1

Name (type): <FILTER_1_5_REPULSION> (integer)

Description: The repulsions filter checks the fragment placements and perhaps erases some solutions. All placements resulting in strained fragment conformations, e.g. due to a number of cis-oriented single-bonds, will be discarded. If <FILTER_1_5_REPULSION> equals 1, the filter will be applied. This has to be handled with care since the filter is very strict!

Default value: 0

Possible values: 0,1

Name (type): <FILTER_POLAR_HYDRO> (integer)

Description: The hydro filter checks the fragment placements and perhaps erases some solutions. All placements resulting in close contacts of non-interacting polar hydrogen atoms of the fragment and the receptor will be discarded. If <FILTER_POLAR_HYDRO> equals 1, the filter will be applied.

Default value: 1

Possible values: 0,1

Name (type): <FILTER_OUT_OF_SITE> (integer)

Description: The out of site filter checks the fragment placements and perhaps erases some solutions. All placements where heavy atoms of the fragment are placed outside of a sphere enclosing all active site atoms will be discarded. If <FILTER_OUT_OF_SITE> equals 1, the filter will be applied.

Default value: 0

Possible values: 0,1

Name (type): <FILTER_BURIED> (integer)

Description: The buried filter checks the fragment placements and perhaps erases some solutions. All placements resulting in an unsaturated hydrogen bond donor or acceptor within a hydrophobic pocket of the active site will be discarded. If <FILTER_BURIED> equals 1, the filter will be applied.

Default value: 1

Possible values: 0,1

Name (type): <FILTER_UNSATURATED> (integer)

Description: The buried filter checks the fragment placements and perhaps erases some solutions. All placements resulting in an orientation of the fragment in which a minimum saturation of all possible hydrogen bonding atoms is not fulfilled will be discarded. (See <MIN_INTERACTION_SATURATION>)

Default value: 1

Possible values: 0,1

Name (type): <MIN_INTERACTION_SATURATION> (floating point)

Description: The minimum fraction of all hydrogen bonding atoms which have to have at least one match each during the buildup process.

Default value: 0.3

Possible values: 0.0 – 1.0

A.2.3 Grafik-Einstellungen

Die Einstellungen der Visualisierung für FlexNovo werden in einer gesonderten Datei abgelegt, typischerweise `graphic.dat`. Darin finden sich die Parameter für die Darstellung von Fragmenten und Platzierungen sowie der daran annotierten Informationen.

```
#application FRAGSPACE
# -----
# Definitions for the default FRAGMENT graphics settings
# -----
@fragment-defaults

# -----
# FRAGMENT SELADM
# -----

switch MOL_OBJ_NUMBER           8
switch ORG_MODE                  1
switch TMP_FILES                 1
switch APPEND_MODE              0

# -----
# FRAGMENT SELGRA
# -----

switch MOL_DISP_MODE            1
switch DRAW_HYDROGEN            1
switch DRAW_INTERACT_GEOMS      0
switch DRAW_ALL_CONTACT_TYPES   1
list  ACTIVE_CONTACT_TYPES      0 1 2 3 4 5 6 7
switch DRAW_ALL_COMPONENTS      1
switch DRAW_SURFACE             0

# -----
# FRAGMENT SELCOL
# -----

# default color mode for drawing ligand
colormode FRAGMENT              ATOM
# default color for UNIQUE color mode
color    FRAGMENT               red
# defaults for coloring according to dock energy
# ENERGY color mode
# nof colors for ENERGY rainbow
switch  ENERGY_STEPS           10
```

```

# start / end energy values for ENERGY rainbow
scalar ENERGY_MIN -100.000
scalar ENERGY_MAX 0.000
# start / end colors for ENERGY rainbow
color ENERGY_COLOR_1 red
color ENERGY_COLOR_2 blue

# default color mode for drawing geometries
colormode GEOMETRY CONTACT
# default color for UNIQUE color mode
color GEOMETRY red

# default color mode for drawing surface
colormode SURFACE SURF_ATOM
# default color for UNIQUE color mode
color SURFACE light steel blue
# default color for reentrant / concave patches
color SURFPATCH_COLOR_0 yellow
# default color for saddle patches
color SURFPATCH_COLOR_1 yellow
# default color for convex patches
color SURFPATCH_COLOR_2 yellow
# nof colors / start / end colors for CEN_DIST rainbow
switch N_NOF_CENDIST_COLORS 10
color N_CENDIST_COLOR_0 blue
color N_CENDIST_COLOR_1 red

# -----
# FRAGMENT SELLAB
# -----

switch ATOM_NAMES 1
switch INFILE_NUMBERS 1
switch SYBYL_TYPE_STRINGS 0
switch FORMAL_CHARGES 0
switch PARTIAL_CHARGES 0

# -----
# Definitions for the default PLACEMENT graphics settings
# -----
@placement-defaults

# -----

```

A Die Benutzungsschnittstelle von FlexNovo

```
# PLACEMENT SELADM
# -----

switch MOL_OBJ_NUMBER          0
switch ORG_MODE                 2 15 254
switch TMP_FILES                1
switch APPEND_MODE              0

# -----
# PLACEMENT SELGRA
# -----

switch INCLUDE_FRAGMENT        1
switch INCLUDE_RECEPTOR      1
switch DRAW_OVERLAP            0
switch DRAW_ALL_CONTACT_TYPES  1
list  ACTIVE_CONTACT_TYPES     0 1 2 3 4 5 6 7

# -----
# PLACEMENT SELCOL
# -----

# default color mode for drawing docking interactions
colormode PLACEMENT            ENERGY
# default color for UNIQUE color mode
color    PLACEMENT              yellow
# defaults for coloring according to interaction ENERGY and OPT_ENERGY
# nof colors for OPT_/ENERGY rainbow
switch  ENERGY_STEPS          10
# start / end energy values for OPT_/ENERGY rainbow
scalar  ENERGY_MIN            -8.3  # max energy for interaction
scalar  ENERGY_MAX            0.0
# start / end colors for OPT_/ENERGY rainbow
color   ENERGY_COLOR_1        magenta
color   ENERGY_COLOR_2        cyan

# default color mode for drawing docking overlap
colormode OVERLAP              UNIQUE
# default color for UNIQUE color mode
color   OVERLAP                 green

color SURFACE                   undefined
switch N_NOF_CENDIST_COLORS     1
color GEOMETRY                  undefined
```

```
color N_CENDIST_COLOR_0          blue
color N_CENDIST_COLOR_1          red

# -----
# PLACEMENT SELLAB
# -----

switch INTERACTION_TYPES         0
switch ENERGIES                   1
switch OPTIMAL_ENERGIES          0

# -----
# PLACEMENT DRAW
# -----

switch NEW                        1
switch PLACEMENT_NUMBER          2 #no effect due to ask_dock_entry
switch COORDS                    -2 #no effect due to ask_dock_entry
switch DRAW_ATOMS                1
switch DRAW_BONDS                1
switch MUL_DRAW                  0
switch FIRST                     0
switch LAST                      0
scalar N_CENDIST_RANGE_0         Inf
scalar N_CENDIST_RANGE_1         0.000
scalar PLACEMENT_ENERGY          0.000
scalar SIZE                      0.000

#end_application
```

A.3 FlexNovo-Kommandos

Die Befehle des FlexNovo-Programms sind in Anlehnung an FlexX in einer hierarchischen und kommandozeilenbasierten Menüstruktur organisiert. Aus Konsistenzgründen und zur Vereinfachung der Bedienung des Programms für erfahrene Benutzer von FlexX ist sowohl die Benennung als auch die Syntax der Befehle soweit wie möglich identisch oder zumindest sehr ähnlich zu denen von FlexX.

Im Folgenden werden nur die für FlexNovo neu hinzugefügten Menüs und deren Befehle aufgeführt. Das Untermenü FRAGSPACE enthält alle Befehle zum Laden, Analysieren und Anzeigen von Fragmenten. Die Kommandos zum Berechnen, Laden, Speichern, Analysieren und Anzeigen von Platzierungsinformation für Fragmente befinden sich im Untermenü PLACING. Physikochemische Filterkriterien können im Untermenü FILTER geladen, angezeigt und angewandt werden. Die eigentlichen FlexNovo-Funktionen sind im Untermenü BUILDUP zu finden, das auch Befehle zum Laden, Speichern, Analysieren und Anzeigen von Lösungen enthält.

An den jeweiligen Stellen sind zum besseren Verständnis der Eingabeoptionen und aus Gründen der Vollständigkeit die entsprechenden Passagen aus dem FlexX-Benutzerhandbuch integriert. Eine vollständige Zusammenstellung der übrigen Menüs und Befehle kann dem FlexNovo-Benutzerhandbuch entnommen werden. Dort werden alle weiteren Kommandos im Zusammenhang mit der zugrunde liegenden Datenbasis, der Behandlung des Rezeptors sowie des Pharmakophors beschrieben.

A.3.1 Arbeiten mit Fragmenträumen (FRAGSPACE Untermenü)

Die meisten der folgenden Befehle können entweder auf alle Fragmente des Fragmentraums oder aber auf eine Untermenge (Selektion) angewandt werden. Letzere kann mit dem Befehl SELECT erzeugt werden oder durch Anwendung einer Filterspezifikation für physikochemische Eigenschaften (Kommando APPLY im Untermenü FILTER) zustande kommen.

A.3.1.1 Reading (READ)

Syntax: READ <filename>

Description: Reads a fragment space, stored in an `fsf` file, from disk. A previously loaded space will automatically be deleted.

A.3.1.2 Fragment space information (INFO)

Syntax: INFO

Description: Displays some statistical information about the space currently loaded on the screen:

- Number of link types, number of links for each type and total
- Link compatibility matrix (which link can be connected to which)

- terminal group information for each link type (SMILES string of terminal group or MOL2 filename, corresponding atom and bond type and specified torsion angle)
- Total number of fragments, names of all fragment and link files
- Link histogram (number of times a specific link type occurs)
- Estimation of the number of molecules contained in the space is calculated, based on the maximum diameter in the molecule (longest chain of fragments) and the maximum number of fragments.

Requirements: A fragment space must be loaded.

A.3.1.3 Checking geomtry and rule compatibility (CHECK)

Syntax: CHECK

Description: Uses a rather simplistic rule-based approach to identify fragments which might undergo local configurational changes upon connection or substitution of the link atom with its terminal group. This is done at the moment for Nitrogen atoms having a link atom attached only. Note, that it is crucial to have correct SYBYL atom types assigned.

Important notes: The outcome of the function and the geometry correction heavily depends on the flag <USE_GEOM_CORRECTION>.

A.3.1.4 Deleting (DELETE)

Syntax: DELETE

Description: Removes a fragment space from FlexNovo's main memory.

A.3.1.5 Creating a fragment selection (SELECT)

Syntax: SELECT <frag-id range> <file-id range> <mol-id range> <nof-links range> <link-type list> <fragment name substring>

Description: Generates a selection of fragments from the currently loaded fragment space following the specification of the **SELECT** parameters. All parameters are optional in the sense that they can be set to undefined.

Requirements: A fragment space must be loaded.

A.3.1.6 Fragment information (FLIST/FINFO)

Syntax: FLIST <format> [<datatype>]

Description: Prints a list of all fragments which are currently loaded or selected. The output table contains the fragment number, fragment name, file ID, molecule ID, number of links and link names.

Requirements: A fragment space must be loaded.

Syntax: FINFO <format> [<datatype> <mode>]

Description: Displays the main characteristics of a fragment, such as information about the size, the topology, the interactions and the physicochemical properties.

A.3.1.7 Initializing fragments (INIT)

Syntax: INIT <datatype>

Description: INIT performs the fragment initialization, i.e. for atom types, formal charges and so on. The initialization is the same as is carried out when loading a ligand.

Important notes: It is not necessary to use this command for a build-up calculation or for filtering. The values needed by either one of the other functions will be calculated on demand. However, **we recommend using this command before starting to work with a particular fragment space in order to check if there are potential problems/inconsistencies with regard to, for example, the fragment geometry or atom-/bond-type definitions!**

A.3.1.8 Writing (WRITE)

Syntax: WRITE <datatype> <filename> <set name> <append> <coord type> <multi file> <coord type> <placement selection>

Description: Stores fragments or a selection of fragments in a file or multiple files respectively. The default format is MOL2 [Tripos, 2008]. Alternative formats can be selected by the appropriate filename extension. Possible formats are MOL [Tripos, 2008] (extension `.mol`) or PDB [Berman *et al.*, 2000] (extension `.pdb`). If <set name> is set to 'y', each fragment will be written to a separate file, the filename is determined by the fragment sybyl name. If <append> is set to 'y', the fragment is appended to any previously existing file. Otherwise any previously existing file will be overwritten. If <multi file> is set to 'y', all entries are written in one file. If <placement selection> is set to '0', the fix coordinates of the input file will be used, otherwise single or multiple placements are written.

A.3.1.9 Selecting admin settings for drawing fragments (SELADM)

Syntax: SELADM <graphics object number> [<start fifo object>] [<end fifo object>] <temp file> <append>

Description: With SELADM you can specify the graphics object numbers used for drawing fragments and you can determine whether the graphics files are internal temporary files used only by FlexNovo or saved for further use. For yes/no questions you can enter either 'y', 'yes' or '1' for yes, and similarly 'n', 'no' or '0' for no.

<graphics object number> Enter integer:

- (1–255)** The graphics created with the **DRAW** command will be displayed in graphics object `<graphics object number>`.
- 0** *fifo* mode – the graphics generated by subsequent **DRAW** commands will be sent to a range of graphics objects in a first-drawn-first-overwrite manner. You will be asked to enter two more parameters:
- `<start fifo object>` The start graphics object for the *fifo* range.
 - `<end fifo object>` The end graphics object for the *fifo* range.
- `<temp file>` Yes/no answer:
- yes** The graphics are written in temporary files and removed after quitting FlexNovo.
 - no** The graphics are written to permanent files chosen by the user. You will be asked for a filename at the end of each **DRAW** command (see **DRAW** below for example).
- `<append>` Yes/no answer:
- yes** Previous graphics files are not overwritten. Instead the current graphics are appended to the previous one in the graphics file.
 - no** The previous graphics file will be overwritten and all previous graphics made with the **DRAW** command in this menu will be lost.

A.3.1.10 Selecting graphics settings for drawing fragments (**SELGRA**)

Syntax: **SELGRA** `<mol display mode>` `<hydro>` `<interact geoms>` `<all contact types>` [`<contact type selection>`] `<all components>` [`<component selection>`] `<surf>`

Description: With **SELGRA** you can set specific default values for drawing fragments. For yes/no questions you can enter either 'y', 'yes' or '1' for yes, and similarly 'n', 'no' or '0' for no.

`<mol display mode>` Specifies the drawing display mode for molecule selection:

- 1** Lines
- 2** Sticks
- 3** Balls & sticks
- 4** Space-filled spheres

(Aside: the default appearance of fragments drawn with **MDRAW** in the **BUILDUP** menu is sticks, independent of the setting here – this helps with visualization of the placements against the protein active site. Use the **MDRAW** command here in the **FRAGSPACE** menu (see below) to override this default setting.)

`<hydro>` Specifies whether and how hydrogens should be drawn on the ligand selection:

- 0** Do not draw hydrogens.
- 1** Draw all hydrogens.
- 2** Draw only hydrogens bonded to hetero atoms (non-carbon atoms).

<interact geoms> Yes/no answer:

- yes** Interaction geometries (interaction surfaces around potential interacting groups) are drawn.
- no** No interaction geometries are drawn.

<all contact types> Yes/no answer:

- yes** Interaction geometries for *all* contact types (interaction types) are drawn if <interact geoms> is set to 'yes'.
- no** Interaction geometries are drawn for a selection of contact types (interaction types). You will be asked to select types from a given list:

<contact type selection> Choose a list of types represented by integers. Enter the list as separate integers or as integer ranges (format $a - b$) separated by ',' or blanks. Note that you need to enclose the expression in quotation marks if it contains blanks e.g. 1, 2, 4, 7 - 9.

<all components> Yes/no answer:

- yes** All components are drawn, i.e. the complete fragment is drawn.
component: On loading a fragment, FlexNovo identifies all rotational bonds (according to its internal definition of rotatable bonds) and splits the fragment into components at these bonds. Later, the components will be used to decide on a base fragment and will form the building blocks for the incremental reconstruction of the fragment in the active site.
- no** Only components are drawn from a selected list. You will be asked to select components from a given list:

<component selection> Choose a list of components represented by integers. Enter the list as separate integers or as integer ranges (format $a - b$) separated by ',' or blanks. Note that you need to enclose the expression in quotation marks if it contains blanks e.g. 1, 2, 4, 7 - 9.

<surf> Determine to draw the surface:

- 0** draw no surface
- 1** draw the molecular surface: If FlexV is used to visualise, the Connolly surface is drawn. Otherwise only concave patches as triangles are drawn.

Note: Hydrogens are not considered when drawing the surface — if hydrogens are to be drawn (<hydro>) they will be ignored when drawing the surface.

Requirements: Depending on the graphical interface, the surface modes are limited. In VRML mode, both lines mode and triangles mode are available. With FlexV, all surface modes can be shown.

Important notes: The Connolly surface is rendered by its analytical calculated patches. This enables selection of the level of curvature approximation but makes the rendering much more complicated. Therefore a few percent of the

patches are rendered incorrectly (we will try to reduce this rate). In addition, there is currently only pairwise cusp trimming.

A.3.1.11 Selecting colors for drawing fragments (SELCOL)

Syntax: SELCOL <fragment color mode> <interact geoms color mode> <surface color mode>

Description: With SELCOL you can set the color modes for the fragment, interaction geometries and molecular surface. For each of these, a selection of color modes are available:

<ref. coords> Yes/no answer:

yes The following modifications concern the settings for drawing the ligand with the reference coordinates.

no The following modifications concern the settings for drawing the ligand with the input coordinates.

<fragment color mode> Choose the color mode for drawing the fragment – color mode selection:

INVISIBLE

ATOM

UNIQUE

ENERGY

<interact geoms color mode> Choose the color mode for drawing the interaction geometries if they are to be drawn. The interaction geometries consist of patches or surfaces that indicate the positions of interacting groups in the molecule. Color mode selection:

INVISIBLE

UNIQUE

CONTACT

<surface color mode> Choose the color mode for coloring the molecular surface if it is to be drawn. Color mode selection:

INVISIBLE

UNIQUE

SURF_ATOM

CEN_DIST

SURFPATCH

The possible color modes are explained below – for some color modes you are also asked to enter some defining colors. Enter your chosen color as either an angle from the color circle (0 – 360 degrees: 0 is invisible, 1–360 runs from dark blue, through red, yellow, green to blue), a color name (as defined in the GRAPHIC static data file), or an RGB(A) value; 3 (4) floating-point numbers separated by blanks or slashes:

Example

```
selcol .... "dark green" ....  
selcol .... green ....  
selcol .... "0.0 0.8 0.1" ....  
selcol .... 0.0/0.8/0.1 ....  
selcol .... 220 ...
```

INVISIBLE The item drawn will be invisible.

ATOM The ligand will be colored according to the element types of the atoms.

The atoms are drawn in the color defined for its element type in the static data file **GRAPHIC**, while the bonds are drawn half and half in the neighboring atom colors.

UNIQUE The object will be drawn in one user-defined color. You will be asked to choose the unique color:

<**color**> Enter your chosen color.

ENERGY Draw the fragment conformation in a color representative of its build-up solution score (energy). A color rainbow will be defined between two given colors across the range of two given build-up scores. You are required to enter:

<**no. of intervals**> Enter the number of intervals (integer) that the energy range will be split into.

<**min energy**> Enter the minimum energy value (floating-point number) for the start of the energy range. The default is the score of the best build-up solution.

<**max energy**> Enter the maximum energy value (floating-point number) for the end of the energy range. The default is the score of the worst scoring build-up solution.

<**first color**> Enter the first color of the color rainbow.

<**second color**> Enter the second (end) color of the color rainbow.

CONTACT The object will be drawn in a color representing its interaction (contact) type. The colors for each type are defined in the **GRAPHIC** static data file.

SURF_ATOM Convex patches in the surface are colored by atom type (see color mode **ATOM**). Any reentrant patches (i.e. saddle and concave patches) are drawn in a user-defined color. You are required to enter the reentrant patch color in this mode:

<**reentrant color**> Enter your chosen color.

CEN_DIST The surface is drawn in a rainbow of colors representing how far the surface lies from the (geometric) center of the molecule. For this mode you are required to enter the start and end colors of the rainbow plus the number of intervals to be colored across the rainbow range:

<**no of intervals**> The number of intervals into which the range will be split.

<**first color**> Start color for the rainbow.

<**second color**> End color for the rainbow.

SURFPATCH The surface is colored according to the surface patch type. You are required to enter colors for the various patch types in this mode:

<**concave color**> Enter your chosen color for concave patches.

<**saddle color**> Enter your chosen color for saddle patches.

<**convex color**> Enter your chosen color for convex patches.

A.3.1.12 Selecting labels for drawing fragments (**SELLAB**)

Syntax: SELLAB <atom name> <infile number> <SYBYL type> <component number> <formal charge> <partial charge>

Description: When the fragment is drawn, FlexNovo stores information in labels for display in the graphic interface. You can choose what should appear in the label using the **SELLAB** command. For yes/no questions you can enter either 'y', 'yes' or '1' for yes, and similarly 'n', 'no' or '0' for no.

<**atom name**> Yes/no answer:

yes Include the atom name as taken from the input file in the label for atoms.

no Atom names will not be included in the label for atoms.

<**infile number**> Yes/no answer:

yes Include the number of the atom as taken from the input file in the label for atoms.

no Infile numbers will not be included in the label for atoms.

<**sybyl type**> Yes/no answer:

yes Include the SYBYL atom types in the label for atoms.

no SYBYL atom types will not be included in the label for atoms.

<**fragment number**> Yes/no answer:

yes Include the fragment number in the label for atoms.

no Fragment numbers will not be included in the label for atoms.

Note: For an explanation of *fragments* see, for example, the LIGAND/SELCOL command.

<**formal charge**> Yes/no answer:

yes Include the formal charges on the atoms in the label for atoms.

no Formal charges will not be included in the label for atoms.

<**partial charge**> Yes/no answer:

yes Include the partial charges on the atoms in the label for atoms.

no Partial charges will not be included in the label for atoms.

Important notes: VRML 1.0 is (to our knowledge) not able to handle 2D text in 3D scenes which makes labeling a little bit strange under visualization with VRML browsers.

A.3.1.13 Listing the graphic items (GRAINF)

Syntax: GRAINF

Description: Outputs a list of all current graphic settings for the fragment.

A.3.1.14 Drawing multiple fragments (MDRAW)

Syntax: MDRAW <mode>

Description: Generates multiple drawings of fragments using coordinate sets taken from the input data.

A.3.2 Arbeiten mit Filter-Spezifikationen (FILTER Untermenü)

A.3.2.1 Reading (READ)

Syntax: READ

Description: Reads a filter definition from a file.

Requirements: A fragment space must be loaded.

A.3.2.2 Filter information (INFO)

Syntax: INFO

Description: Displays the filter rule definitions on the screen.

A.3.2.3 Deleting (DELETE)

Syntax: DELETE

Description: Erases the filter definition from FlexNovo's memory.

A.3.2.4 Applying filter criteria (APPLY)

Syntax: APPLY <datatype>

Description: Applies the filter ruleset on all fragments of the fragment or a selection of fragments. All fragments which violate the filter rules to a certain extent with respect to the rule violation tolerance will not be put in a new selection or excluded from an existing one respectively.

Important notes: This command creates a fragment selection, if not applied to an existing one. Any previous selection made will therefore be overwritten.

Requirements: A fragment space and a filter definition must be loaded.

A.3.3 Platzierung von Fragmenten (PLACING Untermenü)

A.3.3.1 Placing fragments (PLACE)

Syntax: PLACE <datatype> <placement algorithm> <save info file to disk> [<filename> <table format> <output style>] <save placement file to disk> [<code transformations> <directory>] <keep in memory>

Description: Generates placements using the FlexX docking algorithm. Note, that only fully automatic FlexX docking algorithms are accessible with this command. If <save info file to disk> is set, the main characteristics of the placement results can be directly written to a file. The information printed is the same as with the command INFO (see below). If <save placement file to disk> is set, the placements information is written in a FlexNovo-specific file format (.pdf format) on disk. The default directory for this command is the path specified in the entry PREDICT (config.dat). The pdf format is based on ASCII and can therefore be read and edited with standard tools. Because small changes in transformations can result in different solutions, transformation information should be coded by setting <code transformations> to 'y'.

Important notes: Coding works only on machines with specific floating-point representations. Thus, it may be the case that coding cannot be used on your hardware platform. Be careful when reading and writing coded pdf files on different machines.

Because this format stores all the internal information, format changes are necessary from time to time. Therefore pdf files may not be compatible between different FlexNovo versions.

By setting <PDF_COMPRESSION> (config.dat) to 1, placement files can be written in compressed (gzip) format to save disk space.

<save info file to disk> and <save placement file to disk> should in particular be used in cases, where the fragment space is quite large and it is not possible to keep solutions for every fragment in memory at the same time.

Requirements: A fragment space and a receptor must be loaded.

A.3.3.2 Printing placement information (PLINFO/FLIST)

Syntax: PLINFO <datatype> <table format> <output style>

Description: Displays the main characteristics of the placements, such as number of solutions, highest ranking score, etc. on the screen. If <table format> is set to 'y', the result is output in a set of tables, otherwise one of the tables is output on one line. The single-line option is very useful to summarize a placement run over large data sets. All single lines (except for output_style 7, see below) start with receptor name, fragment name, number of solutions and the first three ones send with computation time for base placement and complex construction. In between the contents is

standard Score, RMSD of solution at rank 1; score, RMSD, and rank of highest

ranking solution with RMSD less than 1.0 Å; score, RMSD, and rank of solution with minimal RMSD.

rms Score and rank of highest ranking solution with minimal RMSD and then with RMSD below the thresholds 1.0 Å, 1.5 Å, 2.0 Å, 2.5 Å.

rank Score and RMSD of solution with lowest RMSD within the first 1, 10, 20, 50, 100 solutions.

acc. rms Accumulated minimal RMSD of the first 1, 10, 20, 50, 100 solutions. If $r(i)$ is the RMSD of solution at rank i , the accumulated minimal RMSD of the first k solutions is $k^{-1} \sum_{i=1}^k \min_{1 \leq j \leq i} r(j)$. The basic idea of the accumulated RMSD is to define a quality measure which is independent of discrete RMSD or rank thresholds.

rms history Number of fragments; RMSD and rank of the best solution

frag score This mode is not available in the current version.

screen Individual score contributions, individual interaction types, location relative to probe positions and the changes in the SAS upon complex formation of the receptor and the fragment.

score Fragment name and score only (see INIT below).

Important notes: Use <output style> 7 to generate the format which can be directly used with the INIT command below.

Requirements: Placements must be computed or loaded.

Syntax: FLIST <format> [<datatype>]

Description: Prints a list of all fragments which are currently loaded or selected. The output table contains the fragment number, fragment name, file ID, molecule ID, initialized score, if available, and the score of the best placement, if available.

Requirements: A fragment space must be loaded.

A.3.3.3 Deleting (DELETE)

Syntax: DELETE

Description: Erases all placement information from FlexNovo's memory.

A.3.3.4 Generating a fragment selection (SELECT)

Syntax: SELECT <mode> [<number>]

Description: Generates a selection of fragments from the currently loaded fragment space following the specification of the SELECT parameters. There are three possible modes: Selecting all fragments which have no placements, all fragments which do have placements or selecting the best scoring <number> of fragments for which a score value has been computed or loaded (see command INIT below).

Requirements: Placements must be computed or loaded.

A.3.3.5 Loading placement scores from a file (INIT)

Syntax: INIT <mode> <directory>

Description: INIT performs an initial score evaluation of all fragments or a selection of fragments by reading placement scores from a standard text file. This file has to contain the fragment name and a score value, it can be automatically generated by using INFO with <table_format> set to n and <output style> set to 7.

Important notes: It is highly recommended to make use of this functionality since a proper score evaluation dramatically reduces the computation time.

Requirements: A fragment space must be loaded.

A.3.3.6 Writing placements in pdf format (WRITE)

Syntax: WRITE <datatype> <code transformations> <directory>

Description: Stores placement information in files. The filename is equivalent to the name of the fragment the placement information belongs to. According to the value of <PDF_COMPRESSION> (config.dat) the placement files are written in normal or compressed (gzip) file format.

Important notes: The value of <PDF_COMPRESSION> has to be consistent with the placement file format. Compressed files cannot be loaded when <PDF_COMPRESSION> is set to 0.

Requirements: Placements must be computed or loaded.

A.3.3.7 Reading placements in pdf format (LOAD)

Syntax: LOAD <datatype> <directory>

Description: Reads placement information from files. The filename is equivalent to the name of the fragment the placement information belongs to.

Important notes: The value of <PDF_COMPRESSION> has to be consistent with the placement file format. Compressed files cannot be loaded when <PDF_COMPRESSION> is set to 0.

Requirements: A fragment space and a receptor must be loaded.

A.3.4 Strukturbasierte Suche (BUILDUP Untermenü)**A.3.4.1 Starting the search algorithm (GREEDY)**

Syntax: GREEDY <mode>

Description: GREEDY starts the build-up process with the parameters specified in the config.dat file. The buildup calculation uses as start fragments the best <BUILDUP_QUEUE_SIZE> scoring fragments for which there is placement information loaded or computed. These are determined either by checking all fragments of the fragment space, a selection of fragments or previously loaded

partial solutions. In any case, the whole fragment space is used further during the buildup process.

There are up to <BUILDUP_EXTENSION_CYCLES> extension cycles performed. For each extension cycle there are up to <BUILDUP_QUEUE_SIZE> fragments (meaning partially built-up molecules) kept in the queue. To each compound all fragments are added and possible placements evaluated according to the link connection rules specified in the **fsf** file. If filter criteria are specified, these are evaluated first and the resulting compound is discarded if the filter rules are violated to a certain extent. If the score evaluation is used and a minimum number of compounds has already been constructed, every new compound is evaluated according to its best score achievable. This is done with respect to the scores of the already built-up fragments/molecules. If the score to be expected is too low, the fragment or molecule is discarded without doing any further calculation.

Important notes: In case the buildup is done in a stepwise manner, meaning doing a separate calculation for each extension cycle, the parameters have to be adjusted. The parameter <BUILDUP_RESOLVE_LINKS> has to be set to 0 in order to generate results which have unsubstituted link atoms. Furthermore, the parameter <NOF_BUILDUP_PLACEMENTS> should be increased in order to generate a reasonable amount of placement information for each result.

In case the buildup is done in a parallel manner, meaning doing separate calculations for each fragment in the queue, the parameters have to be adjusted as well. In this context, the parameters <BUILDUP_QUEUE_SIZE> and <NOF_BUILDUP_RESULTS> have to be set to much smaller values than in the sequential case. Otherwise the total computation time will be much higher. In both cases it may also be necessary to adjust the diversity parameters.

Requirements: A fragment space, a receptor and start placements must be available.

A.3.4.2 Listing main statistical build-up information (INFO)

Syntax: INFO

Description: Displays some statistical and timing information of the build-up calculation on the screen.

Requirements: Build-up results must be computed.

A.3.4.3 Deleting build-up results (DELETE)

Syntax: DELETE

Description: Erases all build-up information from FlexNovo's memory.

A.3.4.4 Generating a selection of build-up results (SELECT)

Syntax: SELECT <solution-id range>

Description: Generates a selection of solution list molecules from the currently loaded or computed build-up results list.

Requirements: Build-up results must be computed or be loaded.

A.3.4.5 Listing the build-up solutions (LISTSOL)

Syntax: LISTSOL <table length>

Description: LISTSOL displays a table of the <table length> best scoring solutions on the screen. The table consists of the following elements:

Rank The number of the solution.

Molecule Name The name of the molecule (reflecting the build-up order of the fragments it consists of).

Score The total score of the best placement.

#poses The total number of placements.

Requirements: Build-up results must be computed or be loaded.

A.3.4.6 Outputting information about the build-up results (SOLINFO/PLINFO)

Syntax: SOLINFO <format> [<datatype> <mode>]

Description: Displays the main characteristics of a build-up solution, such as information about the size, the topology, the interactions and the physical properties.

Requirements: Build-up results must be computed or be loaded.

Syntax: PLINFO <datatype> <table format> <output style>

Description: Displays the main characteristics of the build-up solution placements, such as number of solutions, highest ranking score, etc. on the screen. If <table format> is set to 'y', the result is output in a set of tables, otherwise one of the tables is output on one line. The single-line option is very useful to summarize a placement run over large data sets. All single lines (except for output_style 7, see below) start with receptor name, fragment name, number of solutions and the first three ones send with computation time for base placement and complex construction. In between the contents is

standard Score, RMSD of solution at rank 1; score, RMSD, and rank of highest ranking solution with RMSD less than 1.0 Å; score, RMSD, and rank of solution with minimal RMSD.

rms Score and rank of highest ranking solution with minimal RMSD and then with RMSD below the thresholds 1.0 Å, 1.5 Å, 2.0 Å, 2.5 Å.

rank Score and RMSD of solution with lowest RMSD within the first 1, 10, 20, 50, 100 solutions.

acc. rms Accumulated minimal RMSD of the first 1, 10, 20, 50, 100 solutions. If $r(i)$ is the RMSD of solution at rank i , the accumulated minimal RMSD

of the first k solutions is $k^{-1} \sum_{i=1}^k \min_{1 \leq j \leq i} r(j)$. The basic idea of the accumulated RMSD is to define a quality measure which is independent of discrete RMSD or rank thresholds.

rms history Number of fragments; RMSD and rank of the best solution

frag score This mode is not available in the current version.

screen Individual score contributions, individual interaction types, location relative to probe positions and the changes in the SAS upon complex formation of the receptor and the fragment.

score Fragment name and score only (see INIT below).

Requirements: Build-up solutions must be computed or loaded.

A.3.4.7 Writing build-up solutions and placements (WRITE)

Syntax: WRITE <datatype> <filename> <individual filenames> <append> <multi file> <placement selection> <formal charges> <code transformations> <directory>

Description: WRITE stores solutions, meaning compounds and placement data in a number of files. Compounds can be stored in separate or in a multi molecule file. Placements can only be stored in separate files, one for each compound. Note, that at the moment the filename convention for the placement files is <sol no>-<molecule name>.

Important notes: Normally it probably is sufficient to store 10 placements for each compound in the solution list. In case the buildup calculation is done in a stepwise manner, meaning doing a separate calculation for each extension cycle, the number of placements kept should be set to a higher value, 50 to 100 for example.

Requirements: Build-up results must be computed or be loaded.

A.3.4.8 Loading build-up solutions and placements (LOAD)

Syntax: LOAD <filename> <nof solutions> <directory>

Description: LOAD reads solutions, meaning compound and placement data from a file, using the following filename convention for the placement files <sol no>-<molecule name>.

Requirements: A fragment space and a receptor must be loaded.

A.3.4.9 Mergin build-up solutions from different runs (MERGE)

Syntax: MERGE <filename> <nof solutions> <directory>

Description: MERGE allows for loading multiple solution sets. These can for example originate from different runs for the same receptor with the same fragment space.

Important notes: If the different solution sets contain identical molecules, only the molecule with the better score for the best placement will be kept, the other one will be discarded.

Requirements: A fragment space and a receptor must be loaded.

A.3.5 Zeichnen und Auflisten wichtiger Platzierungsinformationen

Die folgenden Befehle sind sowohl in den Untermenüs PLACING als auch BUILDUP enthalten. Sie beziehen sich somit auf Platzierungsinformationen für Fragmente bzw. Lösungen des Aufbaualgorithmus.

A.3.5.1 Listing placements (LISTPLM)

Syntax: LISTPLM <datatype> <selection> <table length>

Description: Displays a table of <table length> placements for each fragment or solution of the build-up calculation selected in <selection> on the screen. (The amount output may depend on your setting of USER_MODE in config.dat.) The table columns depend on the scoring function settings in geometry.dat consisting of the following columns. The column identifiers are shown in parentheses:

No. (SOL_NO) The number of the solution.

Total Score (E_TOTAL) Total score of the build-up solution.

Match Score (E_MATCH) Contribution of the matched interacting groups.

Lipo Score (E_LIPO) Contribution of the lipophilic contact area.

Ambig Score (E_ambiguous) Contribution of the lipophilic-hydrophilic (ambiguous) contact area.

Clash Score (E_CLASH) Contribution of the clash penalty.

Rot Score (E_ROT) fragment conformational entropy score.

RMS Value (RMS) RMSD of coordinates from reference coordinates. If there are no reference coordinates, this column contains the RMSD from the highest ranking solution.

Similarity Index (SIM_IDX) Measure of similarity between solution coordinates and reference coordinates. If there are no reference coordinates, all entries of this column are -1.0. The similarity index score is similar to the RMSD value given in the previous column. The RMSD value however, is strictly restricted to calculate the RMSD between the 2 coordinates assigned to one atom: the build-up solution coordinates and the reference coordinates. The similarity index rather corresponds to a "fuzzy" similarity measure: It is based on an RMSD between the build-up coordinates of an atom and the reference coordinates of the nearest atom of the same SYBYL atom type. For example a symmetric molecule docked back to front will have a bad (high) RMSD but can still achieve a good (low) similarity index.

- #Match** (NOF_MATCH) Number of matches.
- Avg. Volume** (AVG_VOL) Average volume of protein/fragment overlap.
- Max Volume** (MAX_VOL) Maximum volume of protein/fragment overlap.
- Fragmentation No.** (FRAG_NO) Number of the fragmentation used for this prediction.
- Conf. String** (CONF_STR) String displaying the conformation of the fragment (internal notation) (**in DEBUG_MODE only**).
- Sol. String** (SOL_NR_STR) String displaying the rank of the solution after each build-up step (**in DEBUG_MODE only**).
- Frag. String** (FRAG_NR_STR) String displaying the fragment conformation numbers of the solution of the build-up step (**in DEBUG_MODE only**).

A.3.5.2 Listing matches of all placements (LISTMAT)

Syntax: LISTMAT <datatype> <selection> <table length>

Description: Displays a table of <table length> matches for each fragment or solution of the build-up calculation selected in <selection> on the screen. Matches with zero score are automatically omitted from the table. The table has the following columns:

- No.** (SOL_NO) The number of the corresponding solution.
- Lig. Atom** (LIA_ATOM) Ligand atom of the match.
- Lig. ANo.** (LIA_ATOM_NO) Number of interacting atom of fragment.
- Ligand IA-Type** (LIA_TYPE) Ligand interaction type.
- Rec. Atom** (RIA_ATOM) Receptor atom of the match.
- Rec. AA** (RIA_AA) Receptor amino acid.
- Rec. Chain id** (RIA_AA_CHAIN) Receptor chain identifier.
- Rec. AANo** (RIA_AANO) Number of the receptor amino acid.
- Receptor IA-Type** (RIA_TYPE) Receptor interaction type.
- Opt. Energy** (E_OPT) Optimal score (without geometry penalties) of the match.
- Chg.** (CHG) Product of formal charges of the interacting atoms.
- Chg. fact.** (CHG_FAC) Charge factor for the interaction.
- LDev.** (LDEV) Length deviation.
- LDev. fact.** (LDEV_FAC) Length deviation factor.
- ADevL** (ADEVL) Angle deviation on fragment site.
- ADevL fact.** (ADEVL_FAC) Angle deviation factor on fragment site.
- ADevR** (ADEVR) Angle deviation on receptor site.
- ADevR fact.** (ADEVR_FAC) Angle deviation factor on receptor site.
- Res. Engy.** (E_RES) Resulting match score (optimal score multiplied by the charge factor and rescaled by the deviation factors).
- Multip. fact.** (MULTIP_FAC) Interaction multiplicity factor.

A.3.5.3 Listing all placements and matches (LISTALL)

Syntax: LISTALL <datatype> <selection> <table length>

Description: Displays a table of <table length> placements and the corresponding matches for build-up solutions or fragments selected in <selection> on the screen. For a description of the table columns, see the two sections above.

A.3.5.4 Listing all placements and matches for a single fragment (LISTONE)

Syntax: LISTONE <name> <table length>

Description: Displays a table of <table length> placements and the corresponding matches for fragment or solution <name> on the screen. For a description of the table columns, see the two sections above.

A.3.5.5 Performing specific queries on solutions and matches (QUERY)

In many real cases, the number of solutions and matches is very large. It is possible to select specific information from the solution and matches tables. There are three ways of selecting or rearranging the table information:

SELECT specific *columns* FROM the table(s).

Select specific *rows* of the table(s) WHERE a certain condition applies.

Output the information selected in this way SORTed BY some criteria.

An SQL-like language is provided for the user to tell FlexNovo what information to display.

Syntax: QUERY <field list> <table list> [<condition>] [<order list>] <table length>

Description: <field list> is a list of field names, separated by colons. A field name is an identifier for a column of one of the tables. A list of valid strings for field names is output when you enter QUERY without parameters (see section A.3.5.1 and A.3.5.2 for a list of valid strings). In the resulting output, only the table columns represented by <field list> are listed. An asterisk * for <field list> is valid and represents the complete list of field names. <table list> is a list of the names of the tables you want to see, separated by colons. Valid table names are **solutions** and **matches**.

An asterisk * for <table list> is also valid and stands for **solutions**, **matches** (or, equivalently, **matches**, **solutions**). <table list> must contain all tables, the columns of which have been selected by <field list>.

<condition> is optional and selects *rows* of the tables (whereas <field list> selects *columns*). An atomic condition is a field name followed by an arithmetic operator (=, >=, <=, >, < or !=) or the contains-string operator ([]), followed by an appropriate constant. String constants must be enclosed in single quotes. Note that amino acid numbers and atom names must follow the PDB nomenclature, i.e. leading and trailing blanks are essential for the PDB encoding scheme

and must be part of the string constant (amino acid numbers are strings, not integers). Atomic conditions can be joined by the binary Boolean operators **and** and **or**. Conditions can also be nested with brackets '(, ')’.

The underlying semantics for conditions containing an **and** operator for different combinations of tables are different. If you have selected the solutions or the matches table separately, the condition is checked *for each row* separately. For example, this can be used to find out all hydrogen bonds to a specific protein atom with a score less than x kJ/mol.

If both tables are selected and a condition for match table entries is defined containing an **and** command, the condition is checked for the whole set of matches of one solution in common. For example, this can be used to find out all solutions forming interactions to amino acid x *and* amino acid y (see examples below). Note that the original meaning of **and** is lost in this context. Thus the following query is currently impossible: `Show all solutions forming an interaction to amino acid x with score higher than x_1 and to amino acid y with score higher than y_1 .`

`<order list>` is also optional and describes the order in which the selected rows are to be displayed on the screen. An `<order list>` is a list of order specifications, separated by colons. An order specification consists of one of the strings **ascending** or **descending**, followed by a field name. The field name must be an element of `<field list>`. The string **ascending** or **descending** is optional. If it is missing, **ascending** is assumed.

Example

```

QUERY  "sol_no, e_total, e_match, e_lipo, e_rot"  solutions
QUERY  *  solutions "sol_no > 20 and (e_total<-10.0 or nof_match > 3)"  ""
QUERY  *  *  ""  "descending e_total, ascending nof_match, descending e_res"
QUERY  *  *  "ria_aano [] ' 53' and ria_aano [] ' 79'"  ""

```

The first query shows five columns (solution number and four energy values) of the complete solutions table.

The second query shows all columns of the solutions table, but only those solutions (rows) whose number is greater than 20 and whose total energy is either less than -10.0 or whose number of matches is greater than 3.

The third query shows the complete solutions and matches tables, but reordered: the solutions are sorted by decreasing total energies, those with equal total energies are sorted by ascending number of matches. The matches of one solution will be sorted by decreasing resulting energies.

The fourth query shows all solutions forming interactions to amino acids 53 and 79. The pair of subsequent double quotes "" in example 2 (3) represents a missing optional parameter `<order list>` (`<condition>`).

A.3.5.6 Performing a specific query a second time (QHIST)

Syntax: QHIST <query no.>

Description: With QHIST you can perform a previous query again. After typing the command, you will receive a list of the last ten query commands. You can choose one of them by its number <query no.>.

Requirements: A query must have been previously performed using the command QUERY.

A.3.5.7 Selecting the admin settings for drawing placements (SELADM)

Syntax: SELADM <graphics object number> [<start fifo object>] [<end fifo object>] <temp file> <append>

Description: With SELADM you can specify the graphics object numbers used for drawing placements and you can determine whether the graphics files are internal temporary files used only by FlexNovo or saved for further use. For yes/no questions you can enter either 'y', 'yes' or '1' for yes, and similarly 'n', 'no' or '0' for no.

<temp file> Yes/no answer:

yes The graphics are written in temporary files and removed after quitting FlexNovo.

no The graphics are written to permanent files chosen by the user. You will be asked for a filename at the end of each DRAW command (see DRAW below for example).

<append> Yes/no answer:

yes Previous graphics files are not overwritten. Instead the current graphics are appended to the previous one in the graphics file.

no The previous graphics file will be overwritten and all previous graphics made with the DRAW command in this menu will be lost.

A.3.5.8 Selecting graphics settings for drawing placements (SELGRA)

Syntax: SELGRA <include frag> <include rec> <overlap> <all contact types> [<contact type selection>]

Description: With SELGRA, you can set specific values for drawings of placements (build-up solutions). The drawing of a build-up solution contains optionally the fragment/solution in the predicted conformation and receptor, plus a set of dashed lines connecting the interacting groups where interactions are formed. The overlap can also be drawn. The following choices are available. For yes/no questions you can enter either 'y', 'yes' or '1' for yes, and similarly 'n', 'no' or '0' for no.

<include frag> Yes/no answer:

yes The fragment or solution is included in the drawing. The drawing settings for the ligand are taken from the settings in the LIGAND menu,

except that the molecule display mode is set to *sticks* for straightforward visualization with the receptor. The fragment will be drawn in the graphics object for placing (see SELADM)

no The fragment is not included in the drawing

<include rec> Yes/no answer:

yes The receptor is included in the drawing. The drawing settings for the receptor are taken from the settings in the RECEPTOR menu, *except* that the molecule display mode is set to *lines* for straightforward visualization with the ligand. The receptor is drawn to the graphics object set for the receptor. In fact, three versions of the receptor are drawn into this graphics object: the receptor as described above, the receptor as described above plus the receptor surface and the receptor as described above plus the interaction surfaces of the hydrogen donor and acceptor groups on the receptor. The three versions are accessible through a slider in FlexV's Object Control window.

no The receptor is not included in the drawing

<overlap> Yes/no answer:

yes Lines are drawn connecting heavy ligand receptor atom pairs which have a non-zero overlap volume.

no The overlap is not shown

<all contact types> Interactions are shown as dotted lines between the interacting group on the receptor and the interacting group on the ligand. Yes/no answer:

yes All interaction (contact) types are shown

no Interactions from a selection of iinteraction (contact) types are drawn. You will be asked to make a selection of types from a given list:

<contact type selection> Choose a list of types represented by integers. Enter the list as separate integers or as integer ranges (format $a - b$) separated by ',' or blanks. Note that you must enclose the expression in quotation marks if it contains blanks. e.g. 1, 2, 4, 7-9

A.3.5.9 Selecting colors for drawing placements (SELCOL)

Syntax: SELCOL <interact color mode> <overlap color mode>

Description: With SELCOL you can set the color modes for drawing the interactions between the fragment or build-up solution and receptor and for the overlap lines. For each of these, a selection of color modes are available:

<interact color mode> Choose the color mode for drawing the dotted lines representing interactions between the ligand and receptor. Color mode selection:

INVISIBLE

UNIQUE

ENERGY
OPT_ENERGY
CONTACT

<**overlap color mode**> Choose the color mode for drawing the solid lines representing atom pairs in the ligand and receptor with non-zero overlap volume.

Color mode selection:

INVISIBLE
UNIQUE

Below, the possible color modes are explained – for some color modes you are also asked to enter some defining colors. Enter your chosen color as either an angle from the color circle (0 – 360 degrees: 0 is invisible, 1–360 runs from dark blue, through red, yellow, green to blue), a color name (as defined in the GRAPHIC static data file), or an RGB(A) value; 3 (4) floating-point numbers separated by blanks or slashes:

Example

```
selcol .... "dark green" ....
selcol .... green ....
selcol .... "0.0 0.8 0.1" ....
selcol .... 0.0/0.8/0.1 ....
selcol .... 220 ...
```

INVISIBLE The item drawn will be invisible.

UNIQUE The object will be drawn in one user-defined color. You will be asked to choose the unique color:

<**color**> Enter your chosen color.

ENERGY Draw the dotted lines showing interactions in a color representative of their energy. A color rainbow will be defined between two given colors across the range of two given energies. You are required to enter:

<**no. of intervals**> Enter the number of intervals (integer) that the energy range will be split into

<**min energy**> Enter the minimum energy value (floating-point number) for the start of the energy range. The default offered is the best achievable score for an interaction in FlexNovo (-8.3)

<**max energy**> Enter the maximum energy value (floating-point number) for the end of the energy range. The default value of 0.0 is offered.

<**first color**> Enter the first color of the color rainbow

<**second color**> Enter the second (end) color of the color rainbow

OPT_ENERGY Draw the dotted lines showing interactions in a color representative of the optimal energy achievable for this interaction. (The energy for the interaction at the optimal geometry of the two interacting groups.) A color rainbow will be defined between two given colors across the range of two given energies. You are required to enter:

<**no. of intervals**> Enter the number of intervals (integer) that the energy range will be split into

<**min energy**> Enter the minimum energy value (floating-point number) for the start of the energy range. The default offered is the best achievable score for an interaction in FlexNovo (-8.3)

<**max energy**> Enter the maximum energy value (floating-point number) for the end of the energy range. The default value of 0.0 is offered.

<**first color**> Enter the first color of the color rainbow

<**second color**> Enter the second (end) color of the color rainbow

CONTACT The object will be drawn in a color representing its interaction (contact) type. The colors for each type are defined in the **GRAPHIC** static data file. Note: the colors are taken from the contact types on the protein side.

A.3.5.10 Selecting labels for drawing the placements (**SELLAB**)

Syntax: SELLAB <ia type> <energy> <opt energy>

Description: When the placements are drawn, FlexNovo stores information about the interactions between the fragment or build-up solution and receptor for display in the graphics interface. You can choose what should appear in the label using the **SELLAB** command. For yes/no questions you can enter either 'y', 'yes' or '1' for yes, and similarly 'n', 'no' or '0' for no.

<**ia type**> Yes/no answer:

yes Include the type of the interaction in the label

no Do not include the type in the label

<**energy**> Yes/no answer:

yes Include the actual energy score of the interaction in the label

no Do not include the energy score in the label

<**energy**> Yes/no answer:

yes Include the optimal achievable energy score for this interaction in the label – the optimal energy occurs when the positions of the two interacting groups have an optimal geometry

no Do not include the optimal energy score in the label

A.3.5.11 Listing the graphic items (**GRAINF**)

Syntax: GRAINF

Description: Outputs a list of all current graphic settings for drawing fragment or build-up solution placements.

A.3.5.12 Drawing multiple placements (MDRAW)

Syntax: MDRAW <drawing mode> <selection> <placement selection>
<multiple sliders>

Description: MDRAW draws a fragment or build-up solution selection to the graphical viewer specified in the `config.dat` file. Additionally, a selection of placements can be made, so that for each selected fragment or build-up solution, only the selected placements will be drawn. The selections are lists of integers or integer ranges (format a-b) representing fragment build-up solution rank IDs or placement ID's separated by ',' or blanks. Finally, all fragments or build-up solutions and the corresponding placements can be drawn to a single or multiple sliders.

A.4 FlexNovo-Beispieldateien

Der folgende Abschnitt enthält die Konfigurationsdatei sowie vier Beispielskripte aus dem FlexNovo-Benutzerhandbuch. Die Form und Syntax der Konfigurationsdatei sowie der Skripte ist dabei sehr ähnlich zu denen von FlexX.

A.4.1 Konfigurationsdatei

```
#-----#
#
# CONFIG.DAT: Configuration file for the FlexNovo program, version 1.4.x #
#
#-----#

# -----#
# LICENSE FILES
# -----#
# DESCRIPTION:
# FlexNovo starts only with valid license strings. These must be issued
# by the research group for computational molecular design at the Center
# for Bioinformatics of the University of Hamburg and by BioSolveIT GmbH
# respectively. Section LICENSE_FILES is not required if FlexLM is used.
# -----#
@LICENSE_FILES
#

# -----#
# ROOT DIRECTORY
# -----#
# DESCRIPTION:
# FlexNovo needs the definition of some paths and files, for example where
# it should search for fragments, receptors, etc. To make the definition of
# these paths easier, you can define a root directory. Each subsequent
# path definition starting with a directory name or ../ is then relative
# to this directory. Note that ./ is relative to the current directory.
# REMARKS:
# The root directory should be an absolute path.
# Windows: Path names including spaces are supported but must be quoted
# using "<path>"
# -----#
@ROOTDIR      <installation directory>

# -----#
@DIRECTORIES
# -----#
# DESCRIPTION:
# FlexNovo uses user-defined standard directories for all kind of files.
# This is the place where you can define them. If a file name is entered
# in FlexNovo, the program looks first for a local file and then in the
# standard directory. The following standard directories can be defined:
#
# Each path may contain multiple directory names separated by ':' or '+'.
# Directory names may be relative or absolute as described in section
# 'ROOT DIRECTORY'.
#
# Fragspace:
# FRAGSPACE      : read, location of mol2 files of the fragment
# Receptor:
# RECEPTOR     : read, location of the rdf files (receptor description files)
# PDB            : read, location of pdb files of the receptor
```

A.4 FlexNovo-Beispieldateien

```
# SURFACE      : read, location of the sdf files (surface description files)
# SITE : read, location of the active site files
# Ligand:
# LIGAND       : read, location of mol2 files of the ligand
# Optional FlexNovo modules:
# PHARM        : read, location of pharmacophore description files
# Misc.:
# SCRIPT       : read, place where batch files are located
# HELP        : read, location of the FlexNovo User Guide and short help file
# PREDICT     : write, place where output of FlexNovo is written to
# TEMP : write, space for temporary files
# PVM_TEMP    : write, space for pvm temporary files
#
# -----

#application PHARM
PHARM          example/pharm/
#end_application

#application FRAGSPACE
FRAGSPACE      example/fragspace/
#end_application

RECEPTOR     example/receptor/rec/
PDB            example/receptor/pdb/
SURFACE        example/receptor/surf/
SITE           example/receptor/site/
LIGAND         example/ligand/

SCRIPT         example/bat/

HELP           help/

PREDICT        predict/
TEMP           /tmp/
PVM_TEMP       /tmp/

# -----
@STATIC_DATA
# -----
# DESCRIPTION:
# FlexNovo needs a lot of background knowledge which is stored in a set of
# ascii data files. A description of the content of the different
# files can be found in the User Guide.
# REMARKS:
# FlexNovo must have all these files. If one of them is not present under
# the described path, FlexNovo terminates directly in the initialization
# phase.
# SHORT DESCRIPTIONS:
# (for a more detailed description refer to the manual)
# SETTINGS      : tool specified parameters
# CHEMPAR       : general chemistry knowledge of FlexNovo
# AMINO         : amino acid data
#                amino.dat
#                amino_gen.dat (use for FlexE)
#
# Note: The files amino.dat and amino_gen.dat
#       should be unpacked from the package, therefore the following lines
#       AMINO static_dat/amino.dat
#       and
#       AMINO static_dat/amino_gen.dat
#       have to be given below. After the two files have been
```

A Die Benutzungsschnittstelle von FlexNovo

```
#           unpacked, one of the lines must be removed according
#           to if the modul FlexE will be used or not
#
# CHARGES   : charges
#           amino_pcharges.dat
#           amino_pcharges_gen.dat (use for FlexE)
#
# Note: The files amino_pcharges.dat and amino_pcharges_gen.dat
#       should be unpacked form the package, therefore the following lines
#       CHARGES static_dat/amino_pcharges.dat
#       and
#       CHARGES static_dat/amino_pcharges_gen.dat
#       have to be given below. After the two files have been
#       unpacked, one of the lines must be removed according
#       to if the modul FlexE will be used or not
#
# GEOMETRY  : interaction geometries
# FCHARGES  : formal charges
# DELOC     : delocalization of electrons
# CONTACT   : intermolecular contacts
# TORSION   : torsional angle data base (see flag TORSION_MODE)
#           torsion_standard.dat (=>TORSION_MODE 0)
#           torsion_fine.dat    (=>TORSION_MODE 0/1/2)
#
# Note: The files torsion_standard.dat and torsion_fine.dat
#       should be unpacked form the package, therefore the following lines
#       TORSION static_dat/torsion_standard.dat
#       and
#       TORSION static_dat/torsion_fine.dat
#       have to be given below. After the two files have been
#       unpacked, one of the lines must be removed according
#       to which torsion database should be used in FLEXNOVO.
#
# CONTYPE   : contact types
# GRAPHIC   : graphic output, colors
#
# -----
SETTINGS    static_data/flexnovo_settings.dat
CHEMPAR     static_data/chempar.dat
AMINO       static_data/amino_gen.dat
AMINO       static_data/amino.dat
CHARGES     static_data/amino_pcharges_gen.dat
CHARGES     static_data/amino_pcharges.dat
GEOMETRY    static_data/geometry_post_screenscore.dat # example ScreenScore for post scoring
GEOMETRY    static_data/geometry_screenscore.dat     # example ScreenScore for scoring
GEOMETRY    static_data/geometry_plp_score.dat       # example PLP scoring
GEOMETRY    static_data/geometry_chemscore.dat      # example ChemScore scoring
GEOMETRY    static_data/geometry.dat                 # standard FlexX-Score
FCHARGES    static_data/fcharges.dat
DELOC       static_data/delocalized.dat
CONTACT     static_data/contact.dat
TORSION     static_data/torsion_fine.dat
TORSION     static_data/torsion_standard.dat
CONTYPE     static_data/contype.dat
GRAPHIC     static_data/graphic.dat

GENERIC_RDF static_data/generic_metal.rdf

#application LOGP
LOGP        static_data/logp.dat
#end_application
```

```

# Macro filter definitions
FILTER          static_data/filter_macros.dat

# -----
@PROGRAMS
# -----
# DESCRIPTION:
# FlexNovo executes some external programs. All necessary programs are
# part of the FlexNovo distribution.
# REMARKS:
# EDITOR       : must be defined if you want to edit static data files
# VRML_VIEWER  : must be defined if you want to visualize results with the
#                Virtual Reality Modeling Language (GRAPHIC_MODE set to 1)
# PDF_VIEWER   : must be defined if you want to use the manual online
# FLEXNOVO     : location of the FlexNovo executable program
# FLEXV        : must be defined if you want to visualize results with the
#                FlexV-3D viewer (GRAPHIC_MODE set to 2)
# RCGENERATOR  : optional but highly recommended, must be defined if you
#                want to handle flexible ring systems
# CONV_SMILES  : optional, must be defined if you want to import from
#                SMILES
# 3DGENERATOR  : optional, must be defined if you want to clean-up molecule
#                structures or if you want to read files in Sybyl Line
#                Notation
# -----

# programs for all computer systems
EDITOR /usr/bin/vi
VRML_VIEWER /usr/bin/vrweb -bg black -remote
PDF_VIEWER ./acroread

# Intel based PCs with Linux
#system Linux
  FLEXNOVO    bin_Linux/flexnovo
  FLEXV       bin_Linux/flexv -size 750 750 -pos 400 400
  RCGENERATOR bin_Linux/corina -d flexx -i dummies
  CONV_SMILES bin_Linux/corina -i t=smiles,dummies -o t=mol2,m2l -dwh
  3DGENERATOR bin_Linux/corina -i t=mol2,dummies -o t=mol2,m2l
# 3DGENERATOR bin_Linux/corina -i t=sdf,dummies -o t=sdf,m2l
#end_system

#system Linuxx86_64
  FLEXNOVO    bin_Linuxx86_64/flexnovo
  FLEXV       bin_Linuxx86_64/flexv -size 750 750 -pos 400 400
  RCGENERATOR bin_Linuxx86_64/corina -d flexx -i dummies
  CONV_SMILES bin_Linuxx86_64/corina -i t=smiles,dummies -o t=mol2,m2l -dwh
  3DGENERATOR bin_Linuxx86_64/corina -i t=mol2,dummies -o t=mol2,m2l
# 3DGENERATOR bin_Linuxx86_64/corina -i t=sdf,dummies -o t=sdf,m2l
#end_system

# ----- #
#                END OF PATH, FILE AND PROGRAM DEFINITION #
# the remaining part of this file is not important for installing FlexNovo #
# ----- #

# -----
@FLAGS
# -----
# DESCRIPTION:
# Elementary features of FlexNovo are controlled by a set of flags. You can
# define the default value for this flags here. The flags and their
# values are:

```

A Die Benutzungsschnittstelle von FlexNovo

```
# VERBOSITY: defines the level of verbosity, the following is printed:
# 0 : nearly silent, only warnings, error messages and direct results
# 1 : user messages, displaying the program flow
# 2 : important statistical output
# 3 : runtime information (if the output is directed into a file, the
#     verbosity level must be less than 3)
# 9 : all except control output of READ and WRITE commands
# USER_MODE: defines the user mode (enables/disables menus and commands)
# 0 : off, complete menu not present
# 1 : standard user mode
# 2 : advanced user mode
# 3 : expert user mode
# IO_RETRY: if != 0 and FlexNovo is started with the -b option, FlexNovo tries
# to open a file io_retry times (with about 2 minutes delay) before
# printing an error message.
# -1 : never give up
# x : give up after x retries
# PRINT_TIMES: printing runtimes
# 0 : off
# 1 : on, prints the elapsed process time after each command
# PRINT_SIZE: printing current process size (kB)
# 0 : off
# 1 : on, prints the process size after each command
# SIZE_LIMIT: terminates the FlexNovo process, if the memory requirement
# exceeds the limit
# 0 : off
# x : size limit of x kBytes (check after each command)
# GRAPHIC_MODE: selects the graphic front-end
# 1 : vrml
# 2 : flexv
# PLACE_PARTICLES: places particles (i.e. water) during placement
# computation
# 0 : off
# 1 : on, see amino.dat for definition of particles
# RING_MODE: way of computing ring conformations
# 0 : RIGID RINGS (take input coordinates)
# 1 : CORINA
# 2 : CONFORT
# RIGID_TORSIONS: switches flexibility of torsion angles on or off
# 0 : on
# 1 : off
# TORSION_MODE: selection of the torsion angle model
# If the TORSION set to torsion_standard.dat, the flag
# TORSION_MODE must be 0. But if variable TORSION set to
# torsion_fine.dat, the flag TORSION_MODE may be 0, 1 or 2.
# But it should be 2.
# 0 : old MIMUMBA model (torsion angles are intersections of torsion
# angle sets of different patterns)
# static-data file: torsion_standard.dat (or torsion_fine.dat)
# 1 : new MIMUMBA model (merging the statistics)
# Remark: you must change the torsion database together with
# this flag
# static-data file: torsion_fine.dat
# 2 : new MIMUMBA model without preselection of angles (take all 10 deg.)
# Remark: you must change the torsion database together with
# this flag
# static-data file: torsion_fine.dat
# SECONDARY_TORSION_MODE: calculation of torsion angles
# 0 : off
# 1 : on, if no torsion angles for a rotatable bond defined in torsion db
# the torsion potential is calculated in force field and appropriate
# angles are selected
```

```

# KEEP_RCGEN_FILES:
# 0 : delete corina in/out/trace files
# 1 : save all RC generator (CORINA, CONFORT) files and print files names
#     on the screen
# STEREO_MODE:
# 0 : take enantiomer from input mol2 file
# 1 : allow modification of N.3,0.3,S pseudo stereo centers
# 2 : allow modification of double bond E/Z stereo centers
# 4 : allow modification of R/S stereo centers
#     (can be added up, for example: 7 is all, 5 is pseudo N.3,0.3,S and R/S, etc.)
# ASSIGN_FORMAL_CHARGES: performs an automatic assignment of formal
#     charges to the atoms of the fragment
# 0 : off
# 1 : on,
# ASSIGN_DELOCALIZED: automatic assignment of delocalized atom types,
#     bond types, and formal charges
# 0 : off
# 1 : on,
# TEST_ATOM_TYPES: tests the correct assignment of atom types and bond
#     types in fragment files.
# 0 : make no tests
# 1 : test the atom and bond types (makes no assignments)
# 2 : assign atom and bond types
# 3 : never assign atom and bond types even if they are missing
# SDF_MOL_ID_TYPE: determines the field from which the molecule ID in a SD
#     File is taken
# 0 : First line of header block
# 1 : Property line with the name given by SDF_MOL_ID_STRING
# 2 : Property line starting/ending with <ID.. / <id.. / ..ID> / ..id>
# 3 : Take the SDF_MOL_ID_NUMth field in the data section
# SDF_MOL_ID_NUM: determines the field from which the molecule ID in a SD
#     File is taken, if SDF_MOL_ID_TYPE is set to 3
# x : molecule ID is read from the xth property line of type '> <..>'
# ADD_HYDROGENS: automatic addition of missing hydrogens to fragment;
# 0 : no automatic addition of hydrogens
# 1 : add hydrogens according to formula and formal charges from file
# 2 : protonate all acids, keep all amines deprotonated
# 3 : keep acids deprotonated, keep all amines deprotonated
# 4 : keep acids deprotonated, protonate tert. amines
# 5 : keep acids deprotonated, protonate sec.+tert. amines
# 6 : keep acids deprotonated, protonate all amines
# 3D_GEN: perform a (re)generation of 3D coordinates for the fragment molecule structure
#     (necessary and forced activation for fragments from SLN or SMILES files;
#     useful if fragment has bad geometry).
# 0 : off, no clean-up done
# 1 : use clean-up tool defined by 3DGENERATOR
# 2 : if self-overlap test fails, use clean-up tool defined by 3DGENERATOR
# 3D_GEN_FORMAT: select a file-format that should be used as input/output for
#     the 3DGENERATOR executable
#     Note: if this is changed, 3DGENERATOR needs parameter adjustment, too !!
# 0 : SDF
# 1 : MOL2
# KEEP_3D_GEN_FILES:
# 0 : delete cleanup in/out/trace files
# 1 : save all cleanup program files and print files names on the screen
# AUTO_MINIMIZE: use Tripos force field for minimization of fragment
#     internal energy. Slow! Use only, if initial geometry is bad and
#     no structure clean-up program available.
# 0 : off
# 1 : on
# FF_CHARGES: use Coulomb interactions with partial charges in force
#     field. Should be switched off for Tripos force field.

```

A Die Benutzungsschnittstelle von FlexNovo

```
# 0 : off
# 1 : on
# FF_CONVERGENCE_CRIT: convergence criterion in force field energy
#   minimization. Should be set to 1.
# 1 : normal convergence
# 2 : high convergence (very slow)
# FF_DIELECTRIC_FUNC: handling of dielectric constant in electrostatic
#   energy calculation of force field. Should be set to 1.
# 1 : use constant dielectric function
# 2 : use distant dependent dielectric function (scaled by distance r)
# FF_PREOPTIMIZE: use of a Nelder-Mead simplex minimization for initial
#   minimization steps. Should be switched on.
# 0 : off
# 1 : on
# FF_USE_INTRA_CUTOFF: use of intramolecular cut-off in calculation of
#   non-bonding force field energy terms. Should be switched off.
# 0 : off, no cut-off
# 1 : on, use a cut-off of FF_NONBOND_CUTOFF (defined in flexnovo_settings.dat)
# FLEXIBLE_OPTIMIZE: uses a new flexible (torsion-wise) optimizer
# 0 : uses old rigid optimizer for post-optimization only.
#   Requires explicit call of BUILDUP/OPTIMIZE!
# 1 : uses the new flexible optimizer for post-optimization. Again,
#   requires explicit call of BUILDUP/OPTIMIZE!
# 2 : uses the new flexible optimizer - also during incremental
#   construction (untested feature; no explicit optimize call
#   needed!)
# MOL_NAME: switches between different output name strings
# 1 : mol2 molecule name (from input file)
# 2 : mol2 molecule name + infile number (multi mol file) or
#   mol2 molecule name + solution number (placement solution)
# 3 : output filename + infile number (multi mol file) or
#   output filename + solution number (placement solution)
# RESTRICT_COMP_TO_STRUCTURE: combine only conformers of the same
#   ensemble structure. NOTE: FlexE becomes slow if set to 1.
# 0 : no restrictions
# 1 : only conformers of the same ensemble structure are compatible
# USE_PVM_FEATURE:
# 0 : a batch script is executed only sequentially
# 1 : the execution of a parallel batch file is available
# PDF_COMPRESSION:
# 0 : placement files are written in standard format
# 1 : placement files are written in compressed gzip format
# QUERY_SASTAB
# 0 : disable sastab computation with listsol (default)
# 1 : enable sastab for listsol
# QUERY_BURIEDNESS
# 0 : disable buriedness computation with listsol (default)
# 1 : enable buriedness for listsol
# -----
# User interface
VERBOSITY          3
IO_RETRY           1
USER_MODE          3
PRINT_TIMES        1
PRINT_SIZE         1
SIZE_LIMIT         0
GRAPHIC_MODE       2

# fragment flexibility handling
RIGID_TORSIONS     0
RING_MODE          1
TORSION_MODE       0
```

```

# fragment data preparation
TEST_ATOM_TYPES      1
ADD_HYDROGENS        0
ASSIGN_FORMAL_CHARGES 1
ASSIGN_DELOCALIZED   1
KEEP_RCGEN_FILES     0
STEREO_MODE          7
SECONDARY_TORSION_MODE 0

# Placing particles
PLACE_PARTICLES      0

# Flexible optimization during complex buildup
FLEXIBLE_OPTIMIZE    1

# Preprocessing and consistency checking of input data
SDF_MOL_ID_TYPE      1
SDF_MOL_ID_NUM       1
3D_GEN               2
3D_GEN_FORMAT        1 # 0 - SDF / 1 - MOL2
KEEP_3D_GEN_FILES    0
MOL_NAME             1

# Options for the build-in force field
AUTO_MINIMIZE        0
FF_CHARGES           0
FF_CONVERGENCE_CRIT  1
FF_DIELECTRIC_FUNC   1
FF_PREOPTIMIZE       1
FF_USE_INTRA_CUTOFF  0

# PVM flag
USE_PVM_FEATURE      0

# Query flags
QUERY_SASTAB         0
QUERY_BURIEDNESS     0

#application FRAGSPACE
PDF_COMPRESSION      1
#end_application

# -----
@ID_STRINGS
# -----
# DESCRIPTION:
# This optional section is for parsing of SD Files (SDF) and Sybyl Line
# Notation (SLN) only. The field names that determine the molecule-ID
# can be defined here. If they are not defined, "ID" is assumed for SDF
# and "RegID" for SLN. Both values cannot be changed on runtime of FlexNovo!
# -----
SDF_MOL_ID_STRING    MOLREGNO
SLN_MOL_ID_STRING    RegID

# -----
@PARALLEL
# -----
# DESCRIPTION:
# This section is for FlexNovo with PVM interface only. It describes the hosts
# where a parallel script can be executed. After each host name, the maximal
# number of allowed processes and the optional nice value follows.

```

A Die Benutzungsschnittstelle von FlexNovo

```
# -----  
# -----  
@ALIASES  
# -----  
# DESCRIPTION:  
# To make working with FlexNovo easier, you can define an alias for each  
# command and sub-menu in FlexNovo. Commands and sub-menu names are not  
# case sensitive.  
# REMARKS:  
# You can only define ONE alias per command or sub-menu.  
# -----  
# general commands / main menu commands  
END          E  
QUIT        Q  
QUIT|Y      X  
SCRIPT      S  
LIST        L  
HELP        H  
TOFLEXV     TOF  
# submenus  
RECEPTOR  REC  
LIGAND      LIG  
DATABASE    DB  
# special menu commands  
WRITE       W  
READ        R  
READREF     RR  
QUERY       Y  
LISTSOL     LS  
LISTMAT     LM  
LISTALL     LA  
# grafic alias  
DISPLAY     GO  
SELGRA      SG  
SELCOL      SC  
SELADM      SA  
  
# -----  
# End of FlexNovo configuration file  
# -----
```

A.4.2 Skripte für typische Berechnungen

Mit den folgenden Skripten können typische Rechnungen mit FlexNovo durchgeführt werden. Dabei sind für das Preprocessing und für den eigentlichen Aufbaualgorithmus jeweils Beispiele für sowohl sequenzielle als auch parallele Berechnungen enthalten.

Script 1: Preprocessing

The first example is a simple script which loads a protein, the pharmacophore information and the fragment space in a first step. In the second phase all steps necessary for running the actual FlexNovo calculation are performed.

Example

```
# SCRIPT: Perform docking calculations under pharmacophore constraints
#         and store placements and docking info output
#
# parameters are: $(protein)      = name of protein rdf file
#                 $(fragspace)   = name of the fragspace fsf file
#                 $(anchor_pharm) = name of anchor-pharmacophore definition
#                 $(filter)      = name of the filter definition file
#                 $(placements)  = name of the directory for the placement files
#                 $(info_pharm)  = name of the file with the scores of the
#                               calculation under pharmacophore-type constraints
#                 $(info_score)  = name of the file with the best estimated scores
#                               of all fragments of the fragment space

# Part 1: Loading data
RECEPTOR
  READ "$(protein)"
END
PHARM
  READ "$(anchor_pharm)"
END
FRAGSPACE
  READ "$(fragspace)"
END

# Part 2: Preprocessing phase
FILTER                                # optional step:
  READ "$(filter)"                    # read filter file if you want to do prefiltering
  APPLY a                              # generate selection according to the criteria \
END                                    # specified (optional)
PLACING
  # generate placements for all fragments in the selection and store the \
  # fragment scores and placement files on disk
  PLACE s 3 y "$(info_pharm)" n 7 y y "$(placements)" n
END
PHARM
  DELETE                              # delete pharmacophore
END
PLACING
  # generate placements for all fragments of the fragment space and store \
  # the fragment score information only
  PLACE a 3 y "$(info_score)" n 7 n n
END
```

Script 2: Build-up calculation

The second example uses the data generated by the first script, performs a build-up calculation and prints some statistical information as well as the list of resulting molecules on the screen.

Example

```

# SCRIPT: Perform a build-up calculation with a set of fragments
#
# parameters are: $(protein)      = name of protein rdf file
#                   $(fragspace)   = name of the fragspace fsf file
#                   $(buildup_pharm) = name of buildup pharmacophore definition
#                   $(info_pharm)   = name of the file with the scores of the
#                   calculation under pharmacophore-type constraints
#                   $(info_score)  = name of the file with the best estimated scores
#                   of all fragments of the fragment space
#                   $(filter)      = name of the filter definition file
#                   $(placements)  = name of the directory with the start placements
#                   $(infofile)    = name of the file for writing the results info
#                   $(results)     = name of the directory for the results

# Part 1: Loading data
RECEPTOR
  READ "$(protein)"
END
FRAGSPACE
  READ "$(fragspace)"
END
PHARM
  READ "$(buildup_pharm)"
END
PLACING
  INIT "$(info_pharm)" # load the scores of the pharm-calculation
  SELECT b 100         # select the 100 best scoring fragments
  INIT "$(info_score)" # load the scores of the score-estimation-calculation
  LOAD s "$(placements)" # load starting placements
END
FILTER
  READ "$(filter)" # load filter definition file
END

# Part 2: Calculation phase
BUILDUP
  GREEDY s           # start greedy buildup using all fragments with /
                    # placements information as starting fragments. All /
                    # fragments will be used further in the buildup process
  INFO               # print some statistical information
  LISTSOL %         # print list of resulting compounds
                    # write solutions to disk
  WRITE a "$(results)/$(fragspace)" n y y 1 y y "$(results)"
  SELOUTP "$(infofile)" a y
  SOLINFO 0
  SOLINFO 1 a
END

```

Script 3: Preprocessing in parallel

The third example goes through a list of fragments and places them into the same active site. A placement file for each fragment is written to a file together with a text file containing the fragment name and the score of the corresponding best pose predicted.

Example

```
# SCRIPT: Place a set of fragments into one receptor and save placements
#
# parameters are: $(protein)      = name of protein rdf file
#                   $(fragspace)  = name of the fragspace fsf file
#                   $(anchor_pharm) = name of anchor-pharmacophore definition
#                   $(filter)     = name of the filter definition file
#                   $(placements) = name of the directory for the placement files
#                   $(info_pharm) = name of the file with the scores of the
#                   calculation under pharmacophore-type constraints
#                   $(info_score)  = name of the file with the best estimated scores
#                   of all fragments of the fragment space

# Part 1: Loading data
RECEPTOR
  READ "$(protein)"
END
PHARM
  READ "$(anchor_pharm)"
END
FRAGSPACE
  READ "$(fragspace)"
END

# Part 2: go through the list of fragments
for_each $(frag) in $(fraglist)
  output " --- Processing fragment " $(frag)
  FRAGSPACE
    SELECT not_def not_def not_def not_def not_def "$(frag)"
  END
  PLACING
    # generate placements for the selected fragment and store its best score \
    # and the placement file on disk
    PLACE s 3 y "$(info_pharm)" n 7 y y "$(placements)" n
  END
  PHARM
    DELETE
  END
  PLACING
    # generate placements for the selected fragment and store \
    # the fragment score information only
    PLACE s 3 y "$(info_score)" n 7 n n
  END
end_for
```

The file `$(fraglist)` must be located in the directory where FlexNovo is executed. It contains one line per fragment with the fragment name.

Script 4: Build-up calculation in parallel

The forth example goes through a list of fragments, loads starting orientations and score evaluation from a file and starts the build-up calculation. The parameters used for this calculation have to be different compared to the sequential case!

Example

```
# SCRIPT: Perform a greedy sequential buildup with a set of fragments
#
# parameters are: $(protein)      = name of protein rdf file
#                   $(fragspace)  = name of the fragspace fsf file
#                   $(buildup_pharm) = name of buildup pharmacophore definition
#                   $(info_pharm)  = name of the file with the scores of the
#                                   calculation under pharmacophore-type constraints
#                   $(info_score)  = name of the file with the best estimated scores
#                                   of all fragments of the fragment space
#                   $(filter)      = name of the filter definition file
#                   $(placements)  = name of the directory with the start placements
#                   $(infofile)    = name of the file for writing the results info
#                   $(results)     = name of the directory for the results

# Part 1: Loading data
RECEPTOR
  READ $(protein)
END
FRAGSPACE
  READ $(fragspace)
END
PHARM
  READ "$(buildup_pharm)"
END
PLACING
  INIT "$(info_score)" # load the scores of the score-estimation-calculation
END
FILTER
  READ "$(filter)" # load filter definition file
END

# Part 2: go through the list of fragments
for_each $(frag) in $(fraglist)
  output " --- Build-up fragment " $(frag)
  FRAGSPACE
    SELECT not_def not_def not_def not_def not_def "$(frag)"
  END
  PLACING
    LOAD s "$(placements)" # load starting placements
  END
  BUILDUP
    GREEDY s
    INFO
    LISTSOL %
    WRITE a "$(results)/$(frag)" n y y 1 y y "$(results)"
    SELOUTP "$(infofile)" a y
    SOLINFO 1 a
  END
end_for
```

The file `$(fraglist)` must be located in the directory where FlexNovo is executed. It contains one line per fragment with the fragment name.

B Implementierung

Im folgenden Kapitel sind die Details der Implementierung von FlexNovo zusammengefasst. Dabei wird auf die einzelnen Teile der Programmbibliothek eingegangen und die jeweiligen Module sowie deren Inhalte kurz erläutert. Daran schließt sich die dokumentierte Versionshistorie der FlexNovo-Release-Versionen an.

B.1 Die Programmbibliothek

FlexNovo ist in ANSI C entwickelt und besteht zurzeit aus etwa 25000 Zeilen Programmcode. Die Implementierung gliedert sich in mehrere Teilbereiche. Den größten Teil nimmt dabei die *Fragmentraum-Bibliothek* ein. Darauf bauen ein Modul für physikochemische Filterkriterien sowie ein Docking-Modul für Fragmente auf. Letzteres greift dabei unter anderem auf die FlexX-Algorithmen zur Platzierung von Fragmenten zu. Die eigentliche FlexNovo-Funktionalität befindet sich im Buildup-Modul.

B.1.1 Das Fragspace-Modul

Die Quellcode-Dateien des Fragspace-Moduls sind in der folgenden Tabelle zusammengestellt und werden kurz hinsichtlich ihres Inhalts charakterisiert.

Tabelle B.1: Bestandteile des Fragspace-Moduls

Datei	Inhalt
menu_fragspace.h	Deklarationen der Menü-Funktionen für die Kommandos im Untermenü FRAGSPACE
menu_fragspace.c	Implementierung der Menü-Funktionen für die Kommandos im Untermenü FRAGSPACE
fragspace.h	Datentypen und Funktionsprototypen zur Speicherung und Verwaltung von Fragmenträumen
fragspace.c	Funktionen zur Speicherung und Verwaltung von Fragmenträumen
fragspace_combine.h	Funktionsprototypen zur Verknüpfung von Fragmenten
fragspace_combine.c	Funktionen zur Verknüpfung von Fragmenten

Fortsetzung auf nächster Seite

Tabelle B.1 – Fortsetzung von vorheriger Seite

Datei	Inhalt
fragspace_database.h	Funktionsprototypen zur Verwaltung und Abfrage der internen Datenbanken mit Platzierungsinformationen für Fragmente
fragspace_database.c	Funktionen zur Verwaltung und Abfrage der internen Datenbanken mit Platzierungsinformationen für Fragmente
fragspace_descriptor.h	Datentypen und Funktionsprototypen zur Berechnung und Verwendung des physikochemischen Deskriptors
fragspace_descriptor.c	Funktionen zur Berechnung und Verwendung des physikochemischen Deskriptors
fragspace_fragments.h	Funktionsprototypen zur Verwaltung, Manipulation und zum Speichern und Laden von Fragmenten
fragspace_fragments.c	Funktionen zur Verwaltung, Manipulation und zum Speichern und Laden von Fragmenten
fragspace_geometry.h	Funktionsprototypen zur Berechnung der Geometrieanpassungen bei der Fragmentverknüpfung
fragspace_geometry.c	Funktionen zur Berechnung der Geometrieanpassungen bei der Fragmentverknüpfung
fragspace_graphic.h	Funktionsprototypen zur Verwaltung, Manipulation und Darstellung von Fragmenten und Platzierungen
fragspace_graphic.c	Funktionen zur Verwaltung, Manipulation und Darstellung von Fragmenten und Platzierungen
fragspace_molecule.h	Funktionsprototypen zur Annotation und Manipulation von Daten an molekularen Graphen
fragspace_molecule.c	Funktionen zur Annotation und Manipulation von Daten an molekularen Graphen
fragspace_rules.h	Funktionsprototypen zur Verwaltung und Manipulation des Regelwerks eines Fragmentraums
fragspace_rules.c	Funktionen zur Verwaltung und Manipulation des Regelwerks eines Fragmentraums
fragspace_selection.h	Datentypen und Funktionsprototypen zur Verwaltung und Manipulation einer Selektion von Fragmenten
fragspace_selection.c	Funktionen zur Verwaltung und Manipulation einer Selektion von Fragmenten

B.1.2 Das Filter-Modul

Die Quellcode-Dateien des Filter-Moduls sind in der folgenden Tabelle zusammengestellt und werden kurz hinsichtlich ihres Inhalts charakterisiert.

Tabelle B.2: Bestandteile des Filter-Moduls

Datei	Inhalt
menu_filter.h	Deklaration der Menü-Funktionen für die Kommandos im Untermenü FILTER
menu_filter.c	Implementierung der Menü-Funktionen für die Kommandos im Untermenü FILTER
fragspace_filter.h	Datentypen und Funktionsprototypen zur Verwaltung, Manipulation und zum Laden von Filter-Spezifikationen
fragspace_filter.c	Funktionen zur Verwaltung, Manipulation und zum Laden von Filter-Spezifikationen

B.1.3 Das Docking-Modul

Die Quellcode-Dateien des Docking-Moduls sind in der folgenden Tabelle zusammengestellt und werden kurz hinsichtlich ihres Inhalts charakterisiert.

Tabelle B.3: Bestandteile des Docking-Moduls

Datei	Inhalt
menu_placing.h	Deklaration der Menü-Funktionen für die Kommandos im Untermenü PLACING
menu_placing.c	Implementierung der Menü-Funktionen für die Kommandos im Untermenü PLACING
fragspace_placements.h	Datentypen und Funktionsprototypen zur Verwaltung, Manipulation sowie zum Speichern, Laden und Anzeigen von Platzierungsinformationen
fragspace_placements.c	Funktionen zur Verwaltung, Manipulation sowie zum Speichern, Laden und Anzeigen von Platzierungsinformationen

B.1.4 Das Buildup-Modul

Die Quellcode-Dateien des buildup-Moduls sind in der folgenden Tabelle zusammengestellt und werden kurz hinsichtlich ihres Inhalts charakterisiert.

Tabelle B.4: Bestandteile des Buildup-Moduls

Datei	Inhalt
menu_flexnovo.h	Deklaration der Menü-Funktionen für die Kommandos im Untermenü BUILDUP
menu_flexnovo.c	Implementierung der Menü-Funktionen für die Kommandos im Untermenü BUILDUP
flexnovo.h	Datentypen und Funktionsprototypen zur Speicherung und Verwaltung der für FlexNovo relevanten Informationen
flexnovo.c	Funktionen Speicherung und Verwaltung der für FlexNovo relevanten Informationen
flexnovo_buildup.h	Datentypen und Funktionsprototypen zur Verwaltung, Manipulation sowie zum Speichern, Laden und Anzeigen von FlexNovo-Ergebnissen
flexnovo_buildup.c	Funktionen zur Verwaltung, Manipulation sowie zum Speichern, Laden und Anzeigen von FlexNovo-Ergebnissen
flexnovo_database.h	Funktionsprototypen zur Verwaltung und Abfrage der internen Datenbanken mit Platzierungsinformationen für FlexNovo-Ergebnisse
flexnovo_database.c	Funktionen zur Verwaltung und Abfrage der internen Datenbanken mit Platzierungsinformationen für FlexNovo-Ergebnisse
flexnovo_pharm.h	Funktionsprototypen zur Verwaltung, Annotation und Manipulation von Pharmakophor-Informationen während des Aufbaualgorithmus
flexnovo_pharm.c	Funktionen zur Verwaltung, Annotation und Manipulation von Pharmakophor-Informationen während des Aufbaualgorithmus

B.2 FlexNovo-Versionshistorie

Im Folgenden ist die Historie der FlexNovo-Release-Pakete zusammengestellt. In Anlehnung an die Struktur der entsprechenden Zusammenstellung für FlexX sind die Änderungen in sieben Kategorien eingeteilt, in denen die unterschiedlichen Teilaspekte jeweils in Stichpunkten zusammengefasst sind.

Die Einträge sind dem FlexNovo-Benutzerhandbuch entnommen und deshalb in englischer Sprache verfasst ([Zahlen] beziehen sich auf interne Referenznummern des Entwicklungssystems).

Version 1.4.2

1. Bug fixes

- Fixed a bug in the fragment initialization routine, which led to inconsistencies in some special cases, for instance, when filtering and placing fragments.
- Fixed a bug in the calculation of enclosing spheres that occurred when estimating the volume of partial solutions. This led to numerical instabilities and assertion failures later on.
- Fixed a bug in the treatment of partial solutions where the removal failed in some rare special cases. This led to a failed assertion later on.

2. Computing

3. Menus and user interface

4. File formats

5. Graphics

6. Misc.

7. Documentation

Version 1.4.0

1. Bug fixes

- Fixed a bug in the temporary file generation where a check was missing if the user is allowed to actually access an already existing file. Before, the subsequent generation of all temporary files failed.
- Fixed a bug in the handling of the molar refractivity values. The filter did not check whether or not a range for mol_refrac was provided in the specification, which resulted in wrongly discarded fragments/solutions.
- Fixed a bug in the PLACING/SELECT command where a check was missing for the 'best # mode' if any scores were annotated at all. This led to a failed assertion later on.
- Fixed a number of (minor) bugs and special cases that occurred during transfer, counting, and loading of placements.
- Fixed a bug when writing fragments or solutions in SD format that led to program crashes before.

2. Computing

- A 'complete' pharmacophore definition can now be used during the build-up calculation. The specification of the pharmacophore has to be done in such a way, that the algorithm is able to iteratively fulfill one or more additional constraints in each fragment addition step, i.e. in each extension cycle. Naturally, all final solutions are checked against the complete pharmacophore definition. Note that logical expressions cannot be used in this context.
- The 3DGENERATOR program can now be used. It generates low-energy (input) coordinates during the initialization of fragments and partial solutions. If the corresponding flag 3D_GEN is set to 2 (default), the program will try to correct the coordinates if the fragment is self-overlapping. If the value of the flag is set to 1, EVERY fragment and solution that is generated will be processed, which will result in a huge increase in computation time. If the flag is set to 0, the program will not be used.
- There are now the manual and perturbation modes available for the placing of fragments. These modes work correctly only if the coordinates of the corresponding fragment(s) are already within the active site of the receptor. The algorithm then generates a single (manual) or a low number (perturbate) of sampled start placements for the fragment(s) that are derived from the input coordinates. These can then be used as start placement(s)s for the build-up calculation.
- The SIS base placement algorithm is now available. It will be used automatically during the fragment placing if no solution could be obtained by either of the triangle (mode 3) or the line (mode 2) modes. It can also be selected explicitly (mode 1) by the user.

3. Menus and user interface

- When writing fragments or flexnovo solutions to disk there is now a default value for the coordinate selection given and used, in case the user does not provide one. Before nothing happened and an error message was printed.

4. File formats

- Improved the support for output in SD format a little.

5. Graphics

6. Misc.

- Increased the default value for the NOF_PLACEMENTS parameter in the flexnovo settings file in order to better account for the more fine-grained torsional sampling of links, which is a result of an earlier bug fix. Depending on the size of the fragment space, this may increase runtime.

7. Documentation

Version 1.3.6

1. Bug fixes

- Fixed an error in the ranking of fragment placements that led to instabilities in the sorting function in case two placements cannot be distinguished from each other by (dock_entry) key values. [186]
- Fixed a bug in the clustering of fragment placements that led to a very coarse-grained sampling with respect to the torsional angles retained at the end. Additionally, for intermediate solutions of the build-up algorithm, the larger RMS value is now used for clustering. [187]
- Added some more consistency checks for the fragment space file parser that now checks also for the definition of the link compatibility matrix.

2. Computing

- Significantly improved the treatment of dummy/link atoms: The assignment of sybyl atom types and vdW-volumes for links is now more consistent, especially with respect to the TEST_ATOM_TYPES flag in the config.dat file and the values specified in the champar.dat file.

3. Menus and user interface

- The build-up function now prints also a summary in case there are no extendable solutions found in an extension cycle.

4. File formats

- Increased the radii of dummy atoms in the chempar.dat file to the size of a hydrogen atom. Fixed also an error in the specification of sybyl atom types (table @atom_types) for dummy atoms.

5. Graphics

6. Misc.

7. Documentation

Version 1.3.5

1. Bug fixes

- Fixed a bug in the processing of the reference fragments. These were extended in the 'wrong' order according to the score, i.e., from low scores to high ones, instead of from high scores to low ones.
- Fixed a bug that occurred upon transferring partial solutions. In some cases, the matches were not transferred correctly so that one match would be missed and the other would be counted twice. This led also to different scores of the transferred solution. Additionally, the first set of coordinates of the reference was transferred twice to the target, and the last coordinate set of the reference was therefore discarded. This bug is also fixed now [185]
- Fixed a bug in the build-up algorithm where inclusion SMARTS pattern were applied to intermediate results instead of to final results only.

- Fixed a bug in the calculation of the build-up statistics where the amount of placements generated was not counted correctly in some cases.
- The buriedness and saturation placement filter functions were not checked correctly when carrying out the second step of the fragment placement algorithm. Also some small memory leaks occurred when using these filter functions together with pharmacophore-type constraints.
- Fixed some minor errors that could result in small memory leaks in some special cases.
- Improved robustness of fragment space and filter format file parser and added additional consistency checks.

2. Computing

3. Menus and user interface

- Adjusted the amount of output of the build-up algorithm for higher verbosity levels.
- Improved the error handling and modified the output a little when loading in pharmacophore-type constraints.
- New flag `USE_ACCESS_SCALING` for turning on and off access scaling.

4. File formats

- Chempar.dat is splitted in two parts:
 - The new chempar.dat with the general chemistry knowledge (CHEMPAR)
 - flexnovo_settings.dat with tool specified parameters (SETTINGS)
- Changed two identifier in the fragment space file (fsf) for consistency: `@link_terminals => @link_terminal_groups` `@link_connects => @link_compatibility_matrix`
- Added additional warnings and error messages for missing or wrong specifications in the fragment space file

5. Graphics

- Fixed the annotation of match energies when drawing placements for fragments or build-up solutions.

6. Misc.

- New programm parameter 'LOGP_CLASS' is added:
 - `LOGP_CLASS = 1` : Wildmann/Crippen descriptors
 - `LOGP_CLASS = 2` : Ghose/Crippen descriptors

7. Documentation

Version 1.3.1

1. Bug fixes

- Fixed a bug in the creation of the initial fragment queue for the build-up calculation: In case a fragment was skipped because an identical fragment was already inserted in the queue, the remaining fragments were not processed correctly. [96]
- Fixed the handling of a special case in the geometry calculations when connecting fragments. This led to a fatal error before. [107]
- Fixed some output inconsistencies for verbosity levels lower than 4
- Improved robustness of fragment space and filter format file parser
- Fixed the output of the LIGAND INFO command where some special characters accidentally appeared when writing the output into a file
- Fixed a bug where the number of R/S and E/Z stereo centers was swapped in the output of LIGAND/INFO, FRAGSPACE/FINFO and BUILDUP/SOLINFO commands
- Fixed a bug where a ligand was not converted from a SMILES string if the flag 3D_GEN was not set to 1.
- Improved the functionality and consistency of the CHECK function for detecting possible configurational changes if Nitrogens upon fragment connection or link substitution, which cannot be accounted for during the build-up algorithm (see also tutorial in the user guide).
- Fixed a unique bug that occurred in a special case in the build-up algorithm during the calculation enclosing spheres of components. [124,144]
- Fixed a bug in the internal handling of large SMARTS expressions

2. **Computing**

3. **Menus and user interface**

- Added additional warnings and error messages for missing or wrong specifications in the fragment space file

4. **File formats**

5. **Graphics**

6. **Misc.**

7. **Documentation**

Version 1.3.0

1. **Bug fixes**

- Fixed some path handling problems in case the default path has to be used
- Fixed more errors in the handling of filter expressions during the build-up algorithm. These were not checked for molecules in some special cases.
- Fixed a bug in the calculation of molar refractivity values that occurred for a special parametrization, which is different from the one used by default.

2. **Computing**

- There is a new placement filter available (`FILTER_OUT_OF_SITE`), which prevents the algorithm from generating poses where heavy atoms of the fragment are placed outside of a sphere enclosing all active site atoms.
- The number of rotatable bonds is now calculated according to the Boehm scoring function definition, meaning as the number of acyclic, non-terminal bonds between atoms with SP3 hybridisation.
- Filter SMARTS expressions are now checked during the build-up calculation as well. Please keep in mind that this can be very time-consuming, depending on the type and length of the exclusion/inclusion SMARTS lists.

3. Menus and user interface

- The LIGAND menu (known from FlexX) is now included in the program. Reference ligands can now be loaded, visualized and be used, for example, for the definition of protein active sites. The TRANSFORM functions are not available though.
- There are now functions available for drawing fragments and placements. New commands MDRAW in submenus FRAGSPACE, PLACING and BUILDUP. Each of them has its own drawing context so that default values can be set in the corresponding `graphic.dat` static data file or via the menu. New commands SELGRA, SELCOL, SELADM, SELLAB and GRAINF in the submenus FRAGSPACE, PLACING and BUILDUP.
- There are now functions available for listing placement and matches information. New commands LISTALL, LISTPLM, LISTMAT, LISTONE in submenus PLACING and BUILDUP. Each of them prints a table of the corresponding data from an internal database (as known from FlexX). Additionally, custom SQL-like queries can be used via the command QUERY and QHIST.
- A global HELP command is now available for functions, which prints syntax descriptions of all commands available.

4. File formats

5. Graphics

- Default graphic context settings for fragments and placements are now included in the static data file `graphic.dat`. They can be altered in the file directly, or via the corresponding menu commands (see above).

6. Misc.

- FlexNovo now has its own `chempar.dat` file.
- FlexNovo is now based on a fully reworked and improved code library.
- FlexNovo is now available for Linux and Linux 64bit architectures. The 64bit version improves the performance of the program up to 30% but uses significantly (about 30%) more memory at the same time.

7. Documentation

- A complete user guide is now included in the package

- For a detailed description of the new commands available please refer to the user guide and/or use the new internal help command!

Version 1.2.6

1. Bug fixes

- Fixed a segfault when reading a filter definition file without having a fragment space loaded previously.
- Filter SMARTS expressions were not separated correctly in case inclusion and exclusion SMARTS were used at the same time.
- The PLACING/SELECT command had the wrong upper limit when selecting the best scoring # of fragments.
- Fixed an error in the handling of logical expressions for pharmacophore type constraints. Placements fulfilling the constraints were wrongly discarded in some cases. [49]
- Fixed an error that occurred when a pharmacophore was loaded and the build-up algorithm was started without deleting the pharmacophore beforehand. [51]
- Fixed an error that occurred in a special case when loading placement files and resulted in a floating point exception. [50]
- Fixed an error in the handling of filter expressions during the build-up algorithm. These were not checked for the reference fragments before putting them in the final solution list.

2. Computing

3. Menus and user interface

- The FRAGSPACE/FINFO command now prints PSAS instead of SAS.

4. File formats

5. Graphics

6. Misc.

7. Documentation

Version 1.2.5

1. Bug fixes

- Segfault when filtering fragments is now fixed.
- The function FRAGSPACE/READ now checks for redundant entries in the fsf file and handles such cases consistently. This led to a segfault in some cases before.

2. Computing

- The build-up algorithm now stops immediately if no start fragments are selected or no placements are available.

3. Menus and user interface

- New command PLACING/FLIST, which prints a list of fragments with their corresponding initialized score and best placement score, if available.
- The command PLACING/INFO now prints separators between fragments for table format output.

4. File formats

- Changed two filter identifier for consistency reasons: 'psa' is now 'psas', 'ringystems' is now 'rings'.

5. Graphics

6. Misc.

- The CLASH_FACTOR default value changed from 0.6 to 0.7 in order to avoid fragment combinations having strained conformations in some cases.
- All placement geometry filter functions are activated by default.

7. Documentation

- This file now contains a summary of the changes made recently.
- Synchronized standard parameters in the user guide with static data.
- Corrected description of filter functionality.

Version 1.2.2

1. Bug fixes

2. Computing

- A new fragment initialization function is used, which is different with respect to the FlexX initialization. In particular, no TRANSFORM expressions can be used.
- Identical build-up solutions are now identified and filtered using USMILES. Link atoms are identified and treated consistently in the USMILES string.

3. Menus and user interface

- The existence of placement files is now checked before initializing a fragment.

4. File formats

- Fragments and build-up results can now be exported as mol, sln or pdb file. Additionally, coordinate sets can be written to a single or multiple files.
- Updated placement description file format.
- Changed parameters PDF_FORMAT to PDF_COMPRESSION and CFS_PACE to FRAGSPACE for consistency reasons.
- The static data files are now encrypted by default.

5. Graphics

6. Misc.

- FlexNovo now is a stand-alone program.

7. Documentation

- Updated short help files.
- Updated the user guide.

Version 1.2.0

1. Bug fixes

- Removed obsolete warning messages for the logP calculation
- Fixed some directory path managing problems.
- Placement information is now correctly deleted from memory in case only the score info is written to a file.
- Link atoms are now resolved correctly in some special cases.
- SMILES strings for terminal group definition were not used correctly in case of double or triple bonds.
- The PLACING/SELECT command now works correctly.

2. Computing

- Basic placement information can now be written to a file directly during the placement calculation.
- Pharmacophore definitions with logical expressions can now be used

3. Menus and user interface

- New command PLACING/INFO which prints basic placement information
- New command PLACING/INIT which reads fragment scores from a file that are used later as score estimation in the build-up calculation.
- New command FRAGSPACE/CHECK which checks if fragment geometries and the connection rules defined are compatible to each other.
- Adjusted the amount of output the program gives for different verbosity levels.

4. File formats

- The placement description files (pdf) are encoded using the fragment sybyl name only.
- The placement description files can now be written in gzip format as well instead of plain text only before.
- The argument list for a batch script now uses additionally `%{}` instead of `$()` before. This circumvents shell escaping problems for different setups.

5. Graphics

6. Misc.

- FlexNovo is now based on the FlexX Release 2 Kernel.

7. Documentation

C Daten

Das folgende Kapitel enthält die relevanten, im Rahmen dieser Arbeit erstellten und/oder verwendeten Daten und Eingabedateien.

C.1 Shredding-Regelwerke

Die im Rahmen dieser Arbeit entwickelten Shredding-Regelwerke sind in Form der SMARTS-Syntax als Eingabedateien für die Verwendung mit Recore in diesem Kapitel zusammengestellt. Zusätzlich sind auch die Spezifikationen der resultierenden Fragmenträume in Form der Eingabedateien für FlexNovo und Feature Trees enthalten.

C.1.1 Das BRICS-Modell

C.1.1.1 Eingabe-Datei für FlexNovo

```
# Fragment space file for FlexNovo representing the BRICS space
# -----
#
# syntax description:
# @link_types <nof links>
#   <link name 1> <link name 2> ...
#   <link name k> <link name k+1> ... <link name <nof links>>
#
# @fragment_files <nof files>
#   < frag file 1>
#   < frag file 2>
#   :
#   < frag file <nof files>>
#
# @link_terminal_groups
#   <link name 1> <group> [<atom type> <bond type> <bond length> <torsion angle>]
#   <link name 2> <group> [<atom type> <bond type> <bond length> <torsion angle>]
#   :
#   <link name <nof links>> <group> [<atom type> <bond type> <bond length> <torsion angle>]
#
# @link_compatibility_matrix
#   <name of> <name of> [<aatom type> <aatom type>] <bond> <bond> [<torsion>]
#   link 1   link 2   for link 1 for link 2   type length  angle
#   :
#
# ADDITIONAL REMARKS:
# - the <group> entry can either be a file containing the terminal group
#   or a valid <SMILES> string. Please note, that for the latter, a full
#   CORINA license is required (see user guide for further details).
#
# -----

@link_types 16
```

C Daten

```
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16

# R1 C (carbonyl from amides/esters)
# R2 N (amide)
# R3 O (ether/ester)
# R4 C (alkyl from amines/(thio)ethers)
# R5 N (amine)
# R6 C (carbonyl from acylated rings)
# R7 C (olefine)
# R8 C (alkyl from ring substitutions)
# R9 n (aromatic)
# R10 N (lactame)
# R11 S (sulfur from thioethers/sulfoxides)
# R12 S (sulfone from sulfonamides)
# R13 C ('activated' aliphatic ring C, next to N/O/S)
# R14 c ('activated' aromatic ring c, next to N/O/S)
# R15 C (aliphatic ring C, not next to N/O/S)
# R16 c (aromatic ring c, not next to N/O/S)
```

```
@fragment_files 1
  <mol2 file>
```

@link_terminal_groups

#	link	group	aatom	bond	blen	torsion
	R1	[CH3]	*	1	1.507	*
	R2	C(=O)C	*	am	1.337	180
	R3	[CH3]	*	1	1.429	*
	R4	[H]	*	1	1.090	*
	R5	[CH3]	*	1	1.398	*
	R6	[CH3]	*	1	1.507	*
	R7	[CH2]	*	2	1.316	180
	R8	[H]	*	1	1.090	*
	R9	[CH3]	*	1	1.465	*
	R10	[CH3]	*	1	1.465	*
	R11	[CH3]	*	1	1.815	*
	R12	[CH3]	*	1	1.816	*
	R13	[H]	*	1	1.080	*
	R14	[H]	*	1	1.080	*
	R15	[H]	*	1	1.080	*
	R16	[H]	*	1	1.080	*

@link_compatibility_matrix

#	link1	link2	aatom1	aatom2	bond	blen	torsion
	R1	R2	*	*	am	1.355	180
	R1	R3	*	*	1	1.362	*
	R1	R10	*	*	am	1.355	180
	R2	R12	*	*	1	1.656	*
	R2	R14	*	*	1	1.398	*
	R2	R16	*	*	1	1.398	*
	R3	R4	*	*	1	1.452	*
	R3	R13	*	*	1	1.429	*
	R3	R14	*	*	1	1.362	*
	R3	R15	*	*	1	1.429	*
	R3	R16	*	*	1	1.362	*
	R4	R5	*	*	1	1.469	*
	R4	R11	*	*	1	1.815	*
	R5	R13	*	*	1	1.469	*
	R5	R15	*	*	1	1.469	*
	R6	R13	*	*	1	1.507	180
	R6	R14	*	*	1	1.473	180
	R6	R15	*	*	1	1.507	180
	R6	R16	*	*	1	1.473	180

R7	R7	*	*	2	1.316	180
R8	R9	*	*	1	1.465	*
R8	R10	*	*	1	1.465	*
R8	R13	*	*	1	1.507	*
R8	R14	*	*	1	1.473	*
R8	R15	*	*	1	1.507	*
R8	R16	*	*	1	1.473	*
R9	R13	*	*	1	1.465	*
R9	R14	*	*	1	1.355	*
R9	R15	*	*	1	1.465	*
R9	R16	*	*	1	1.355	*
R10	R13	*	*	1	1.465	*
R10	R14	*	*	1	1.355	*
R10	R15	*	*	1	1.465	*
R10	R16	*	*	1	1.355	*
R11	R13	*	*	1	1.816	*
R11	R14	*	*	1	1.762	*
R11	R15	*	*	1	1.816	*
R11	R16	*	*	1	1.762	*
R13	R13	*	*	1	1.530	*
R13	R14	*	*	1	1.507	*
R13	R15	*	*	1	1.530	*
R13	R16	*	*	1	1.507	*
R14	R14	*	*	1	1.473	*
R14	R15	*	*	1	1.507	*
R14	R16	*	*	1	1.473	*
R15	R15	*	*	1	1.530	*
R15	R16	*	*	1	1.507	*
R16	R16	*	*	1	1.473	*

C.1.1.2 Eingabe-Datei für FTree-FS

```
# Fragment space file for FTree-FS representing the BRICS space
# -----
#
# syntax description:
# @link_types <nof links>
#   <link name 1> <link name 2> ...
#   <link name k> <link name k+1> ... <link name <nof links>>
# @link_compatibilities
#   < link name 1> <link name 2> <link name 3> ...
#   :
# Meaning: <link name 1> is compatible with <link name 2>, <link name 3>, ...
# @fragment_files
#   < frag file 1>[<ftree file 1> [<link file 1]]
#   < frag file 2> [<ftree file 2> [<link file 2]]
#   :
# @link_terminals
#   <link name> <atom type> [<bond type> [<formal charge> [<# hydrogen>]]]
#   :
# @link_connects
#   <link name> <atom type> [<bond type>]
# -----

@link_types 16
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16

# R1 C (carbonyl from amides/esters)
# R2 N (amide)
# R3 O (ether/ester)
```

C Daten

```
# R4 C (alkyl from amines/(thio)ethers)
# R5 N (amine)
# R6 C (carbonyl from acylated rings)
# R7 C (olefine)
# R8 C (alkyl from ring substitutions)
# R9 n (aromatic)
# R10 N (lactame)
# R11 S (sulfur from thioethers/sulfoxides)
# R12 S (sulfone from sulfonamides)
# R13 C ('activated' aliphatic ring C, next to N/O/S)
# R14 c ('activated' aromatic ring c, next to N/O/S)
# R15 C (aliphatic ring C, not next to N/O/S)
# R16 c (aromatic ring c, not next to N/O/S)
```

@link_compatibilities

```
R1 R2 R3
R2 R12 R14 R16
R3 R4 R13 R14 R15 R16
R4 R5 R11
R5 R13 R15
R6 R13 R14 R15 R16
R7 R7
R8 R9 R10 R13 R14 R15 R16
R9 R13 R14 R15 R16
R10 R13 R14 R15 R16
R11 R13 R14 R15 R16
R13 R13 R14 R15 R16
R14 R15 R15 R16
R15 R15 R16
R16 R16
```

@fragment_files 1

```
<mol2 file>
```

@link_terminals

#	link	atom	aatom	bond	blen	fcharge	#H
R1	0.2	*		1	1.362	-1.0	
R2	C.3	*		1	1.337	0.0	3
R3	C.3	*		1	1.429	0.0	3
R4	H						
R5	C.3	*		1	1.398	0.0	3
R6	C.3	*		1	1.507	0.0	3
R7	C.2	*		2	1.316	0.0	2
R8	H						
R9	C.3	*		1	1.465	0.0	3
R10	C.3	*		1	1.465	0.0	3
R11	C.3	*		1	1.815	0.0	3
R12	C.3	*		1	1.816	0.0	3
R13	H						
R14	H						
R15	H						
R16	H						

@link_connects

#	link	atom	aatom	bond	blen
R1	N.pl3	*	am		1.355
R2	C.2	*		1	1.656
R3	C.3	*		1	1.452
R4	N.3	*		1	1.469
R5	C.3	*		1	1.398
R6	C.2	*		1	1.507
R7	C.2	*		2	1.316

R8	C.2	*	1	1.465
R9	C.3	*	1	1.465
R10	C.3	*	1	1.465
R11	C.3	*	1	1.815
R12	N.2	*	1	1.816
R13	C.2	*	1	1.530
R14	C.2	*	1	1.473
R15	C.2	*	1	1.507
R16	C.2	*	1	1.473

C.1.1.3 SMARTS-Spezifikation und Filter für Recore

```
# Fragment filter
@filter
fragmentsize 1-16
cutdistance 0-9
cuts 1-9
terminalsize 1-15

# Exclusion smarts
exclude any smarts [!#1&!#6&!#7&!#8&!#9&!#15&!#16&!#17&!#35&!#53]
exclude any smarts [r9,r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20,r21,r22,r23,r24,r25]
exclude any smarts [C;!R]~[C;!R]~[C;!R]~[C;!R]~[C;!R]~[C;!R]
exclude any smarts S.S.S
exclude any smarts [?]
```



```

# Amide, Sulfonamide
N1 [C;!D4;!D1;!R;$$(C(=O));$(C([N;!D1])[!S;!D1][!D1])]-;!@[N;!D1;!$(N(C(=O)))[!S;!D1][!D1]]
N3 [S;!X4;$$(S(=O))]-;!@[#7;X3;!D1;$$(#7)(S(=O))][#6;!D1][!D1];!$(#7)(C(=O))
N6 [C;!D4;!D1;!R;$$(C(=O));$(C([n])[!S;!D1][!D1])]-;!@[n;!$(n@[#6](=O))]]
# prim./sec. Amine
N2a [C;!D4;!D1;!R;$$(C(=O));$(C([N;X3;!D1])[!#6;!D1][!D1])]-;!@[N;X3;!D1;!$(NC=;!@[N;0,S])];$(N([C;!D1;!R])[#6;!D1])!=;![!D1]]
N2b [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[N;X3;!D1;!$(NC=;!@[N,0,S])];$(N([#6;!D1;!R])[#6;!D1])!=;![!D1]]
N2c [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[N;X3;!D1;!$(NC=;!@[N,0,S])];$(N([#6;!D1;!R])[#6;!D1])!=;![!D1]]
N2d [c;!X3;!R;$$(C([n,n,S,s,0,ol]))]-;!@[N;X3;!D1;!$(NC=;!@[N,0,S])];$(N([#6;!D1;!R])[#6;!D1])!=;![!D1]]
# Pyrrrol-Stickstoff
N4 [C;!D4;!D1;!R;$$(C=[C,0,S,N]);!$(C([O,S,N;!R]);$(C([n])!=;![#6;!D1])~[!D1])]-;!@[n;!$(n@[#6](=O))]]
N5a [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[n;!$(n@[#6](=O))]]
N5b [c;!X3;!R;$$(c)[N,n,S,s,0,ol]]-;!@[n;!$(n@[#6](=O))]]
N5c [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[n;!$(n@[#6](=O))]]
N5d [c;!X3;!R;$$(c)[N,n,S,s,0,ol]]-;!@[n;!$(n@[#6](=O))]]
# Lactam-Stickstoff
N7 [C;!D4;!D1;!R;$$(C=[C,0,S,N]);!$(C([O,S,N;!R]);$(C([#7;R;D3])!=;![#6;!D1])~[!D1])]-;!@[#7;R;D3;$$(#7)[#6](=O))]
N8a [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[#7;R;D3;$$(#7)[#6](=O))]
N8b [c;!X3;!R;$$(c)[N,n,S,s,0,ol]]-;!@[#7;R;D3;$$(#7)[#6](=O))]
N8c [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[#7;R;D3;$$(#7)[#6](=O))]
N8d [c;!X3;!R;$$(c)[N,n,S,s,0,ol]]-;!@[#7;R;D3;$$(#7)[#6](=O))]
N9 [C;!D4;!D1;!R;$$(C(=O));$(C([#7;R;D3])[!S;!D1][!D1])]-;!@[#7;R;D3;$$(#7)[#6](=O))]
# Thioether (keine Thio-Lactone oder Thio-Acetale)
S1 [C;!D4;!D1;!R;$$(C(=S));$(C([S;D2])[#6;!D1][!D1])]-;!@[S;D2;$$(S([C;!D1;!R])[#6;!D1])];$(SCS);!$(SC=*)];$(S[P,S])]]
S1a [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[S;D2;$$(S([#6;!D1;R])[#6;!D1])];$(SCS);!$(SC=;!@*);!$(S[P,S])]]
S1b [c;!X3;!R;$$(c)[N,n,S,s,0,ol]]-;!@[S;D2;$$(S([#6;!D1;R])[#6;!D1])];$(SCS);!$(SC=;!@*);!$(S[P,S])]]
S1c [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[S;D2;$$(S([#6;!D1;R])[#6;!D1])];$(SCS);!$(SC=;!@*);!$(S[P,S])]]
S1d [c;!X3;!R;$$(c)[N,n,S,s,0,ol]]-;!@[S;D2;$$(S([#6;!D1;R])[#6;!D1])];$(SCS);!$(SC=;!@*);!$(S[P,S])]]
# Sulfoxide, Sulfone (keine Sulfonamide)
S2 [C;!D4;!D1;!R;$$(C(S=O)[#6;!D1][!D1])]-;!@[S;$$(S(=O)([C;!D1;!R])[#6;!D1])]]
S2a [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[S;$$(S(=O)([#6;!D1;R])[#6;!D1])]]
S2b [c;!X3;!R;$$(c)[N,n,S,s,0,ol]]-;!@[S;$$(S(=O)([#6;!D1;R])[#6;!D1])]]
S2c [C;!D3;!R;$$(C([N,n,S,s,0,ol]))]-;!@[S;$$(S(=O)([#6;!D1;R])[#6;!D1])]]
S2d [c;!X3;!R;$$(c)[N,n,S,s,0,ol]]-;!@[S;$$(S(=O)([#6;!D1;R])[#6;!D1])]]

```

BRICS-Shredding-Regelwerk (Teil 2)

C.1.2 Die RECAP-Reimplementierung

C.1.2.1 Eingabe-Datei für FlexNovo

```
# Fragment space file for FlexNovo representing the RECAP space
# -----
#
# syntax description:
# @link_types <nof links>
#   <link name 1> <link name 2> ...
#   <link name k> <link name k+1> ... <link name <nof links>>
#
# @fragment_files <nof files>
#   < frag file 1>
#   < frag file 2>
#   :
#   < frag file <nof files>>
#
# @link_terminal_groups
#   <link name 1> <group> [<atom type> <bond type> <bond length> <torsion angle>]
#   <link name 2> <group> [<atom type> <bond type> <bond length> <torsion angle>]
#   :
#   <link name <nof links>> <group> [<atom type> <bond type> <bond length> <torsion angle>]
#
# @link_compatibility_matrix
#   <name of> <name of> [<aatom type> <aatom type>] <bond> <bond> [<torsion>]
#   link 1   link 2   for link 1   for link 2   type length   angle
#   :
#
# ADDITIONAL REMARKS:
# - the <group> entry can either be a file containing the terminal group
#   or a valid <SMILES> string. Please note, that for the latter, a full
#   CORINA license is required (see user guide for further details).
# -----

@link_types 20
  R11 R12 R21 R22 R31 R32 R41 R42 R51 R52 R60 R71 R72 R81 R82 R91 R92 R10 R1 R2

# R11 Amide C
# R12 Amide N
# R21 Ester C
# R22 Ester O
# R31 Amine C
# R32 Amine N
# R41 Urea N
# R42 Urea C
# R51 Ether C
# R52 Ether O
# R60 Olefin
# R71 Quarternary nitrogen - aliphatic carbon N
# R72 Quarternary nitrogen - aliphatic carbon C
# R81 Aromatic nitrogen - aliphatic carbon C
# R82 Aromatic nitrogen - aliphatic carbon n
# R91 Lactam nitrogen - aliphatic carbon C
# R92 Lactam nitrogen - aliphatic carbon N
# R10 Aromatic carbon - aromatic carbon
# R1  Sulphonamide S
# R2  Sulphonamide N

@fragment_files 1
  <mol2 file>
```

```
@link_terminal_groups
# link group aatom bond blen torsion
R11 [CH3] * 1 1.507 *
R12 C(=O)C * am 1.337 180
R21 [CH3] * 1 1.507 *
R22 [CH3] * 1 1.429 *
R31 [H] * 1 1.090 *
R32 [CH3] * 1 1.398 *
R41 C(=O)C * am 1.337 180
R42 [CH3] * 1 1.507 *
R51 [H] * 1 1.090 *
R52 [CH3] * 1 1.429 *
R60 [CH2] * 2 1.316 180
R71 [CH3] * 1 1.507 *
R72 [CH3] * 1 1.507 *
R81 [H] * 1 1.090 *
R82 [CH3] * 1 1.465 *
R91 [H] * 1 1.090 *
R92 [CH3] * 1 1.465 *
R10 [H] * 1 1.080 *
R1 [CH3] * 1 1.816 *
R2 C(=O)C * am 1.337 180
```

```
@link_compatibility_matrix
# link1 link2 aatom1 aatom2 bond blen torsion
R11 R12 * * am 1.355 180
R21 R22 * * 1
R31 R32 * * 1
R41 R42 * * am 1.355 180
R51 R52 * * 1
R60 R60 * * 2 1.316 180
R71 R72 * * 1
R81 R82 * * 1
R91 R92 * * 1
R10 R10 * * 1
R1 R2 * * 1 1.656 *
```

C.1.2.2 Eingabe-Datei für FTree-FS

```
# Fragment space file for FTree-FS representing the RECAP space
# -----
#
# syntax description:
# @link_types <nof links>
# <link name 1> <link name 2> ...
# <link name k> <link name k+1> ... <link name <nof links>>
# @link_compatibilities
# <link name 1> <link name 2> <link name 3> ...
# :
# Meaning: <link name 1> is compatible with <link name 2>, <link name 3>, ...
# @fragment_files
# <frag file 1>[<ftree file 1> [<link file 1>]]
# <frag file 2> [<ftree file 2> [<link file 2>]]
# :
# @link_terminals
# <link name> <atom type> [<bond type> [<formal charge> [<# hydrogen>]]]
# :
# @link_connects
# <link name> <atom type> [<bond type>]
```

C Daten

```
# -----  
  
@link_types 20  
  R11 R12 R21 R22 R31 R32 R41 R42 R51 R52 R60 R71 R72 R81 R82 R91 R92 R10 R1 R2  
  
# R11 Amide C  
# R12 Amide N  
# R21 Ester C  
# R22 Ester O  
# R31 Amine C  
# R32 Amine N  
# R41 Urea N  
# R42 Urea C  
# R51 Ether C  
# R52 Ether O  
# R60 Olefin  
# R71 Quarternary nitrogen - aliphatic carbon N  
# R72 Quarternary nitrogen - aliphatic carbon C  
# R81 Aromatic nitrogen - aliphatic carbon C  
# R82 Aromatic nitrogen - aliphatic carbon n  
# R91 Lactam nitrogen - aliphatic carbon C  
# R92 Lactam nitrogen - aliphatic carbon N  
# R10 Aromatic carbon - aromatic carbon  
# R1 Sulphonamide S  
# R2 Sulphonamide N  
  
@link_compatibilities  
  R11 R12  
  R21 R22  
  R31 R32  
  R41 R42  
  R51 R52  
  R60 R60  
  R71 R72  
  R81 R82  
  R91 R92  
  R10 R10  
  R1 R2  
  
@fragment_files 1  
  <mol2 file>  
  
@link_terminals  
# link  atom  aatom  bond  blen  fcharge  #H  
  R11  0.2  *      1     1.362  -1.0     3  
  R12  C.3  *      1     1.337   0.0     3  
  R21  0.2  *      1     1.362  -1.0     3  
  R22  C.3  *      1     1.429   0.0     3  
  R31  H  
  R32  C.3  *      1     1.398   0.0     3  
  R41  C.3  *      1     1.337   0.0     3  
  R42  C.3  *      1     1.337   0.0     3  
  R51  H  
  R52  C.3  *      1     1.429   0.0     3  
  R60  C.2  *      2     1.316   0.0     2  
  R71  C.3  *      1     1.507   0.0     3  
  R72  H  
  R81  H  
  R82  C.3  *      1     1.337   0.0     3  
  R91  H  
  R92  C.3  *      1     1.337   0.0     3  
  R10  H
```

```
R1 C.3 * 1 1.815 0.0 3
R2 C.3 * 1 1.337 0.0 3
```

@link_connects

```
# link atom aatom bond blen
R11 N.pl3 * am 1.355
R12 C.2 * am 1.355
R21 C.2 * 1 1.656
R22 O.2 * 1 1.656
R31 C.3 * 1 1.452
R32 N.3 * 1 1.452
R41 N.3 * am 1.469
R42 C.2 * am 1.469
R51 C.3 * 1 1.398
R52 O.3 * 1 1.398
R60 C.2 * 2 1.316
R71 C.2 * 1 1.507
R72 N.4 * 1 1.507
R81 C.2 * 1 1.465
R82 N.2 * 1 1.465
R91 C.3 * 1 1.465
R92 N.2 * 1 1.465
R10 C.3 * 1 1.465
R1 S.3 * am 1.815
R2 N.2 * am 1.815
```

C.1.2.3 SMARTS-Spezifikation und Filter für Recore

Fragment filter

@filter

fragmentsize 1-16

cutdistance 0-9

cuts 1-9

terminalsize 1-15

Exclusion smarts

exclude any smarts [!#1&!#6&!#7&!#8&!#9&!#15&!#16&!#17&!#35&!#53]

exclude any smarts [r9,r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20,r21,r22,r23,r24,r25]

exclude any smarts [C;!R]~[C;!R]~[C;!R]~[C;!R]~[C;!R]~[C;!R]

exclude any smarts S.S.S

exclude any smarts [?]

```

# Amide
N1a [C;!D4;!R;$$(C(=O));$(C[N;!D1])[N;!D1][!D1]]-;!@[N;!D1];!$(N(C(=O)(C=O));$(N(C(=O))[N;!D1][!D1]])
N1b [C;!D4;!R;$$(C(=O));$(C[N;!D1])[N;!D1][!D1]]-;!@[N;!D1];!$(N(C(=O)(C=O));$(N(C(=O))[N;!D1][!D1]])
# Ester, Urethane (keine Anhydride oder Michael-Akzeptoren)
O2 [C;!D4;!R;$$(C(=O));$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O(C(=O)))[#6;!D1][!D1];!$(O[P,S])
# Amine
N3a [C;!D4;!R;$$(C(=O));$(C[N;X3;!D1])[#6;!D1][!D1]]-;!@[N;X3;!D1];$(N(C;[O;D1;R])[#6;!D1][!D1])
N3b [C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X3;!D1];$(N(C;[O;D1;R])[#6;!D1][!D1])
N3c [c;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X3;!D1];$(N(C;[O;D1;R])[#6;!D1][!D1])
N3d [C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X3;!D1];$(N(C;[O;D1;R])[#6;!D1][!D1])
N3e [c;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X3;!D1];$(N(C;[O;D1;R])[#6;!D1][!D1])
# Harnstoff
N4 [N;!D1];!$(N(C(=O));$(N(C(=O))[!D1][!D1]]-;!@[C;!D4;!R;$$(C(=O));$(C([N;!D1])[N;!D1][!D1]))
# Ether (keine Lactone, Acetale oder Michael-Akzeptoren)
O5a [C;!D4;!R;$$(C(=O));$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
O5b [C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
O5c [c;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
O5d [C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
O5e [c;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
# Olefine (ohne Michael-Akzeptoren)
C6 [#6;!D1];!$([#6]=;!@[C,O,S;!@]);$([#6][!H;!D1][!H;!D1])=;!@[!H;!D1][!H;!D1]]-;!@[!H;!D1][!H;!D1]]-;!@[!H;!D1][!H;!D1]]
# Quarternäre Stickstoffe
N7a [N;X4]-;!@[C;!D4;!R;$$(C(=O));$(C([N;X4;!D1])[#6;!D1][!D1])
N7b [N;X4]-;!@[C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X4;!D1];!$(N(C;[O;D1;R])[#6;!D1][!D1])
N7c [N;X4]-;!@[C;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X4;!D1];!$(N(C;[O;D1;R])[#6;!D1][!D1])
N7d [N;X4]-;!@[C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X4;!D1];!$(N(C;[O;D1;R])[#6;!D1][!D1])
N7e [N;X4]-;!@[C;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X4;!D1];!$(N(C;[O;D1;R])[#6;!D1][!D1])
# Aromatischer Stickstoff - Aliphatischer Kohlenstoff
N8a [C;!D4;!R;$$(C(=O));$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
N8b [C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
N8c [C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
# Lactam-Stickstoff - Aliphatischer Kohlenstoff
N9a [C;!D4;!R;$$(C(=O));$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
N9b [C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
N9c [C;D3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
N9d [C;!D4;!R;$$(C(=O));$(C[O;D2])[#6;!D1][!D1]]-;!@[O;D2];$(O([C;!D1;R])[#6;!D1][!D1]);!$(O(C=*))!$(O[P,S])
# Aromatischer Kohlenstoff - aromatischer Kohlenstoff
C1a [c;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X3;!D1];!$(N(C;[O;D1;R])[#6;!D1][!D1])
C1b [c;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X3;!D1];!$(N(C;[O;D1;R])[#6;!D1][!D1])
C1c [c;X3;R;$$(C[O;D2])[#6;!D1][!D1]]-;!@[N;X3;!D1];!$(N(C;[O;D1;R])[#6;!D1][!D1])
# Sulfonamide
N11 [S;X4;$$(S(=O))]-;!@[#7;X3;!D1];!$([#7](S(=O)))[#6;!D1][!D1];!$([#7](S(=O)))

```

C.2 FlexNovo-Berechnungen

C.2.1 Filter-SMARTS für reaktive chemische Gruppen

Die im Rahmen dieser Arbeit von Baurin et al. adaptierten Spezifikationen für reaktive chemische Substrukturen [Baurin *et al.*, 2004a,b] sind hier in Form der Eingabedateien für FlexNovo zusammengestellt.

```
# Filter description file for FlexNovo
# -----
# (c) Joerg Degen, ZBH Center for Bioinformatics, University of Hamburg
#
# syntax description:
# @filters
# @tolerance <number of tolerated violations>
#
# valid identifiers are: mol_weight, rings, rotatable_bonds,
#                       h_acceptors, h_donors, stereo_centers,
#                       clogp, mol_refrac, psas
# [@identifier <lower limit> <upper limit>]
# :
#
#                               and: excl_smarts, incl_smarts
# [@identifier]
# smarts EXPRESSION
# :
#
# -----
# SMARTS adapted from description in:
# (1) Baurin et al., JCIICS 2004, (44), 643-651
# (2) Baurin et al., JCIICS 2004, (44), 2157-2166

@filters
@excl_smarts
# Filter 1 (1)
# -----
smarts [!#1&!#6&!#7&!#8&!#9&!#16&!#17]           # only H,C,N,O,F,S,Cl
smarts [CD2H1]=[OD1H0]                             # CHO
smarts [CD3H0](=[OD1H0])-[OD2H0]-[CD3H0]=[OD1H0] # Anhydrides
smarts [CD3H0](=[OD1H0])(-[Cl,Br,F])              # COHal
smarts [OD2H0]-[CD3H0](=[OD1H0])(-[Cl,Br,F])      # OCOHal
smarts A-[Cl,Br,I]                                  # Alkyl-Hal
smarts [SD1H1]                                       # SH
smarts [ND2H0]=[CD2H0]=[OD1H0,SD1H0]              # NCO, NCS
smarts [ND2H0]#[CD1H0]                              # isonitrils

smarts [SD4H0](=[OD1H0])(=[OD1H0])[Cl,Br,F]        # SO2-Hal
smarts C=[OD2H0,SD2H0]-[SD1H0,OD1H0]              # C=O-S, C=S-O
smarts C=[SD1H0]                                     # C=S
smarts [NH,O,S;r3]                                   # Epoxide, Aziridine
smarts [SD2H0,OD2H0]-[SD2H0,OD2H0]                # S-S, O-O
smarts [ND2H0][OD1H0]                              # N=O

smarts [SD2H0]-[Cl,Br,F]                            # S-Hal
smarts [NX3]-[Cl,Br,F,I]                            # N-Hal
smarts [SD4H0](=[OD1H0])(=[OD1H0])(-[OD2H0]-C)(-[OD2H0]-C) # disub. sulfates
smarts [SD4H0](=[OD1H0])(=[OD1H0])(-[OD2H0]-C)-C   # disub. sulfonates
smarts [ND2H0]=[CD2H0]=[ND2H0]                    # Carbodimides
smarts [CD2H2]-[CD3H0]=[ND1H1]                     # imines 1
```

C Daten

```
smarts [CD2H1]=[CD3HO]-[ND1H2] # imines 2
smarts [ND2HO]=[ND2HO] # azo-type
smarts [ND2HO]=[ND1HO] # diazo
smarts [CD3HO](-[NX3])(-[NX3])=[SD1HO] # thiourea 1
smarts [CD3HO](-[NX3])(=[NX2])-[SD1H1] # thiourea 1
smarts [NX3]-[OX2] # hydroxylamines
smarts [+] # positively charged

smarts [NX4] # quarternary N
smarts [ND3HO](~[OD1HO])(~[OD1HO])-A # alkyl N-oxides
smarts [ND3HO](~[OD1HO])(~[OD1HO])-a # aromatic nitro
smarts [CD1H2,CD2H1,CD3HO;!$(C=,*)] # C radical, charged
smarts [SD4HO](=[OD1HO])(=[OD1HO])(-a)-a # polyaryl sulfonyl
# bis-guanidinium
smarts [CD3HO]([ND1H2])([ND1H2])([ND2H1]).[CD3HO]([ND1H2])([ND1H2])([ND2H1])
smarts [OD1H1]aaaa[OD1H1] # hydrochinone
smarts [OD1H1]aa[OD1H1] # brencatechine
smarts [OD1HO]=[CD3HO,r6]-[CD3H1,r6]=[CD3H1,r6]-[CD3HO,r6]=[OD1HO]
smarts [OD1HO]=[CD3HO,r6]-[CD3H1,r6]=[OD1HO] # p,o-benzochinone

# condensed polycyclic nonlinear aromatic systems
smarts [CD3HO;r5,r6;R2](:[CD3HO;r5,r6;R2])(:[CD3HO;r5,r6;R2])
smarts [CD3HO;r6;R2][*;r5,r6;R1][CD3HO;r6;R2] # anthracene-like
smarts [CD3H1;r6;R2,R3]12-aa-[CD3H1;r6;R2,R3](-aa1)aa2 # triptycene-like
smarts a1aaaa2aaaa(a12)a1aaaa2aaaaa12 # binaphthyl
smarts [CD3HO]=[CD2HO]=[OD1HO] # ketenes
smarts [ND1H2]aaaa[ND1H2] # 1,4-dianiline
smarts [ND2H1,ND3HO](a1aaa2aaaaa2a1)a1aaaaa1 # b-naphthylaniline

# Filter 2 (2)
# -----
# a-naphthylalanines
smarts [CD3H1]([CD3HO](~[OD1])(~[OD1]))([ND1H2])[CD2H2](a1aaaa2aaaaa12)
smarts [NX3][CX4][NX3] # aminals
smarts [OD2HO][CX4][OD2HO] # acetals
smarts [ND1H2]a # aromatic amine
# barbiturates
smarts [CD3HO]1(=[OD1HO])[NX3][CD3HO](=[OD1HO])[NX3][CD3HO](=[OD1HO])[CX4]1
smarts [CD3HO,R](=[SD1HO])([ND2H1,R])([ND2H1,R]) # cyclic thioureas
smarts [CD2H1]=[ND2HO][ND2H1] # hydrazones
smarts [CD2H1]=[ND2HO][OX2] # oxime (ethers)
smarts [nD3HO,R](~[OD1HO])(a)a # pyridine N-oxides

# additional
# -----
smarts C~C~C~C # more than 4 C
smarts S.S.S # more than 3 S
smarts [CX3]=[CD2H1]-[CD2H1]=[O,N,S] # a,b unsaturated
smarts [CD1H2]=[CD2H1]-[OD2HO,SD2HO]-C # a,b unsat. alkoxy
smarts [O,N]!-CA=A # michael-acceptors
smarts [O,N]!-CA#A # michael-acceptors
smarts C(=*)(=*) # allene type
smarts [c,n;R][O;!R][C;!R](=O)[#6,#7,#8;!R]
```

C.2.2 Pharmakophor-Spezifikationen

Die im Rahmen dieser Arbeit verwendeten Spezifikationen für Pharmakophor-Bedingungen sind hier in Form der Eingabedateien für FlexNovo zusammengestellt.

C.2.2.1 Eingabedatei für 1di8

```
# Anchor-Pharmacophore constraints for 4dfr
@interact
essential _N LEU A 83 h_don
optional _O LEU A 83 h_acc
optional _O GLU A 81 h_acc
@partial 1 2
```

C.2.2.2 Eingabedatei für 1err

```
# Anchor-Pharmacophore constraints for 1err
@interact
essential _NH2 ARG A 394 h_don
essential _OE1 GLU A 353 h_acc 1
```

C.2.2.3 Eingabedateien für 1k22

```
# Anchor-Pharmacophore constraints for 1k22
@interact
essential _OD1 ASP H 189 h_acc 1
essential _OD2 ASP H 189 h_acc 1

# Pharmacophore constraints for 1k22
@interact
essential _OD1 ASP H 189 h_acc 1
essential _OD2 ASP H 189 h_acc 1
essential _N GLY H 219 h_don
@spatial
essential 19.700 -12.290 19.810 2.000 smarts [#6;R]
essential 20.030 -13.590 24.390 2.000 smarts [#6;R]
```

C.2.2.4 Eingabedatei für 4dfr

```
# Anchor-Pharmacophore constraints for 4dfr
@interact
optional _OD1 ASP A 27 h_acc 1
optional _OD2 ASP A 27 h_acc 1
@partial 1 2

# Pharmacophore constraints for 4dfr
@interact
essential _OD1 ASP A 27 h_acc 1
essential _OD2 ASP A 27 h_acc 1
essential _O ILE A 5 h_acc
essential _CG PHE A 31 phenyl_center 1
essential NH2 ARG A 57 h_don
```

C.2.2.5 Eingabedatei für 6cox

```
# Anchor-Pharmacophore constraints for 6cox
@spatial
essential 26.34 22.84 49.64 1.5 SMARTS [#6,#7,#8,#16;r4,r5,r6]
```

C.3 Enumeration der target-spezifischen Fragmenträume

In der folgenden Tabelle sind die Wertebereiche der physikochemischen Eigenschaften zusammengestellt, die für die Enumeration der target-spezifischen Fragmenträume verwendet wurden und aus den DUD Inhibitor-Zusammenstellungen abgeleitet wurden.

Tabelle C.1: Bereiche physikochemischer Eigenschaften der Inhibitor-Sets aus DUD

Target Klasse	mol. Gewicht	rot. B.	Ringe	H-Akz.	H-Don.	c logP
ACE	176.2 - 516.3	3 - 12	0 - 3	4 - 9	0 - 4	-0.2 - 4.2
AChE	166.2 - 482.6	0 - 21	0 - 5	2 - 8	0 - 6	1.1 - 6.0
ALR2	225.2 - 449.2	0 - 8	1 - 4	2 - 7	0 - 4	-1.2 - 6.2
AR	256.4 - 556.5	1 - 9	1 - 5	1 - 8	0 - 3	2.0 - 10.0
CDK2	235.2 - 483.6	0 - 14	2 - 8	4 - 10	1 - 7	-1.1 - 6.2
COX-1	137.1 - 371.4	1 - 11	1 - 3	2 - 6	0 - 4	1.4 - 5.6
COX-2	243.3 - 571.1	0 - 12	2 - 6	0 - 8	0 - 7	1.7 - 9.2
DHFR	206.3 - 521.4	0 - 11	2 - 5	4 - 13	4 - 8	-0.4 - 5.7
EGFr	221.3 - 597.5	0 - 12	2 - 5	2 - 10	0 - 8	-2.6 - 7.3
ER _{Agonisten}	212.2 - 356.4	1 - 11	2 - 5	1 - 6	0 - 4	2.0 - 7.2
ER _{Antagonisten}	347.5 - 508.7	5 - 12	3 - 6	2 - 5	1 - 3	5.1 - 8.8
FGFr1	281.3 - 597.5	0 - 14	3 - 5	4 - 10	1 - 5	0.7 - 7.5
FXa	293.4 - 574.4	1 - 13	2 - 7	4 - 12	1 - 12	-1.9 - 6.7
GART	429.4 - 479.5	7 - 11	2 - 3	11 - 14	4 - 7	-2.7 - 3.6
GPB	180.2 - 424.9	4 - 11	1 - 4	4 - 11	0 - 8	-6.2 - 4.3
GR	271.4 - 460.6	3 - 11	2 - 5	2 - 6	0 - 3	1.1 - 8.8
HIVPR	362.5 - 631.8	5 - 16	2 - 7	3 - 10	0 - 7	2.0 - 8.2
HIVRT	244.3 - 584.3	1 - 13	1 - 5	2 - 9	0 - 4	1.3 - 8.6
HMGA	256.3 - 557.6	3 - 20	2 - 4	3 - 9	0 - 3	2.2 - 6.7
HSP90	296.3 - 433.9	3 - 13	3 - 4	5 - 8	2 - 7	1.7 - 5.0
InhA	197.2 - 646.3	0 - 12	1 - 7	2 - 9	0 - 4	0.8 - 6.9
NA	193.2 - 408.5	1 - 14	1 - 3	5 - 11	1 - 9	-7.2 - 2.4
P38	239.2 - 509.6	0 - 10	2 - 5	2 - 8	0 - 5	0.3 - 7.8
PARP	136.2 - 366.5	0 - 4	1 - 5	3 - 5	1 - 4	0.3 - 6.2
PDE5	271.3 - 620.7	1 - 14	2 - 7	5 - 12	0 - 4	1.4 - 7.4
PDGFr _b	210.2 - 597.5	0 - 12	3 - 5	1 - 10	0 - 5	0.2 - 7.5
PPAR _g	264.3 - 640.7	2 - 15	2 - 6	3 - 10	0 - 3	1.0 - 10.4
SAHH	233.2 - 321.4	3 - 7	2 - 3	6 - 9	4 - 6	-3.8 - -0.9
SRC	226.2 - 597.5	0 - 13	3 - 5	4 - 10	0 - 5	1.6 - 7.5
Thrombin	250.3 - 647.8	2 - 17	2 - 6	3 - 12	1 - 9	-0.4 - 5.0
TK	184.2 - 405.2	3 - 7	1 - 3	5 - 9	2 - 5	-2.2 - 2.0
Trypsin	124.2 - 620.9	3 - 17	1 - 6	2 - 11	5 - 15	-0.3 - 4.7
VEGFr ₂	238.3 - 692.7	0 - 14	2 - 7	2 - 14	0 - 6	1.6 - 8.0
Alle Klassen	122.1 - 692.7	0 - 21	0 - 8	0 - 14	0 - 15	-7.2 - 10.4

Literaturverzeichnis

- Agrafiotis, D.K., Gibbs, A. C., Zhu, F., Izrailev, S. & Martin, E. (2007). Conformational sampling of bioactive molecules: a comparative study. *J Chem Inf Model*, **47** (3), 1067–1086.
- Ali, M.A., Bhogal, N., Findlay, J.B. & Fishwick, C.W. (2005). The first de novo-designed antagonists of the human NK(2) receptor. *J Med Chem*, **48** (18), 5655–5658.
- Allen, F.H. (2002). The Cambridge Structural Database: a quarter of a million crystal structures and rising. *Acta Cryst B*, **58**, 380–388.
- Allu, T. K. & Oprea, T. I. (2005). Rapid evaluation of synthetic and molecular complexity for in silico chemistry. *J Chem Inf Model*, **45** (5), 1237–1243.
- Andrews, K.M & Cramer, R.D. (2000). Toward General Methods of Target Library Design: A Topomer Shape Similarity Searching with Diverse Structures as Queries. *J Med Chem*, **43**, 1723–1740.
- Aronov, A.M. & Bemis, G.W. (2004). A Minimalist Approach to Fragment-Based Ligand Design Using Common Rings and Linkers: Application to Kinase Inhibitors. *Proteins: Structure, Function and Bioinformatics*, **57**, 36–50.
- Baber, J.C. & Feher, M. (2004). Predicting synthetic accessibility: application in drug discovery and development. *Mini Rev Med Chem*, **4** (6), 681–692.
- Barakat, M.T. & Dean, P.M. (1995a). The atom assignment problem in automated de novo drug design. 1. Transferability of molecular fragment properties. *J Comput Aided Mol Des*, **9** (4), 341–350.
- Barakat, M.T. & Dean, P.M. (1995b). The atom assignment problem in automated de novo drug design. 2. A method for molecular graph and fragment perception. *J Comput Aided Mol Des*, **9** (4), 351–358.
- Barakat, M.T. & Dean, P.M. (1995c). The atom assignment problem in automated de novo drug design. 3. Algorithms for optimization of fragment placement onto 3D molecular graphs. *J Comput Aided Mol Des*, **9** (4), 359–372.
- Barakat, M.T. & Dean, P.M. (1995d). The atom assignment problem in automated de novo drug design. 4. Tests for site-directed fragment placement based on molecular complementarity. *J Comput Aided Mol Des*, **9** (5), 448–456.

- Barakat, M.T. & Dean, P.M. (1995e). The atom assignment problem in automated de novo drug design. 5. Tests for envelope-directed fragment placement based on molecular similarity. *J Comput Aided Mol Des*, **9** (5), 457–462.
- Bartlett, P.A., Shea, G.T., Telfer, S.J. & Waterman, S. (1989). CAVEAT: A Program to Facilitate the Structure-Derived Design of Biologically Active Molecules. In: *Molecular Recognition: Chemical and Biological Problems*, (Roberts, S.M., Hrsg.), S. 182–196. Royal Society of Chemistry.
- Baurin, N., Aboul-Ela, F., Barril, X., Davis, B., Drysdale, M., Dymock, B., Finch, H., Fromont, C., Richardson, C., Simmonite, H. & Hubbard, R.E. (2004a). Design and Characterization of Libraries of Molecular Fragments for Use in NMR Screening against Protein Targets. *J Chem Inf Comput Sci*, **44**, 2157–2166.
- Baurin, N., Baker, R., Richardson, C., Chen, I., Foloppe, N., Potter, A., Jordan, A., Roughley, S., Parrat, M., Greaney, P., Morley, D. & Hubbard, R.E. (2004b). Drug-like Annotation and Duplicate Analysis of a 23-Supplier Chemical Database Totalling 2.7 Million Compounds. *J Chem Inf Comput Sci*, **44**, 643–651.
- Baxendale, I.R., Hayward, J.J., Ley, S.V. & Tranmer, G.K. (2007). Pharmaceutical Strategy and Innovation: An Academics Perspective. *ChemMedChem*, **2** (6), 768–788.
- Belda, I., Madurga, S., Llorca, X., Martinell, M., Tarrago, T., Piqueras, M. G., Nicolas, E. & Giralt, E. (2005). ENPDA: an evolutionary structure-based de novo peptide design algorithm. *J Comput Aided Mol Des*, **19** (8), 585–601.
- Bemis, G.W. & Murcko, M.A. (1996). The Properties of Known Drugs, 1. Molecular Frameworks. *J Med Chem*, **39**, 2887–2893.
- Bemis, G.W. & Murcko, M.A. (1999). Properties of Known Drugs. 2. Side Chains. *J Med Chem*, **42**, 5095–5099.
- Bergmann, R., Linusson, A. & Zamora, I. (2007). SHOP: scaffold HOPping by GRID-based similarity searches. *J Med Chem*, **50** (11), 2708–2717.
- Berman, H.M. (2008). The Protein Data Bank: a historical perspective. *Acta Cryst A*, **64**, 88–95.
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. & Bourne, P.E. (2000). The Protein Data Bank. *Nucleic Acids Res*, **28** (1), 235–242.
- Bernstein, F.C., Koetzle, T.F., Williams, G.J., Meyer, E.E., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). The Protein Data Bank: a computer-based archival file for macromolecular structures. *J Mol Biol*, **112**, 535–542.
- BibDesk (2008). <http://bibdesk.sourceforge.net/>.

- Bissantz, C., Folkers, G. & Rognan, D. (2000). Protein-Based Virtual Screening of Chemical Databases. 1. Evaluation of Different Docking/Scoring Combinations. *J Med Chem*, **43**, 4759–4767.
- Bitetti-Putzer, R., Joseph-McCarthy, D, Hogle, J.M. & Karplus, M. (2001). Functional group placement in protein binding sites: a comparison of GRID and MCSS. *J Comput-Aided Mol Design*, **15**, 935–960.
- Bleicher, K.H., Böhm, H.-J., Müller, K. & Alanine, A.I. (2003). Hit and Lead Generation: Beyond High-Throughput Screening. *Nat Rev Drug Discov*, **2**, 369–378.
- Boda, K. & Johnson, A.P. (2006). Molecular Complexity Analysis of de Novo Designed Ligands. *J Med Chem*, **49** (20), 5869–5879.
- Boda, K., Seidel, T. & Gasteiger, J. (2007). Structure and reaction based evaluation of synthetic accessibility. *J Comput Aided Mol Des*, **21** (6), 311–325.
- Bohacek, R.S. & McMartin, C. (1994). Multiple highly diverse structures complementary to enzyme binding sites: results of extensive application of a de novo design method incorporating combinatorial growth. *J Am Chem Soc*, **116**, 5560–5571.
- Bohacek, R.S., McMartin, C., Glunz, P. & Rich, D.H. (1999). GrowMol, a de novo computer program, and its application to thermolysin and pepsin: results of the design and synthesis of a novel inhibitor. *IMA Volumes in Mathematics and Its Applications*, **108**, 103–114.
- Bohacek, R.S., McMartin, C. & Guida, W.C. (1996). The art and practice of structure-based drug design: a molecular modelling perspective. *Med Res Rev*, **16**, 3–50.
- Böhm, H.-J. (1992a). The computer program LUDI: A new method for the de novo design of enzyme inhibitors. *J Comput-Aided Mol Design*, **6**, 61–78.
- Böhm, H.-J. (1992b). LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads. *J Comput-Aided Mol Design*, **6**, 593–606.
- Böhm, H.-J. (1994). The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure. *J Comput-Aided Mol Design*, **8**, 243–256.
- Böhm, H.-J. (1995). Site directed structure generation by fragment joining. *Perspect Drug Discov Design*, **3**, 21–33.
- Böhm, H.-J. (1996a). Current computational tools for de novo ligand design. *Current Opinion in Biotechnology*, **7**, 433–436.
- Böhm, H.-J. (1996b). Towards the automatic design of synthetically accessible protein ligands: Peptides, amides and peptidomimetics. *J Comput-Aided Mol Design*, **10**, 265–272.

- Böhm, H.-J. (1998). Prediction of binding constants of protein ligands: A fast method for the prioritization of hits obtained from de novo design or 3D database search programs. *J Comput-Aided Mol Design*, **12**, 309–323.
- Böhm, H.-J., Flohr, A. & Stahl, M. (2004). Scaffold Hopping. *Drug Discov Today: Technologies*, **1** (3), 217–224.
- Böhm, H.-J. & Klebe, G. (1996). What Can We Learn from Molecular Recognition in Protein-Ligand Complexes for the Design of New Drugs? *Angew Chem Int Ed Engl*, **35**, 2588–2614.
- Böhm, H.-J., Klebe, G. & Kubinyi, H. (2002). *Wirkstoffdesign*. Spektrum Akademischer Verlag.
- Bolin, J.T., Filman, D.J., Matthews, D.A., Hamling, R.C. & Kraut, J. (1982). Crystal Structures of Escherichia coli and Lactobacillus casei Dihydrofolate Reductase Refined at 1.7 Å Resolution. *J Biol Chem*, **257** (22), 13650–13662.
- Broughton, H.B. & Watson, I.A. (2004). Selection of heterocycles for drug design. *J Mol Graph Model*, **23**, 51–58.
- Brown, N., McKay, B. & Gasteiger, J. (2006). A novel workflow for the inverse QSPR problem using multiobjective optimization. *J Comput Aided Mol Des*, **20** (5), 333–341.
- Brown, N., McKay, B., Gilardoni, F. & Gasteiger, J. (2004). A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *J Chem Inf Comput Sci*, **44** (3), 1079–1087.
- Brown, R.D. & Martin, Y.C. (1996). Use of Structure-Activity Data To Compare Structure-Based Clustering Methods and Descriptors of Use in Compound Selection. *J Chem Inf Comput Sci*, **36**, 572–584.
- Brown, R.D. & Martin, Y.C. (1997). The Information Content of 2D and 3D Structural Descriptors Relevant to Ligand-Receptor Binding. *J Chem Inf Comput Sci*, **37**, 1–9.
- Brzozowski, A.M., Pike, A.C., Dauter, Z., Hubbard, R.E., Bonn, T., Engstrom, O., Ohman, L., Greene, G.L., Gustafsson, J.A. & Carlquist, M. (1997). Molecular basis of agonism and antagonism in the oestrogen receptor. *Nature*, **389** (6652), 753–758.
- Budin, N., Ahmed, S., Majeux, N. & Caffisch, A. (2001). An evolutionary approach for structure-based design of natural and non-natural peptidic ligands. *Comb Chem High Throughput Screen*, **4** (8), 661–673.
- Buijsman, R. (2004). Structural Aspects of Kinases and Their Inhibitors. In: *Chemogenomics in Drug Discovery*, (Kubinyi, H. & Müller, G., Hrsg.), S. 191–219. Wiley-VCH, Weinheim.

- Bures, M.G. (1997). Recent techniques and applications in pharmacophore mapping. In: *Practical application of computer-aided drug design*, (Charifson, P.S., Hrsg.), S. 39–72. Marcel Dekker, New York.
- Bursulaya, B.D., Totrov, M., Abagyan, R. & Brooks III, C.L. (2003). Comparative study of several algorithms for flexible ligand docking. *J Comput-Aided Mol Design*, **17**, 755–763.
- Butcher, S.J., Grimes, J.M., Makeyev, E.V., Bamford, D.H. & Stuart, D.I. (2001). A mechanism for initiating RNA-dependent RNA polymerization. *Nature*, **410** (6825), 235–240.
- Bywater, R.P., Poulsen, T.A., Rogen, P. & Hjorth, P.G. (2004). De Novo Generation of Molecular Structures Using Optimization To Select Graphs on a Given Lattice. *J Chem Inf Comput Sci*, **44**, 856–861.
- CambridgeSoft (2005). *ChemDraw*. CambridgeSoft Corporation, Cambridge, MA, USA.
- Carhart, R., Smith, D.H. & Venkataraghavan, R. (1985). Atom pairs as molecular features in structure-activity studies: definition and application. *J Chem Inf Comput Sci*, **23**, 64–73.
- Carter, J.S. (1998). Recently reported inhibitors of cyclooxygenase-2. *Exp Opin Ther Patents*, **8** (1), 21–29.
- Cavanagh, J., Fairbrother, W.J., Skelton, N.J. & Palmer, A.G. (1996). *Protein NMR Spectroscopy: Principles and Practice*. Academic Press.
- Cheng, Y. & Prusoff, W.H. (1973). Relationship between inhibition constant (K_i) and concentration of inhibitor which causes 50 per cent inhibition (I_{50}) of an enzymatic-reaction. *Biochem Pharmacol*, **22**, 3099–3108.
- Clark, D.E., Frenkel, D., Levy, S.A., Li, J., Murray, C.W., Robson, B., Waszkowycz, B. & Westhead, D.R. (1995). PRO_LIGAND: An approach to de novo molecular design. 1. Application to the design of organic molecules. *J Comput-Aided Mol Design*, **9**, 13–32.
- Clark, D.E., Jones, G., Willet, P., Kenny, P.W. & Glen, R.C. (1994). Pharmacophoric Pattern Matching in Files of Three-Dimensional Chemical Structures: Comparison of Conformational-Searching Algorithms for Flexible Searching. *J Chem Inf Comput Sci*, **34**, 197–206.
- Clark, D.E. & Murray, C.W. (1995). PRO_LIGAND: An Approach to de Novo Molecular Design. 5. Tools for the Analysis of Generated Structures. *J Chem Inf Comput Sci*, **35** (5), 914–923.
- Claußen, H., Buning, C., Rarey, M. & Lengauer, T. (2001). FlexE: Efficient Molecular Docking Considering Protein Structure Variations. *J Mol Biol*, **308**, 377–395.

- Coghlan, M.J. & Kort, M.E. (1999). Novel ligands of steroid hormone receptors. *Exp Opin Ther Patents*, **9** (11), 1523–1536.
- Cole, J.C., Murray, C.W., Nissink, J.W., Taylor, R.D. & Taylor, R. (2005). Comparing protein-ligand docking programs is difficult. *Proteins*, **60** (3), 325–332.
- Congreve, M., Carr, R., Murray, C. & Jhoti, H. (2003). A 'Rule of Three' for fragment-based lead discovery? *Drug Discov Today*, **8** (19), 867–877.
- Corey, E.J. & Cheng, X.-M. (1995). *The Logic of Chemical Synthesis*. Wiley & Sons, New York.
- Corey, E.J., Czakó, B. & Kürti, L. (2007). *Molecules and Medicine*. Wiley & Sons, New York.
- Corey, E.J. & Wipke, W.T. (1969). Computer-assisted design of complex organic synthesis. *Science*, **166** (3902), 178–192.
- Cramer, R.D., Soltanshahi, F., Jilek, R. & Campbell, B. (2007). AllChem: generating and searching 10^{20} synthetically accessible structures. *J Comput Aided Mol Des*, **21** (6), 341–350.
- Crippen, G.M. & Havel, T.F. (1988). *Distance Geometry and Molecular Conformation*. Research Studies Press LTD.
- Cross, K.P., Myatt, G., Yang, C., Fligner, M.A., Verducci, J.S. & Blower, P.E.Jr. (2003). Finding Discriminating Structural Features by Reassembling Common Building Blocks. *J Med Chem*, **46**, 4770–4775.
- Cummings, M.D., DesJarlais, R.L., Gibbs, A.C., Mohan, V. & Jaeger, E.P. (2005). Comparison of Automated Docking Programs as Virtual Screening Tools. *J Med Chem*, **48**, 962–976.
- Davies, T.G., Bentley, J., Arris, C.E., Boyle, F.T., Curtin, N.J., Endicott, J.A., Gibson, A.E., Golding, B.T., Griffin, R.J., Hardcastle, I.R., Jewsbury, P., Johnson, L.N., Mesguiche, V., Newell, D.R., Noble, M.E.M., Tucker, J.A., Wang, L. & Whitfield, H.J. (2002). Structure-based design of a potent purine-based cyclin-dependent kinase inhibitor. *Nat Struct Biol*, **9** (10), 745–749.
- Daylight (2007). <http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html> (Version vom Sep. 2007). Daylight Chemical Information Systems, Inc.
- Dean, P.M. (1995). Defining molecular similarity and complementarity for drug design. In: *Molecular Similarity in Drug Design*, (Dean, P.M., Hrsg.), S. 1–23. Blackie Academic & Professional, London.
- Dean, P.M. (2007). Chemical genomics: a challenge for de novo drug design. *Mol Biotechnol*, **37** (3), 237–245.

- Dean, P.M., Lloyd, D.G. & Todorov, N.P. (2004). De novo drug design: Integration of structure-based and ligand-based methods. *Current Opinion in Drug Discovery and Development*, **7** (3), 347–353.
- Degen, J. & Rarey, M. (2006). FlexNovo: Structure-Based Searching in Large Fragment Spaces. *ChemMedChem*, **1** (8), 854–868.
- Degen, J. & Rarey, M. (2007). *FlexNovo*. Center for Bioinformatics, Hamburg.
- Degen, J., Wegscheid-Gerlach, C., Zaliani, A. & Rarey, M. (2008). On The Art of Compiling and Using ‘Drug-Like’ Chemical Fragment Spaces. *ChemMedChem*, **3** (10), 1503–1507.
- DeLano (2006). *PyMOL*. DeLano Scientific LLC, Palo Alto, CA, USA.
- van Deursen, R. & Reymond, J.-L. (2007). Chemical Space Travel. *ChemMedChem*, **2** (5), 636–640.
- DeWitte, R.S., Ishchenko, A.V. & Shakhnovich, E.I. (1997). SMOG: de Novo Design Method Based on Simple, Fast, and Accurate Free Energy Estimates. 2. Case Studies in Molecular Design. *J Am Chem Soc*, **119**, 4608–4617.
- DeWitte, R.S. & Shakhnovich, E.I. (1996). SMOG: de Novo Design Method Based on Simple, Fast, and Accurate Free Energy Estimates. 1. Methodology and Supporting Evidence. *J Am Chem Soc*, **118**, 11733–11744.
- Dey, F. & Caffisch, A. (2008). Fragment-based de novo ligand design by multiobjective evolutionary optimization. *J Chem Inf Model*, **48** (3), 679–690.
- Dobson, C.M. (2004). Chemical space and biology. *Nature*, **432**, 824–828.
- Doscher, M.S. & Richards, F.M. (1963). The Activity of an Enzyme in the Crystalline State: Ribonuclease S. *J Biol Chem*, **238**, 2399–2406.
- Douguet, D., Munier-Lehmann, H., Labesse, G. & Pochet, S. (2005). LEA3D: A Computer-Aided Ligand Design for Structure-Based Drug Design. *J Med Chem*, **48**, 2457–2468.
- Douguet, D., Thoreau, E. & Grassy, G. (2000). A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm. *J Comput-Aided Mol Design*, **14**, 449–466.
- Downs, G.M. & Willet, P. (1996). Similarity Searching in Databases of Chemical Structures. In: *Reviews in Computational Chemistry*, (Lipkowitz, K.B. & Boyd, D.B., Hrsg.), Bd. 7, S. 1–66. VCh Publishers, Inc., New York.
- van Drie, J.H. (2007). Computer-aided drug design: the next 20 years. *J Comput Aided Mol Des*, **21** (10-11), 591–601.

- Dullweber, F., Stubbs, M.T., Musil, D., Stürzebecher, J. & Klebe, G. (2001). Factorising ligand affinity: a combined thermodynamic and crystallographic study of trypsin and thrombin inhibition. *J Mol Biol*, **313** (3), 593–614.
- Dumas, J. (2001). Protein kinase inhibitors: emerging pharmacophores 1997 - 2000. *Exp Opin Ther Patents*, **11** (3), 405–429.
- Durant, J.L., Leland, B.A., Henry, D.R. & Nourse, J.G. (2002). Reoptimization of MDL keys for use in drug discovery. *J Chem Inf Comput Sci*, **42** (6), 1273–1280.
- Eisen, M.B., Wilez, D.C., Karplus, M. & Hubbard, R.E. (1994). HOOK: A Program for Finding Novel Molecular Architectures That Satisfy the Chemical and Steric Requirements of a Macromolecular Binding Site. *Proteins: Structure, Function and Genetics*, **19**, 199–221.
- Eksterowicz, J.E., Evensen, E., Lemmen, C., Brady, G.P., Lanctot, J.K., Bradley, E.K., Saiah, E., Robinson, L.A., Grootenhuis, P.D.J. & Blaney, J.M. (2002). Coupling structure-based design with combinatorial chemistry: application of active site derived pharmacophores with informative library design. *J Mol Graph Model*, **20**, 469–477.
- Eldridge, M.D, Murray, C.W., Auton, T.R., Paolini, G.V. & Mee, R.P. (1997). Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *J Comput Aided Mol Des*, **11** (5), 425–445.
- Erlanson, D.A., McDowell, R.S. & O'Brien, T. (2004). Fragment-Based Drug Discovery. *J Med Chem*, **47** (14), 3463–3482.
- Ewing, T.J.A. & Kuntz, I.D. (1997). Critical evaluation of search algorithms for automated molecular docking and database screening. *J Comput Chem*, **18**, 1175–1189.
- Fattori, D. (2004). Molecular recognition: the fragment approach in lead generation. *Drug Discov Today*, **9** (5), 229–238.
- Fechner, U. & Schneider, G. (2006). Flux (1): a virtual synthesis scheme for fragment-based de novo design. *J Chem Inf Model*, **46** (2), 699–707.
- Fechner, U. & Schneider, G. (2007). Flux (2): comparison of molecular mutation and crossover operators for ligand-based de novo design. *J Chem Inf Model*, **47** (2), 656–667.
- Fink, T., Bruggesser, H. & Reymond, J.-L. (2005). Virtual exploration of the small-molecule chemical universe below 160 Daltons. *Angew Chem Int Ed Engl*, **44** (10), 1504–1508.
- Fink, T. & Reymond, J.-L. (2007). Virtual exploration of the chemical universe up to 11 atoms of C, N, O, F: assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery. *J Chem Inf Model*, **47** (2), 342–353.

- Firth-Clark, S., Todorov, N. P., Alberts, I. L., Williams, A., James, T. & Dean, P. M. (2006a). Exhaustive de novo design of low-molecular-weight fragments against the ATP-binding site of DNA-gyrase. *J Chem Inf Model*, **46** (3), 1168–1173.
- Firth-Clark, S., Willems, H. M., Williams, A. & Harris, W. (2006b). Generation and selection of novel estrogen receptor ligands using the de novo structure-based design tool, SkelGen. *J Chem Inf Model*, **46** (2), 642–647.
- Fischer, E. (1894). Einfluß der Konfiguration auf die Wirkung der Enzyme. *Ber Dtsch Chem Ges*, **27**, 2985–2993.
- Fischer, J.R. & Rarey, M. (2007). SwiFT: An Index Structure for Reduced Graph Descriptors in Virtual Screening and Clustering. *J Chem Inf Model*, **47** (4), 1341–1353.
- Flower, R.J. (1998). On the Properties of Bit String-Based Measures of Chemical Similarity. *J Chem Inf Comput Sci*, **38**, 379–386.
- Frenkel, D., Clark, D. E., Li, J., Murray, C. W., B, R. Obson, Waszkowycz, B. & Westhead, D. R. (1995). PRO_LIGAND: an approach to de novo molecular design. 4. Application to the design of peptides. *J Comput Aided Mol Des*, **9** (3), 213–225.
- Friesen, R.W., Brideau, C., Chan, C.C., Charleson, S., Deschenes, D., Dube, D., Ethier, D., Fortin, R., Gauthier, J.Y., Girard, Y., Gordon, R., Greig, G.M., Riendeau, D., Savoie, C., Wang, Z., Wong, E., Visco, D., Xu, L.J. & Young, R.N. (1998). 2-pyridinyl-3-(4-methylsulfonyl)phenylpyridines: selective and orally active cyclooxygenase-2 inhibitors. *Biorg Med Chem Lett*, **8** (19), 2777–2782.
- Gasteiger, J. (2007). De novo design and synthetic accessibility. *J Comput Aided Mol Des*, **21** (6), 307–309.
- Gasteiger, J. Pfortner, M., Sitzmann, M., Höllering, R., Sacher, O., Kostka, T. & Karg, N. (2000). Computer-assisted synthesis and reaction planning in combinatorial chemistry. *Perspect Drug Discov Design*, **20**, 245–246.
- Gastreich, M., Lilienthal, M., Briem, H. & Claussen, H. (2006). Ultrafast de novo docking combining pharmacophores and combinatorics. *J Comput Aided Mol Des*, **20** (12), 717–734.
- Gedeck, P., Rohde, B. & Bartels, C. (2006). QSAR—how good is it in practice? Comparison of descriptor sets on an unbiased cross section of corporate data sets. *J Chem Inf Model*, **46** (5), 1924–1936.
- Gegnas, L.D. (2000). Patent focus on agents for osteoporosis: September 1999 - February 2000. *Exp Opin Ther Patents*, **10** (6), 833–846.
- Gehlhaar, D.K., Moerder, K.E., Zichi, D., Sherman, C.J., Ogden, R.C. & Freer, S.T. (1995). De Novo Design of Enzyme Inhibitors by Monte Carlo Ligand Generation. *J Med Chem*, **38**, 466–472.

- Gerlach, C., Munzel, M., Baum, B., Gerber, H.-D., Craan, T., Diederich, W.E. & Klebe, G. (2007). KNOBLE: a knowledge-based approach for the design and synthesis of readily accessible small-molecule chemical probes to test protein binding. *Angew Chem Int Ed Engl*, **46** (47), 9105–9109.
- Gillet, V.J., Johnson, A.P., Mata, P., Sike, S. & Williams, P. (1993). SPROUT: A program for structure generation. *J Comput-Aided Mol Design*, **7**, 127–153.
- Gillet, V.J., Johnson, A.P. & Sike, S. (1990). Automated structure design in 3D. *Tetrahedron Comput Methodol*, **3** (6), 681–696.
- Gillet, V.J., Myatt, G., Zsoldos, Z. & Johnson, A.P. (1995). SPROUT, HIPPO and CAESA: Tools for de novo structure generation and estimation of synthetic accessibility. *Perspect Drug Discov Design*, **3**, 34–50.
- Gillet, V.J., Newell, W., Mata, P., Myatt, G., Sike, S., Zsoldos, Z. & Johnson, A.P. (1994). SPROUT: Recent Developments in the de Novo Design of Molecules. *J Chem Inf Comput Sci*, **34**, 207–217.
- Gillet, V.J. & Nicolotti, O. (2000). Evaluation of reactand-based and product-based approaches to the design of combinatorial libraries. *Perspect Drug Discov Design*, **20**, 265–287.
- Gillet, V.J., Willet, P. & Bradshaw, J. (2003). Similarity Searching Using Reduced Graphs. *J Chem Inf Comput Sci*, **43** (2), 338–345.
- Glen, R.C. & Adams, S.E. (2006). Similarity metrics and descriptor spaces - Which combinations to choose? *QSAR Comb Sci*, **25** (12), 1133–1142.
- Glen, R.C. & Payne, A.W. (1995). A genetic algorithm for the automated generation of molecules within constraints. *J Comput Aided Mol Des*, **9** (2), 181–202.
- Globus, A., Lawton, J. & Wipke, W.T. (1999). Automatic Molecular design using evolutionary algorithms. *Nanotechnology*, **10**, 290–299.
- gnuplot (2008). <http://www.gnuplot.info/>.
- Gohlke, H., Hendlich, M. & Klebe, G. (2000). Knowledge-based scoring function to predict protein-ligand interactions. *J Mol Biol*, **295** (2), 337–356.
- Gohlke, H. & Klebe, G. (2002). Ansätze zur Beschreibung und Vorhersage der Bindungsaffinität niedermolekularer Liganden an makromolekulare Rezeptoren. *Angew Chem*, **114**, 2764–2798.
- Gola, J., Obrezanova, O., Champness, E. & Segall, M. (2006). ADMET property prediction: The state of the art and current challenges. *QSAR Comb Sci*, **25** (12), 1172–1180.
- Goodford, P.J. (1985). A Computational Procedure for Determining Energetically Favorable Binding Sites on Biologically Important Macromolecules. *J Med Chem*, **28**, 849–857.

- Grace (2008). <http://plasma-gate.weizmann.ac.il/Grace/>.
- Gribbon, P., Lyons, R., Laffin, P., Bradley, J., Chambers, C., Williams, B.S., Keighley, W. & Sewing, A. (2005). Evaluating real-life high-throughput screening data. *J Biomol Screen*, **10** (2), 99–107.
- Gribbon, P. & Sewing, A. (2005). High-throughput drug discovery: what can we expect from HTS? *Drug Discov Today*, **10** (1), 17–22.
- Griewel, A. & Rarey, M. (2005). From greedy to branch & bound and back: Assessing optimization strategies for incremental construction molecular docking tools. In: *German Conference on Bioinformatics GCB 2005*, S. 119–130, Hamburg. Gesellschaft für Informatik (GI), Bonn.
- Gubernator, K., Broger, C., Bur, D., Doran, D., Geyer, P., Muller, K. & Schaumann, T. (1995). Structure-based ligand design. In: *Computer-Aided Drug Design in Industrial Research*, (Herrmann, E. & Frankle, R., Hrsg.), S. 61–77. Springer, Berlin.
- van Gunsteren, W.F., Bakowies, D., Baron, R., Chandrasekhar, I., Christen, M., Daura, X., Gee, P., Geerke, D.P., Glattli, A., Hunenberger, P.H., Kastenholtz, M. A., Ostentbrink, C., Schenk, M., Trzesniak, D., van der Vegt, N.F.A. & Yu, H.B. (2006). Biomolecular modeling: Goals, problems, perspectives. *Angew Chem Int Ed*, **45** (25), 4064–4092.
- Hajduk, P.J. (2006). Fragment-based drug design: how big is too big? *J Med Chem*, **49** (24), 6972–6976.
- Hajduk, P.J. & Greer, J. (2007). A decade of fragment-based drug design: strategic advances and lessons learned. *Nat Rev Drug Discov*, **6** (3), 211–219.
- Halperin, I., Ma, B., Wolfson, H. & Nussinov, R. (2002). Principles of Docking: An Overview of Search Algorithms and a Guide to Scoring Functions. *Proteins: Structure, Function and Genetics*, **47**, 409–443.
- Hanessian, S. (2005). Man, machine and visual imagery in strategic synthesis planning: Computer-perceived precursors for drug candidates. *Curr Opin Drug Discovery Dev*, **8** (6), 798–819.
- Hanessian, S. (2006). Structure-based organic synthesis of drug prototypes: a personal odyssey. *ChemMedChem*, **1** (12), 1301–1330.
- Hann, M.M., Leach, A.R. & Harper, G (2001). Molecular Complexity and Its Impact on the Propability of Finding Leads for Drug Discovery. *J Chem Inf Comput Sci*, **41**, 856–864.
- Harris, R., Olson, A.J. & Goodsell, D.S. (2008). Automated prediction of ligand-binding sites in proteins. *Proteins*, **70** (4), 1506–1517.

- Hartenfeller, M., Proschak, E., Schüller, A. & Schneider, G. (2008). Concept of combinatorial de novo design of drug-like molecules by particle swarm optimization. *Chem Biol Drug Des*, **72** (1), 16–26.
- Hartshorn, M.J., Murray, C.W., Cleasby, A., Frederickson, M., Tickle, I.J. & Jhoti, H. (2005). Fragment-Based Lead Discovery Using X-ray Crystallography. *J Med Chem*, **48**, 403–413.
- Herschhorn, A., Lerman, L., Weitman, M., Gleenberg, I. O., Nudelman, A. & Hizi, A. (2007). De Novo Parallel Design, Synthesis and Evaluation of Inhibitors against the Reverse Transcriptase of Human Immunodeficiency Virus Type-1 and Drug-Resistant Variants. *J Med Chem*, **50** (14), 3402.
- Hert, J., Willett, P., Wilton, D.J., Acklin, P., Azzaoui, K., Jacoby, E. & Schuffenhauer, A. (2006). New methods for ligand-based virtual screening: use of data fusion and machine learning to enhance the effectiveness of similarity searching. *J Chem Inf Model*, **46** (2), 462–470.
- Hessler, G., Zimmermann, M., Matter, H., Evers, A., Naumann, T., Lengauer, T. & Rarey, M. (2005). Multiple-ligand-based virtual screening: methods and applications of the MTree approach. *J Med Chem*, **48** (21), 6575–6584.
- Hindle, S.A., Rarey, M., Bunning, C. & Lengauer, T. (2002). Flexible docking under pharmacophore type constraints. *J Comput-Aided Mol Design*, **16**, 129–149.
- Ho, C.M.W. & Marshall, G.R. (1993). SPLICE: a program to assemble partial query solutions from three-dimensional database searches into novel ligands. *J Comput-Aided Mol Design*, **7** (6), 623–647.
- Höllering, R., Gasteiger, J., Steinhauer, L., Schulz, K.-P. & Herwig, A. (2000). Simulation of Organic Reactions: From the Degradation of Chemicals to Combinatorial Synthesis. *J Chem Inf Comput Sci*, **40**, 482–494.
- Honma, T. (2003). Recent Advances in De Novo Design Strategy for Practical Lead Identification. *Medicinal Research Reviews*, **23** (5), 606–632.
- Honma, T., Hayashi, K., Aoyama, T., Hashimoto, N., Machida, T., Fukasawa, K., Iwama, T., Ikeura, C., Ikuta, M., Suzuki-Takahashi, I., Iwasawa, Y., Hayama, T., Nishimura, S. & Morishima, H. (2001). Structure-Based Generation of a Class of Potent Cd4k Inhibitors: New de Novo Design Strategy and Library Design. *J Med Chem*, **44**, 4615–4627.
- Hopkins, A.L., Groom, C.R. & Alex, A. (2004). Ligand efficiency: a useful metric for lead generation. *Drug Discov Today*, **9** (10), 430–431.
- Howard, N., Abell, C., Blakemore, W., Chessari, G., Congreve, M., Howard, S., Jhoti, H., Murray, C. W., Seavers, L. C. & van Montfort, R. L. (2006). Application of fragment screening and fragment linking to the discovery of novel thrombin inhibitors. *J Med Chem*, **49** (4), 1346–1355.

- Huang, N., Shoichet, B. K. & Irwin, J. J. (2006). Benchmarking sets for molecular docking. *J Med Chem*, **49** (23), 6789–6801.
- Hubbard, R.D. & Wilsbacher, J.L. (2007). Advances towards the Development of ATP-Competitive Small-Molecule Inhibitors of the Insulin-Like Growth Factor Receptor (IGF-IR). *ChemMedChem*, **2** (1), 41–46.
- Irwin, J.J. & Shoichet, B.K. (2005). ZINC—a free database of commercially available compounds for virtual screening. *J Chem Inf Model*, **45** (1), 177–182.
- Ishchenko, A.V. & Shakhnovich, E.I. (2002). SMoG2001 (SMoG2001): an improved knowledge-based scoring function for protein-ligand interactions. *J Med Chem*, **45** (13), 2770–2780.
- Jacobsson, M. & Karlen, A. (2006). Ligand bias of scoring functions in structure-based virtual screening. *J Chem Inf Model*, **46** (3), 1334–1343.
- Jain, A.N. (2003). Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. *J Med Chem*, **46** (4), 499–511.
- Jain, A.N. (2004). Ligand-Based Structural Hypotheses for Virtual Screening. *J Med Chem*, **47**, 947–961.
- Johnson, A.P., Boda, K., Lengyel, T. & Weaver, S. (2004). Improved methods for the de novo design of synthetically accessible ligands. In: *228th ACS National Meeting*, Philadelphia, PA, United States. American Chemical Society.
- Johnson, M.A. & Maggiora, G.M., Hrsg. (1990). *Concepts and Applications of Molecular Similarity*. Wiley & Sons, New York.
- Jorgensen, W.L. (2004). The many roles of computation in drug discovery. *Science*, **303** (5665), 1813–1818.
- Kalgutkar, A.S. (1999). Selective cyclooxygenase-2 inhibitors as non-ulcerogenic anti-inflammatory agents. *Exp Opin Ther Patents*, **9** (7), 831–849.
- Kazius, J., McGuire, R. & Bursi, R. (2005). Derivation and Validation of Toxicophores for Mutagenicity Prediction. *J Med Chem*, **48**, 312–320.
- Kitchen, D.B., Decornez, H., Furr, J.R. & Bajorath, J. (2004). Docking and scoring in virtual screening for drug discovery: methods and applications. *Nat Rev Drug Discov*, **3**, 935–949.
- Klebe, G. (1993). Structural alignment of molecules. In: *3D QSAR in Drug Design: Theory, Methods and Applications*, (Kubinyi, H., Hrsg.), S. 173–199. ESCOM, Leiden.
- Klebe, G. (1998). Molecular similarity - a guideline for the design of new protein ligands. In: *Rational Molecular Design in Drug Research*, (Liljefors, T., Jorgensen, F.S. & Krosgaard-Larsen, P., Hrsg.), S. 151–160. Munksgaard, Kopenhagen.

- Klebe, G. (2000). Recent developments in structure-based drug design. *J Mol Med*, **78**, 269–281.
- Klebe, G. & Mietzner, T. (1994). A fast and efficient method to generate biologically relevant conformations. *J Comput Aided Mol Des*, **8** (5), 583–606.
- Knegtel, R.M., Boelens, R., Ganadu, M.L. & Kaptein, R. (1991). The solution structure of a monomeric insulin. A two-dimensional ¹H-NMR study of des-(B26-B30)-insulin in combination with distance geometry and restrained molecular dynamics. *Eur J Biochem*, **202** (2), 447–458.
- Kolb, H.C., Finn, M.G. & Sharpless, K.B. (2001). Click Chemistry: Diverse Chemical Function from a Few Good Reactions. *Angew Chem Int Ed Engl*, **40** (11), 2004–2021.
- Kolb, H.C. & Sharpless, K.B. (2003). The growing impact of click chemistry on drug discovery. *Drug Discov Today*, **8** (24), 1128–1137.
- Kolb, P. & Caffisch, A. (2006). Automatic and efficient decomposition of two-dimensional structures of small molecules for fragment-based high-throughput docking. *J Med Chem*, **49** (25), 7384–7392.
- Koshland, D.E. & Neet, K.E. (1968). The catalytic and regulatory properties of enzymes. *Annu Rev Biochem*, **37**, 359–410.
- Krier, M., de Araújo-Júnior, J.X., Schmitt, M., Durantou, J., Justiano-Basaran, H., Lugnier, C., Bourguignon, J.-J. & Rognan, D. (2005). Design of Small-Sized Libraries by Combinatorial Assembly of Linkers and Functional Groups to a Given Scaffold: Application to the Structure-Based Optimization of a Phosphodiesterase 4 Inhibitor. *J Med Chem*, **48**, 3816–3822.
- Kubinyi, H., Hrsg. (1993). *3D-QSAR in Drug Design: Theory, Methods and Applications*. ESCOM, Leiden.
- Kubinyi, H. (1998a). Combinatorial and computational approaches in structure-based drug design. *Curr Opin Drug Discov Devel*, **1** (1), 16–27.
- Kubinyi, H. (1998b). Structure-based design of enzyme inhibitors and receptor ligands. *Curr Opin Drug Discov Devel*, **1** (1), 4–15.
- Kubinyi, H. (2003). Drug research: myths, hype and reality. *Nat Rev Drug Discov*, **2**, 665–668.
- Kubinyi, H., Folkers, G. & Martin, Y.C., Hrsg. (1997). *3D-QSAR in Drug Design: Recent Advances*. ESCOM/Kluwer, Leiden.
- Kühlbrandt, W. & Williams, K.A. (1999). Analysis of macromolecular structure and dynamics by electron cryo-microscopy. *Curr Opin Chem Biol*, **3**, 537–543.

- Kuntz, I.D. (1992). Structure-Based Strategies for Drug Design and Discovery. *Science*, **257**, 1078–1082.
- Kuntz, I.D., Meng, E. & Shoichet, B.K. (1994). Structure-Based Molecular Design. *Acc Chem Res*, **27** (5), 117–123.
- Kurumbail, R.G., Stevens, A.M., Gierse, J.K., McDonald, J.J., Stegeman, R.A., Pak, J.Y., Gildehaus, D., Miyashiro, J.M., Penning, T.D., Seibert, K., Isakson, P.C. & Stallings, W.C. (1996). Structural basis for selective inhibition of cyclooxygenase-2 by anti-inflammatory agents. *Nature*, **384** (6610), 644–648.
- Ladd, M.F.C. & Palmer, R.A. (2003). *Structure determination by X-ray crystallography*. Springer.
- Lajiness, M.S., Maggiora, G.M. & Shanmugasundaram, V. (2004). Assessment of the Consistency of Medicinal Chemists in Reviewing Sets of Compounds. *J Med Chem*, **47**, 4891–4896.
- Lameijer, E.W., Kok, J.N., Back, T. & Ijzerman, A.P. (2006a). Mining a chemical database for fragment co-occurrence: discovery of 'chemical clichés'. *J Chem Inf Model*, **46** (2), 553–562.
- Lameijer, E.W., Kok, J.N., Back, T. & Ijzerman, A.P. (2006b). The molecule evaluator. An interactive evolutionary algorithm for the design of drug-like molecules. *J Chem Inf Model*, **46** (2), 545–552.
- Lauri, G. & Bartlett, P.A. (1994). CAVEAT: A program to facilitate the design of organic molecules. *J Comput-Aided Mol Design*, **8**, 51–66.
- Lazar, C., Kluczyk, A., Kiyota, T. & Konishi, Y. (2004). Drug Evolution Concept in Drug Design: 1. Hybridisation Method. *J Med Chem*, **47**, 6973–6982.
- Leach, A.R., Bryce, R.A. & Robinson, A.J. (2000). Synergy between combinatorial chemistry and de novo design. *J Mol Graphics Modell*, **18**, 358–367.
- Leach, A.R. & Hann, M.M. (2000). The in silico world of virtual libraries. *Drug Discov Today*, **5** (8), 326–336.
- Leach, A.R. & Kilvington, S.R. (1994). Automated molecular design: A new fragment-joining algorithm. *J Comput-Aided Mol Design*, **8**, 283–298.
- Leach, A.R. & Lewis, R.A. (1994). A ring-bracing approach to computer-assisted ligand design. *J Comp Chem*, **15** (2), 233–240.
- Leach, A.R., Shoichet, B.K. & Peishoff, C.E. (2006). Prediction of protein-ligand interactions. Docking and scoring: successes and gaps. *J Med Chem*, **49** (20), 5851–5855.
- Leeson, P.D., Davis, A.M. & Steele, J. (2004). Drug-like properties: guiding principles for design - or chemical prejudice? *Drug Discov Today: Technologies*, **1** (3), 189–195.

- Lewell, X.Q., Judd, D.B., Watson, S.P. & Hann, M.M. (1998). RECAP—Retrosynthetic Combinatorial Analysis Procedure: A Powerful Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J Chem Inf Comput Sci*, **38**, 511–522.
- Lewis, R.A. (1990). Automated site-directed drug design: Approaches to the formation of 3D molecular graphs. *J Comput-Aided Mol Design*, **4**, 205–210.
- Lewis, R.A. (1992). Automated site-directed drug design: a method for the generation of general three-dimensional molecular graphs. *J Mol Graphics*, **10**, 131–143.
- Lewis, R.A. (2005). A General Method for Exploiting QSAR Models in Lead Optimization. *J Med Chem*, **48**, 1638–1648.
- Lewis, R.A. & Dean, P.M. (1989). Automated site-directed drug design: the concept of spacer skeletons for primary structure generation. *Proc R Soc Lond*, **B 236**, 125–140.
- Lewis, R.A., Roe, D.C., Huang, C., Ferrin, T.E., Langridge, R. & Kuntz, I.D. (1992). Automated site-directed drug design using molecular lattices. *J Mol Graphics*, **10**, 66–78.
- Lipinski, C. & Hopkins, A. (2004). Navigating chemical space for biology and medicine. *Nature*, **432**, 855–861.
- Lipinski, C.A., Lombardo, F., Dominy, B.W. & Feeney, P.J. (1997). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv Drug Deliv Rev*, **23**, 3–25.
- Liu, H., Duan, Z., Luo, Q. & Shi, Y. (1999). Structure-based ligand design by dynamically assembling molecular building blocks at binding site. *Proteins*, **36** (4), 462–470.
- Liu, Q., Masek, B., Smith, K. & Smith, J. (2007). Tagged fragment method for evolutionary structure-based de novo lead generation and optimization. *J Med Chem*, **50** (22), 5392–5402.
- Livingstone, D.J. (2000). The Characterization of Chemical Structures Using Molecular Properties. A Survey. *J Chem Inf Comput Sci*, **40**, 195–209.
- Lloyd, D.G., Buenemann, C.L., Todorov, N.P., Manallack, D.T. & Dean, P.M. (2004). Scaffold Hopping in De Novo Design. Ligand Generation in the Absence of Receptor Information. *J Med Chem*, **47** (3), 493–496.
- Luo, Z., Wang, R. & Lai, L. (1996). RASSE: a new method for structure-based drug design. *J Chem Inf Comput Sci*, **36** (6), 1187–1194.
- Lyne, P.D. (2002). Structure-based virtual screening: an overview. *Drug Discov Today*, **7** (20), 1047–1055.

- Maaß, P., Schulz-Gasch, T., Stahl, M. & Rarey, M. (2007). Recore: a fast and versatile method for scaffold hopping based on small molecule crystal structure conformations. *J Chem Inf Model*, **47** (2), 390–399.
- MacLean, D. & Martin, E.J. (2004). On the Representation of Combinatorial Libraries. *J Comb Chem*, **6** (1), 1–11.
- Magarian, R.A., Overacre, L.B., Singh, S. & Meyer, K.L. (1994). The medicinal chemistry of non-steroidal antiestrogens: a review. *Curr Med Chem*, **1**, 61–104.
- Mancera, R. L. (2002). De novo ligand design with explicit water molecules: an application to bacterial neuraminidase. *Journal of Computer-Aided Molecular Design*, **16** (7), 479–499.
- Martin, Y.C. (2006). What works and what does not: Lessons from experience in a pharmaceutical company. *QSAR Comb Sci*, **25** (12), 1192–1200.
- Mata, P., Gillet, V.J., Johnson, A.P., Lampreia, J., Myatt, G.J., Sike, S. & Stebbings, A. (1995). SPROUT-3D structure generation using templates. *J Chem Inf Comput Sci*, **35**, 479–493.
- Mausser, H. & Guba, W. (2008). Recent developments in de novo design and scaffold hopping. *Curr Opin Drug Discov Devel*, **11** (3), 365–374.
- Mausser, H. & Stahl, M. (2007). Chemical fragment spaces for de novo design. *J Chem Inf Model*, **47** (2), 318–324.
- Merlot, C., Domine, D. & Church, D.J. (2002). Fragment analysis in small molecule discovery. *Current Opinion in Drug Discovery and Development*, **5** (3), 391–399.
- Microsoft (2004). *Microsoft Excel*. Microsoft Corporation, Redmond, WA, USA.
- Miranker, A. & Karplus, M. (1991). Functionality Maps of Binding Sites: A Multiple Copy Simultaneous Search Method. *Proteins: Structure, Function and Genetics*, **11**, 29–34.
- Miranker, A. & Karplus, M. (1995). An Automated Method for Dynamic Ligand Design. *Proteins: Structure, Function and Genetics*, **23**, 472–490.
- Moon, J.B. & Howe, W.J. (1991). Computer Design of Bioactive Molecules: A Method for Receptor-Based de Novo Ligand Design. *Proteins: Structure, Function and Genetics*, **11**, 314–328.
- Muegge, I. & Rarey, M. (2001). Small Molecule Docking and Scoring. In: *Reviews in Computational Chemistry*, (Lipkowitz, K.B. & Boyd, D.B., Hrsg.), Bd. 17, S. 1–60. Wiley-VCH, Weinheim, New York.

- Murray, C.W., Clark, D.E., Auton, T.R., Firth, M.A., Li, J., Sykes, R.A., Waszkowycz, B., Westhead, D.R. & Young, S.C. (1997). PRO_SELECT: Combining structure-based design and combinatorial chemistry for rapid lead discovery. 1. Technology. *J Comput-Aided Mol Design*, **11**, 193–207.
- Murray, C.W., Clark, D.E. & Byrne, D.G. (1995). PRO_LIGAND: An approach to de novo molecular design. 6. Flexible fitting in the design of peptides. *J Comput-Aided Mol Design*, **9**, 381–395.
- Mutschler, E., Geisslinger, G., Kroemer, H.K., Ruth, P. & Schäfer-Korting, M. (2008). *Mutschler Arzneimittelwirkungen*. Wissenschaftliche Verlagsgesellschaft, Stuttgart.
- Nachbar, R.B. (2000). Molecular Evolution: Automated Manipulation of Hierarchical Chemical Topology and Its Application to Average Molecular Structures. *Genetic Programming and Evolvable Machines*, **1**, 57–94.
- Nicolaou, K.C. (2003). Perspectives in total synthesis: a personal account. *Tetrahedron*, **59**, 6683–6738.
- Nicolaou, K.C. & Montagnon, T. (2008). *Molecules that Changed the World*. Wiley-VCH, Weinheim.
- Nilakantan, R., Nunn, D. S., Greenblatt, L., Walker, G., Haraki, K. & Mobilio, D. (2006). A family of ring system-based structural fragments for use in structure-activity studies: database mining and recursive partitioning. *J Chem Inf Model*, **46** (3), 1069–1077.
- Nishibata, Y. & Itai, A. (1991). Automatic Creation of Drug Candidate Structures Based on Receptor Structure: Starting Point for Artificial Lead Generation. *Tetrahedron*, **47**, 8985–8990.
- Nishibata, Y. & Itai, A. (1993). Confirmation of usefulness of a structure construction program based on three-dimensional receptor structure for rational lead generation. *J Med Chem*, **36** (20), 2921–2928.
- Norinder, U. & Bergstrom, C. A. (2006). Prediction of ADMET Properties. *ChemMedChem*, **1** (9), 920–937.
- Oprea, T.I. (2000). Property distribution of drug-related chemical databases. *J Comput-Aided Mol Design*, **14**, 251–264.
- Oprea, T.I., Davis, A.M., Teague, S.J. & Leeson, P.D. (2001). Is There a Difference between Leads and Drugs? A Historical Perspective. *J Chem Inf Comput Sci*, **41**, 1308–1315.
- Pärn, J., Degen, J. & Rarey, M. (2007). Exploring fragment spaces under multiple physicochemical constraints. *J Comput Aided Mol Des*, **21** (6), 327–340.
- Pearlman, D.A. & Murcko, M.A. (1993). CONCEPTS: New Dynamic Algorithm for De Novo Drug Suggestion. *J Comput Chem*, **14** (10), 1184–1193.

- Pearlman, D.A. & Murcko, M.A. (1996). CONCERTS: Dynamic Connection of Fragments as an Approach to de Novo Ligand Design. *J Med Chem*, **39**, 1651–1663.
- Pegg, S.C.-H., Haresco, J.J. & Kuntz, I.D. (2001). A genetic algorithm for structure-based de novo design. *J Comput-Aided Mol Design*, **15**, 911–933.
- Pellegrini, E. & Field, M.J. (2003). Development and testing of a de novo drug-design algorithm. *J Comput-Aided Mol Design*, **17** (10), 621–641.
- Penning, T.D., Talley, J.J., Bertenshaw, S.R., Carter, J.S., Collins, P.W., Docter, S., Graneto, M.J., Lee, L.F., Malecha, J.W., Miyashiro, J.M., Rogers, R.S., Rogier, D.J., Yu, S.S., Burton, E.G., Cogburn, J.N., Gregory, S.A., Koboldt, C.M., Perkins, W.E., Seibert, K., Veenhuizen, A.W., Zhang, Y.Y. & Isakson, P.C. (1997). Synthesis and biological evaluation of the 1,5-diarylpyrazole class of cyclooxygenase-2 inhibitors: identification of 4-[5-(4-methylphenyl)-3-(trifluoromethyl)-1H-pyrazol-1-yl]benzenesulfonamide (SC-58635, celecoxib). *J Med Chem*, **40** (9), 1347–1365.
- Petsko, G.A. (1996). For medicinal purposes. *Nature*, **384**, 7–9.
- Pham, T.A. & Jain, A.N. (2006). Parameter estimation for scoring protein-ligand interactions using negative training data. *J Med Chem*, **49** (20), 5856–5868.
- Pickett, S.D., McLay, I.M. & Clark, D.E. (2000). Enhancing the hit-to-lead properties of lead optimization libraries. *J Chem Inf Comput Sci*, **40** (2), 263–272.
- Pierce, A.C., Rao, G. & Bemis, G.W. (2004). BREED: Generating Novel Inhibitors through Hybridisation of Known Ligands. Application to CDK2, P38 and HIV Protease. *J Med Chem*, **47**, 2768–2775.
- Pirok, G., Mate, N., Varga, J., Szegezdi, J., Vargyas, M., Dorant, S. & Csizmadia, F. (2006). Making 'real' molecules in virtual space. *J Chem Inf Model*, **46** (2), 563–568.
- Polanski, J., Bak, A., Gieleciak, R. & Magdziarz, T. (2006). Modeling robust QSAR. *J Chem Inf Model*, **46** (6), 2310–2318.
- Poli, R., Kennedy, J. & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, **1** (1), 33–57.
- Pollock, S.N., Coutsiaris, E.A., Wester, M.J. & Oprea, T.I. (2008). Scaffold Topologies. 1. Exhaustive Enumeration up to Eight Rings. *J Chem Inf Model*, **48** (7), 1304–1310.
- Pritchard, J.F., Jurima-Romer, M., Reimer, M.L.J., Mortimer, E., Rolfe, B. & Cayen, M.N. (2003). Making better drugs: decision gates in non-clinical drug development. *Nat Rev Drug Discov*, **2**, 542–553.
- PubChem (2007). <ftp://ftp.ncbi.nih.gov/pubchem> (Version vom 10. Nov. 2007).
- Raju, T.N. (2000). The Nobel chronicles. *Lancet*, **355**, 1022.

- Rarey, M. (2001). Protein-Ligand Docking in Drug Design. In: *Bioinformatics - From Genomes to Drugs*, (Lengauer, T., Hrsg.), Bd. I: Basic Technologies, S. 315–360. Wiley-VCH, Weinheim.
- Rarey, M., Degen, J. & Reulecke, I. (2007). Docking and Scoring for Structure-based Drug Design. In: *Bioinformatics - From Genomes to Therapies*, (Lengauer, T., Hrsg.), Bd. 2 d. Reihe *Bioinformatics - From Genomes to Therapies*, S. 541–599. Wiley-VCH, Weinheim, Weinheim.
- Rarey, M. & Dixon, J.S. (1998). Feature trees: A new molecular similarity measure based on tree matching. *J Comput-Aided Mol Design*, **12**, 471–490.
- Rarey, M., Kramer, B. & Lengauer, T. (1997). Multiple automatic base selection: Protein-ligand docking based on incremental construction without manual intervention. *J Comput-Aided Mol Design*, **11**, 369–384.
- Rarey, M., Kramer, B. & Lengauer, T. (1999a). Docking of hydrophobic ligands with interaction-based matching algorithms. *Bioinformatics*, **15** (3), 243–250.
- Rarey, M., Kramer, B. & Lengauer, T. (1999b). The Particle Concept: Placing Discrete Water Molecules During Protein-Ligand Docking Predictions. *Proteins: Structure, Function and Genetics*, **34**, 17–28.
- Rarey, M., Kramer, B., Lengauer, T. & Klebe, G. (1996a). A Fast Flexible Docking Method using an Incremental Construction Algorithm. *J Mol Biol*, **261**, 470–489.
- Rarey, M. & Lengauer, T. (2000). A recursive algorithm for efficient combinatorial library docking. *Perspect Drug Discov Design*, **20**, 63–81.
- Rarey, M. & Stahl, M. (2001). Similarity searching in large combinatorial chemistry spaces. *J Comput-Aided Mol Design*, **15**, 497–520.
- Rarey, M., Wefing, S. & Lengauer, T. (1996b). Placement of Medium Sized Molecular Fragments into Active Sites of Proteins. *J Comput-Aided Mol Design*, **10**, 41–54.
- Rarey, M. (original author) (2008). *flexX*. BioSolveIT, St. Augustin.
- Raymond, J.W. & Willet, P. (2002a). Effectiveness of graph-based and fingerprint-based similarity measures of virtual screening of 2D chemical structure databases. *J Comput-Aided Mol Design*, **16**, 59–71.
- Raymond, J.W. & Willet, P. (2002b). Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J Comput-Aided Mol Design*, **16**, 521–533.
- Raymond, J.W. & Willet, P. (2003). Similarity Searching in Databases of Flexible 3D Structures Using Smoothed Bounded Distance Matrices. *J Chem Inf Comput Sci*, **43**, 908–916.

- Rees, D.C., Congreve, M., Murray, C.W. & Carr, R. (2004). Fragment-based lead discovery. *Nat Rev Drug Discov*, **3**, 660–672.
- Renner, S., Hechenberger, M., Noeske, T., Bocker, A., Jatzke, C., Schmücker, M., Parsons, C. G., Weil, T. & Schneider, G. (2007). Searching for Drug Scaffolds with 3D Pharmacophores and Neural Network Ensembles. *Angew Chem Int Ed Engl*, **46** (28), 5336–5339.
- Reulecke, I., Lange, G., Albrecht, J., Klein, R. & Rarey, M. (2008). Towards an integrated description of hydrogen bonding and dehydration: decreasing false positives in virtual screening with the HYDE scoring function. *ChemMedChem*, **3** (6), 885–897.
- Reynolds, C.H., Bembenek, S.D. & Tounge, B.A. (2007). The role of molecular size in ligand efficiency. *Bioorg Med Chem Lett*, **17** (15), 4258–4261.
- Roe, D.C. & Kuntz, I.D. (1995). BUILDER v.2: Improving the chemistry of a de novo design strategy. *J Comput-Aided Mol Design*, **9**, 269–282.
- Rosania, G.R. & Chang, Y.-T. (2000). Targeting hyperproliferative disorders with cyclin dependent kinase inhibitors. *Exp Opin Ther Patents*, **10** (2), 215–230.
- Rotstein, S.H. & Murcko, M.A. (1993). GroupBuild: A Fragment-Based Method for De Novo Drug Design. *J Med Chem*, **36**, 1700–1710.
- Rouvray, D.H. (1995). Similarity in chemistry: Past, present and future. In: *Molecular Similarity I*, (Sen, K.D., Hrsg.), Bd. 173, S. 1–30. Springer-Verlag, Heidelberg.
- Rüegg, U.T. & Burgess, G.M. (1989). Staurosporine, K-252 and UCN-01: potent but nonspecific inhibitors of protein kinases. *Trends in Pharmacological Science*, **29**, 253–257.
- Sadowski, J. & Gasteiger, J. (1993). From Atoms and Bonds to Three-Dimensional Atomic Coordinates: Automatic Model Builders. *Chem Rev*, **93**, 2567–2581.
- Sadowski, J., Gasteiger, J. & Klebe, G. (1994). Comparison of Automatic Three-Dimensional Model Builders Using 639 X-Ray Structures. *J Chem Inf Comput Sci*, **34**, 1000–1008.
- Sanz, F., Giraldo, J. & Manaut, F., Hrsg. (1995). *QSAR and Molecular Modelling: Concepts, Computational Tools and Biological Applications*. J.R. Prous, Barcelona.
- Saporito, M.S., Hudkins, R.L. & Maroney, A.C. (2002). Discovery of CEP-1347/KT-7515, an inhibitor of the JNK/SAPK pathway for the treatment of neurodegenerative diseases. *Prog Med Chem*, **40**, 23–62.
- Schneider, G. (2002). Trends in Virtual Combinatorial Library Design. *Curr Med Chem*, **9**, 2095–2101.

- Schneider, G. & Böhm, H.-J. (2002). Virtual screening and fast automated docking methods. *Drug Discov Today*, **7** (1), 64–70.
- Schneider, G. & Fechner, U. (2005). Computer-Based De Novo Design of Drug-Like Molecules. *Nat Rev Drug Discov*, **4**, 649–663.
- Schneider, G., Lee, M.-L., Stahl, M. & Schneider, P. (2000). De novo design of molecular architectures by evolutionary assembly of drug-derived blocks. *J Comput-Aided Mol Design*, **14**, 487–494.
- Schneider, G., Neidhart, W., Giller, T. & Schmid, G. (1999). 'Scaffold-Hopping' by Topological Pharmacophore Search: A Contribution to Virtual Screening. *Angew Chem Int Ed Engl*, **38** (19), 2894–2896.
- Schnur, D.M., Hermsmeier, M.A. & Tebben, A.J. (2006). Are target-family-privileged substructures truly privileged? *J Med Chem*, **49** (6), 2000–2009.
- Schüller, A., Hähnke, V. & Schneider, G. (2007). SmiLib v2.0: A Java-Based Tool for Rapid Combinatorial Library Enumeration. *QSAR Comb Sci*, **26** (3), 407–410.
- Schüller, A., Suhartono, M., Fechner, U., Tanrikulu, Y., Breitung, S., Scheffer, U., Göbel, M.W. & Schneider, G. (2008). The concept of template-based de novo design from drug-derived molecular fragments and its application to TAR RNA. *J Comput Aided Mol Des*, **22** (2), 59–68.
- SciTegic (2007). *PipelinePilot*. SciTegic Inc., San Diego, CA, USA.
- Seifert, M.H.J. (2006). Assessing the discriminatory power of scoring functions for virtual screening. *J Chem Inf Model*, **46** (3), 1456–1465.
- Selassie, C.D., Fang, Z.-X., Li, R.-L., Hansch, C., Debnath, G., Klein, T.E., Langridge, R. & Kaufmann, B.T. (1989). On the Structure Selectivity Problem in Drug Design. A Comparative Study of Benzylpyrimidine Inhibition of Vertebrate and Bacterial Dihydrofolate Reductase via Molecular Graphics and Quantative Structure-Activity Relationships. *J Med Chem*, **32**, 1895–1905.
- Service, R.F. (2008). Chemistry. Click chemistry clicks along. *Science*, **320** (5878), 868–869.
- Shenkin, P.S., Yarmush, D.L., Fine, R.M., Wang, H. & Levinthal, C. (1987). Predicting antibody hypervariable loop conformation I. Ensembles of random conformations for ringlike structures. *Biopolymers*, **26**, 2053–2085.
- Sheridan, R.P. (2002). The Most Common Chemical Replacements in Drug-Like Compounds. *J Chem Inf Comput Sci*, **42**, 103–108.
- Sheridan, R.P. (2003). Finding Multiactivity Substructures by Mining Databases of Drug-Like Compounds. *J Chem Inf Comput Sci*, **43**, 1037–1050.

- Shewchuk, L., Hassell, A., Wisely, B., Rocque, W., Holmes, W., Veal, J. & Kuyper, L.F. (2000). Binding mode of the 4-anilinoquinazoline class of protein kinase inhibitor: X-ray crystallographic studies of 4-anilinoquinazolines bound to cyclin-dependent kinase 2 and p38 kinase. *J Med Chem*, **43** (1), 133–138.
- Siegel, M.G. & Vieth, M. (2007). Drugs in other drugs: a new look at drugs as fragments. *Drug Discov Today*, **12** (1-2), 71–79.
- Sielecki, T.M., Boylan, J.F., Benfield, P.A. & Trainor, G.L. (2000). Cyclin-Dependent Kinase Inhibitors: Useful Targets in Cell Cycle Regulation. *J Med Chem*, **43** (1), 1–18.
- Sippl, W. (2000). Receptor-based 3D QSAR analysis of estrogen receptor ligands. Merging the accuracy of receptor-based alignments with the computational efficiency of ligand-based methods. *J Comput-Aided Mol Design*, **14** (6), 559–572.
- Sneider, W. (1996). *Drug Prototypes and Their Exploitation*. Wiley & Sons, New York.
- Sousa, S.F., Fernandes, P.A. & Ramos, M.J. (2006). Protein-ligand docking: current status and future challenges. *Proteins*, **65** (1), 15–26.
- Sperandio, O., Andrieu, O., Miteva, M. A., Vo, M. Q., Souaille, M., Delfaud, F. & Villoutreix, B. O. (2007). MED-SuMoLig: A New Ligand-Based Screening Tool for Efficient Scaffold Hopping. *J Chem Inf Model*, **47** (3), 1097–1110.
- Stahl, M. & Böhm, H.-J. (1998). Development of filter functions for protein-ligand docking. *J Mol Graphics Modell*, **16**, 121–132.
- Stahl, M. & Rarey, M. (2001). Detailed Analysis of Scoring Functions for Virtual Screening. *J Med Chem*, **44**, 1035–1042.
- Stahl, M., Todorov, N.P., James, T., Mauser, H., Böhm, H.-J. & Dean, P.M. (2002). A validation study on the practical use of automated de novo design. *J Comput-Aided Mol Design*, **16**, 459–478.
- Tame, J.R.H. (1999). Scoring functions: A view from the bench. *J Comput-Aided Mol Design*, **13**, 99–108.
- Tame, J.R.H. (2005). Scoring functions—the first 100 years. *J Comput Aided Mol Des*, **19** (6), 445–451.
- Tanimoto, T.T. (1957). *IBM Internal Report 17. Nov.*
- Taylor, R.D., Jewsbury, P.J. & Essex, J.W. (2002). A review of protein-small molecule docking methods. *J Comput-Aided Mol Design*, **16**, 151–166.
- Teague, S.J., Davis, A.M., Leeson, P.D. & Oprea, T. (1999). The Design of Leadlike Combinatorial Libraries. *Angew Chem Int Ed Engl*, **38** (24), 3743–3748.

- Thompson, D.C., Denny, A.R., Nilakantan, R., Humblet, C, Joseph-McCarthy, D. & Feyfant, E. (2008). CONFIRM: connecting fragments found in receptor molecules. *J Comput Aided Mol Des*, **22**, 761–772.
- Thomson (2004). *World Drug Index (Version 2004)*. Philadelphia, PA, USA.
- Todorov, N.P., Buenemann, C.L. & Alberts, I.L. (2006). De novo ligand design to an ensemble of protein structures. *Proteins*, **64** (1), 43–59.
- Todorov, N.P. & Dean, P.M. (1997). Evaluation of a method for controlling molecular scaffold diversity in de novo ligand design. *J Comput-Aided Mol Design*, **11**, 175–192.
- Todorov, N.P. & Dean, P.M. (1998). A branch-and-bound method for optimal atom-type assignment in de novo ligand design. *J Comput-Aided Mol Design*, **12**, 335–349.
- Traxler, P. & Furet, P. (1999). Strategies toward the design of novel and selective protein tyrosine kinase inhibitors. *Pharmacol Ther*, **82**, 195–206.
- Tripes (2005). *Sybyl*. Tripes Inc., St Louis, MO, USA.
- Tripes (2008). *Tripes Mol2 File Format*. Tripes, L.P., St Louis, MO, USA.
- Tschinke, V. & Cohen, N.C. (1993). The NEWLEAD Program: A Method for the Design of Candidate Structures from Pharmacophoric Hypotheses. *J Med Chem*, **36**, 3863–3870.
- Turner, D.B., Willet, P., Ferguson, A.M. & Heritage, T.W. (1995). Similarity searching in files of three-dimensional structures: Evaluation of similarity coefficients and standardisation methods for field-based similarity searching. *SAR and QSAR in Environmental Research*, **3**, 101–130.
- Verdonk, M.L. & Rees, D.C. (2008). Group efficiency: a guideline for hits-to-leads chemistry. *ChemMedChem*, **3** (8), 1179–1180.
- Vieth, M., Siegel, M.G., Higgs, R.E., Watson, I.A., Robertson, D.H., Savin, K.A., Durst, G.L. & Hipkind, P.A. (2004). Characteristic Physical Properties and Structural Fragments of Marketed Oral Drugs. *J Med Chem*, **47**, 224–232.
- Vieth, M. & Sutherland, J. J. (2006). Dependence of molecular properties on proteomic family for marketed oral drugs. *J Med Chem*, **49** (12), 3451–3453.
- Vinkers, H.M., de Jonge, M.R., Daeyaert, F.F.D., Heeres, J., Koymans, L.M.H., van Lenthe, J.H., Lewi, P.J., Timmerman, H., Aken, K.V. & Janssen, P.A.J. (2003). SYNOPSIS: SYNthesize and OPTimize System in Silico. *J Med Chem*, **46**, 2765–2773.
- Walters, W.P., Stahl, M.T. & Murcko, M.A. (1998). Virtual screening - an overview. *Drug Discov Today*, **3** (4), 160–178.

- Wang, N., DeLisle, R. K. & Diller, D. J. (2005). Fast small molecule similarity searching with multiple alignment profiles of molecules represented in one-dimension. *J Med Chem*, **48** (22), 6980–6990.
- Wang, R., Lu, Y. & Wang, S. (2003). Comparative Evaluation of 11 Scoring Functions for Molecular Docking. *J Med Chem*, **46**, 2287–2303.
- Wang, R., Ying, G. & Lai, L. (2000). LigBuilder: a multi-purpose program for structure-based drug design. *J Mol Model*, **6** (7-8), 498–516.
- Warren, G. L., Andrews, C. W., Capelli, A. M., Clarke, B., LaLonde, J., Lambert, M. H., Lindvall, M., Nevins, N., Semus, S. F., Senger, S., Tedesco, G., Wall, I. D., Woolven, J. M., Peishoff, C. E. & Head, M. S. (2006). A critical assessment of docking programs and scoring functions. *J Med Chem*, **49** (20), 5912–5931.
- Waszkowycz, B. (2002). Structure-based approaches to drug design and virtual screening. *Current Opinion in Drug Discovery and Development*, **5** (3), 407–413.
- Waszkowycz, B., Clark, D. E., Frenkel, D., Li, J., Murray, C. W., Robson, B. & Westhead, D. R. (1994). PRO_LIGAND: an approach to de novo molecular design. 2. Design of novel molecules from molecular field analysis (MFA) models and pharmacophores. *J Med Chem*, **37** (23), 3994–4002.
- van de Waterbeemd, H., Hrsg. (1995). *Chemometric Methods in Molecular Design*, Bd. 2 d. Reihe *Methods and Principles in Medicinal Chemistry*. VCh Publishers, Weinheim.
- van de Waterbeemd, H. & Gifford, E. (2003). ADMET in Silico Modelling: Towards Prediction Paradise? *Nat Rev Drug Discov*, **2**, 192–204.
- Weber, L. (2005). Current status of virtual combinatorial library design. *QSAR Comb Sci*, **24** (7), 809–823.
- Weininger, D. Weininger, A. & Weininger, J.L. (1989). SMILES. 2. Algorithm for generation of unique SMILES notation. *J Chem Inf Comput Sci*, **29**, 97–101.
- Weininger, D. (1988). SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J Chem Inf Comput Sci*, **28**, 31–36.
- Wester, M.J., Pollock, S.N., Coutsiias, E.A., Allu, T.K., Muresan, S & Oprea, T.I. (2008). Scaffold Topologies. 2. Analysis of Chemical Databases. *J Chem Inf Model*, **48** (7), 1311–1324.
- Westhead, D.R., Clark, D.E., Frenkel, D., Li, J., Murray, C.W., Robson, B. & Waszkowycz, B. (1995). PRO_LIGAND: An approach to de novo molecular design. 3. A genetic algorithm for structure refinement. *J Comput-Aided Mol Design*, **9**, 139–148.
- Willett, P. (1995). Searching for Pharmacophoric Patterns in Databases of Three-Dimensional Chemical Structure. *J Mol Recognition*, **8**, 290–303.

- Willett, P. (1996). Molecular diversity techniques for chemical databases. *Information Research*, **2** (3).
- Willett, P. (2006). Enhancing the effectiveness of ligand-based virtual screening using data fusion. *QSAR Comb Sci*, **25** (12), 1143–1152.
- Zaliani, A. (2007). personal communication.
- Zhu, J., Fan, H., Liu, H. & Shi, Y. (2001a). Structure-based ligand design for flexible proteins: application of new F-DycoBlock. *J Comput Aided Mol Des*, **15** (11), 979–996.
- Zhu, J., Yu, H., Fan, H., Liu, H. & Shi, Y. (2001b). Design of new selective inhibitors of cyclooxygenase-2 by dynamic assembly of molecular building blocks. *J Comput Aided Mol Des*, **15** (5), 447–463.
- Zhu, X., Kim, J.L., Newcomb, J.R., Rose, P.E., Stover, D.R., Toledo, L.M., Zhao, H. & Morgenstern, K.A. (1999). Structural analysis of the lymphocyte-specific kinase Lck in complex with non-selective and Src family selective kinase inhibitors. *Structure*, **7** (6), 651–661.
- ZINC (2006). <http://zinc.docking.org/subset1/3/index.html> (Version vom 2. Mai 2006).

Hilfsmittel

Für die Erstellung dieser Arbeit wurden die folgenden Hilfsmittel verwendet:

- Das im Rahmen dieser Arbeit entwickelte Software-Werkzeug FlexNovo basiert auf der Programmbibliothek des molekularen Docking-Programms FlexX [Rarey *et al.*, 1996a] und dessen Erweiterung zur Berücksichtigung von Pharmakophoren FlexX-Pharm [Hindle *et al.*, 2002]. Die Entwicklung von FlexNovo erfolgte in ANSI C mit dem GNU C/C++ Compiler (Versionen 3.1.1 bis 3.4.6) auf einem Intel-Workstation Computer (Pentium-XEON Prozessor, 3.0 GHz) unter SUSE Linux (Versionen 8.0 bis 10.2).
- Zur Aufbereitung der verwendeten Kristallstrukturen wurde das Programm Sybyl [Tripos, 2005] (Version 6.9) verwendet.
- Zur Fragmentierung der Molekül-Kataloge wurde das Programm Recore [Maaß *et al.*, 2007] (Version 1.8.1) und für die Aufbereitung der Eingabedaten PipelinePilot [SciTegic, 2007] (Version 6.1) verwendet.
- Zur Erstellung der in dieser Arbeit enthaltenen dreidimensionalen Abbildungen der Ergebnisse wurde das Programm PyMOL [DeLano, 2006] (Version 0.99rc6) verwendet. Zur Erstellung der zweidimensionalen Moleküldarstellungen wurde das Programm ChemDraw [CambridgeSoft, 2005] (Version 10.0) verwendet.
- Zur Erzeugung der Histogramme und Diagramme wurden die Programme gnuplot [gnuplot, 2008] (Version 4.0) und Grace [Grace, 2008] (Version 5.1.19) sowie Excel [Microsoft, 2004] (Versionen 2002 und 2004) verwendet.
- Sämtliche Berechnungen wurden auf Intel-Workstation Computern (Pentium-XEON Prozessor, 3.0 GHz) unter SUSE Linux (Versionen 8.0 bis 10.2) durchgeführt.
- Die vorliegende Arbeit wurde mit dem Textsatzpaket $T_{E}X$ und der Makroerweiterung $L_{A}T_{E}X 2_{\epsilon}$ unter Zuhilfenahme des KOMA-Script Pakets erstellt. Zur Literaturverwaltung wurde BibDesk [BibDesk, 2008] (Version 1.3.18) in Verbindung mit $BIB_{T}E_{X}$ verwendet.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich meine Dissertation „Entwicklung eines strukturbasierten de-novo-Design-Verfahrens und chemischer Fragmenträume für den rationalen Wirkstoffentwurf“ selbständig, ohne unerlaubte Hilfe angefertigt und mich dabei keiner anderen als der von mir ausdrücklich bezeichneten Quellen bedient habe.

Die Dissertation wurde in der jetzigen oder einer ähnlichen Form noch bei keiner anderen Hochschule eingereicht und hat noch keinen sonstigen Prüfungszwecken gedient.

Hamburg, den 15.08.2008

(Jörg Degen)

Lebenslauf

	Jörg Degen
30.09.1976	geboren in Reutlingen

Sep. 1987 – Juli 1993	Nikolaus-August-Otto Gesamtschule in Bad Schönborn
Sep. 1993 – Juni 1996	Carl-von-Ossietzky Oberstufengymnasium in Wiesbaden
Juni 1996	Allgemeine Hochschulreife
Okt. 1996	Immatrikulation im Diplom-Studiengang Chemie an der Technischen Universität Darmstadt
Dez. 2001	Diplom-Prüfung
Jan. – Juli 2002	Anfertigung der Diplomarbeit unter Betreuung von Prof. Dr. P.C. Schmidt am Institut für Physikalische Chemie der Technischen Universität Darmstadt: „Implementierung der Elektronenlokalisierungsfunktion in das FP-LMTO-Programm NFP und Tests an ausgewählten Systemen“
Juli 2002	Abschluss als Diplom-Ingenieur der Fachrichtung Chemie
Dez. 2002 – Aug. 2008	Anfertigung der Promotionsarbeit unter Betreuung von Prof. Dr. M. Rarey am Zentrum für Bioinformatik der Universität Hamburg
April 2006	Keystone Symposia Stipendium Keystone Symposium on Structure Based Drug Discovery, Whistler, British-Columbia, Kanada
Nov. 2007	Posterpreis 3rd German Conference on Chemoinformatics, Goslar

Dez. 2002 – März 2008	Wissenschaftlicher Mitarbeiter am Zentrum für Bioinformatik der Universität Hamburg
Sep. 2004 – März 2008	Mitarbeiter im Projekt NovoBench, das im Rahmen des BMBF-Programms BioChance+ gefördert wurde.
Aug. 2008 – Okt. 2008	Mitarbeiter im Konsortium „Combating Drug Resistance in Chronic Myeloid Leukemia and HIV-1 Infection“, das im Rahmen des BMBF-Programms „Innovative Therapieverfahren auf molekularer und zellulärer Basis“ gefördert wird.
