

**Entwicklung eines Protein-Ligand-Dockingprogrammes auf
Basis populationsbasierter Algorithmen**

Dissertation

zur Erlangung des akademischen Grades

Doctor rerum naturalium (Dr. rer. nat.)

vorgelegt der

Naturwissenschaftlichen Fakultät I
Biowissenschaften

der Martin-Luther-Universität Halle-Wittenberg

von

Herrn Dipl.-Biochem. René Meier
geb. 15. November 1977 in Meißen

Gutachter:

1. Prof. Dr. Wolfgang Sippl, Halle/Saale
2. Prof. Dr. Ivo Große, Halle/Saale
3. Prof. Dr. Martin Zacharias, Bremen

Halle(Saale), 18.05.2009

Die vorliegende Arbeit wäre nicht möglich gewesen, ohne die gute und intensive Zusammenarbeit mit Prof. Dr. Wolfgang Sippl. Ihm möchte ich herzlich für die Möglichkeit danken, diese Arbeit anfertigen zu können.

Mein Dank geht ebenfalls an die beiden Gutachter Prof. Dr. Ivo Große und Prof. Dr. Martin Zacharias für die freundliche Begutachtung dieser Arbeit.

Dr. Carsten Baldauf bin ich für alle Diskussionen und Anregungen danken.

Für die gute und freundliche Arbeitsatmosphäre möchte ich allen Mitarbeitern der Abteilung pharmazeutische Chemie danken.

Besonderer Dank geht an Lena für die Unterstützung bei der schriftlichen Ausarbeitung.

Ohne Birgit wäre diese Arbeit sicher nie zu einem glücklichen Ende gekommen. Ihr möchte ich für ihre Geduld, Hilfe und ihre Motivation danken. Meinen Eltern möchte ich für die immerwährende Unterstützung von Beginn des Studiums bis zum Ende dieser Arbeit danken.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Bestehende Ansätze für das Moleküldocking	9
1.2	Optimieralgorithmen	11
1.3	Bewertungsfunktionen	19
1.4	Zielsetzung	24
2	Material und Methoden	27
2.1	Datensätze	27
2.1.1	<i>Astex Diverse Set</i>	27
2.1.2	<i>PDBbind</i>	28
2.1.3	Estrogenrezeptor	29
2.1.4	Acetylcholinesterase	31
2.1.5	Protein-Arginin Methyltransferase 1	33
2.2	Bibliotheken und Software	33
2.2.1	Xerces-C++	33
2.2.2	OpenSceneGraph	34
2.2.3	OpenMPI	35
2.2.4	Boost C++ Libraries	36
2.3	RMSD-Wert Bestimmung kleiner Moleküle	37
2.4	Quaternionen	40
2.5	X-SCORE	42
3	Ergebnisse und Diskussion	47
3.1	Design des Dockingprogrammes	47
3.1.1	XML-Parser für die Konfigurationsdatei	50
3.1.2	Molekülgraph	52
3.1.3	Metaheuristik/PSO	54
3.1.4	Fitnessfunktion/p-Score	58

Inhaltsverzeichnis

3.2	Parametrisierung	65
3.2.1	Wahl der Parameter des PSO	66
3.2.2	Parametrisierung von p-Score	73
3.3	Genauigkeit des Dockingprogrammes	76
3.4	Eignung für VS-Experimente	81
4	Zusammenfassung und Ausblick	85
A	Datensätze	89
A.1	<i>Astex Diverse Set</i>	89
A.2	<i>PDBbind core set</i>	92
	Literaturverzeichnis	99

1 Einleitung

Wirkstoffentwicklung für ein neues Medikament ist heutzutage ein langwieriger und teurer Prozess. Für die Gruppe der neuen, innovativen Wirkstoffe gehen Studien von einer Entwicklungszeit von bis zu zwölf Jahren und Kosten zwischen 0,5 und 2 Mrd. Dollar bis zur Zulassung des Medikaments aus [1, 2]. Bei den gesuchten Wirkstoffen handelt es sich um oral gut bioverfügbare Moleküle, die als Agonist oder Antagonist an den für die Krankheit relevanten Proteinen wirken und so den gewünschten therapeutischen Effekt erzielen. Diese Moleküle werden auch als Effektoren oder Liganden des Proteins bezeichnet.

Zu Beginn der Wirkstoffentwicklung sind Leitstrukturen gesucht. Leitstrukturen sind Moleküle, die die gewünschte Affinität zum Zielprotein besitzen, aber noch Raum für die Optimierung pharmakodynamischer und pharmakokinetischer Eigenschaften lassen. Um das Auffinden neuer Leitstrukturen möglichst effizient zu gestalten, werden in der systematischen Suche robotergestützte Synthese- und Analysesysteme eingesetzt. Ab einem Durchsatz von mehr als 10000 Proben pro Tag wird von *high throughput screening* (HTS) gesprochen. Obwohl diese Systeme mit geringem personellen Aufwand in kurzer Zeit große Substanzbibliotheken durchsuchen können, ist der finanzielle Aufwand erheblich. Sowohl die Synthese als auch die Testung benötigen Chemikalien, Verbrauchsmaterialien oder biologisches Material.

Im Gegensatz zum kostenintensiven HTS etablierten sich in den späten 1990er Jahren verschiedene Methoden, die unter dem Begriff *virtual screening* (VS) zusammengefasst werden. Diese Methoden werden benutzt, um noch vor den ersten experimentellen Tests die Anzahl der Kandidaten gezielt und sinnvoll einzuschränken [3, 4], und so den finanziellen Aufwand zu reduzieren. Zum Einsatz kommen dabei computergestützte Methoden, die aus einer Vielzahl von Verbindungen erfolgversprechende Kandidaten auswählen. Je nach experimenteller Datenlage wird dabei zwischen zwei verschiedenen Vorgehens-

1 Einleitung

weisen unterschieden.

Zum einen werden ligandbasierte VS-Methoden durch Substanzbibliotheken mit heterogenen Aktivitätsdaten ermöglicht. Zu dieser Gruppe gehören die Suche mit Pharmakophorhypothesen, quantitative Struktur-Aktivitäts-Beziehung (QSAR), *Comparative Molecular Field Analysis* (CoMFA) [5] und einfache Ähnlichkeitsansätze. Diese Methoden vergleichen aktive und inaktive Liganden anhand verschiedener Ähnlichkeitskriterien. Die so gewonnenen Informationen werden nachfolgend eingesetzt, um unbekannte Moleküle zu klassifizieren. Die Klassifizierung kann quantitativ (Aktivitätsvorhersage) oder qualitativ (Binder oder Nichtbinder) erfolgen.

Andererseits ermöglicht das Vorhandensein von Strukturdaten des Zielproteins strukturbasiertes VS. Diese Strukturdaten können entweder durch experimentelle Methoden wie Röntgenkristallstrukturanalyse und NMR oder durch Homologiemodellierung gewonnen werden. Gemeinsam ist allen Methoden die Vorhersage der Aktivität anhand einer dreidimensionalen (3D) Struktur des Komplexes aus Kandidatenmolekül und Zielprotein. Dazu werden mathematische Modelle benutzt, die die räumlichen und energetischen Kriterien der Bindung beschreiben. Deshalb ist strukturbasiertes VS ein wissenschaftlicher Ansatz im Gegensatz zu dem rein phänomenologischen Vorgehen beim HTS. Der Erfolg eines VS hängt direkt mit der Qualität und Quantität der Information über das Zielprotein und die Bindungsvorgänge zusammen. Trotz dieser Anforderungen hat sich das VS als Bestandteil der Wirkstoffforschung etabliert und es wurden bereits für mindestens 50 Zielproteine Liganden mit mikromolarer Bindungsaktivität erfolgreich vorhergesagt [4].

Die für das strukturbasierte VS benötigten 3D-Strukturen der Komplexe aus Zielprotein und Kandidatenmolekül können mit verschiedenen Methoden erzeugt werden. Die Überlagerung pharmakophorer Eigenschaften der Liganden in der Bindetasche des Proteins stellt eine Möglichkeit dar. Eine weitere Möglichkeit ist das Moleküldocking. Mit Moleküldocking werden Methoden zur Vorhersage von Komplexen aus zwei oder mehr Molekülen bezeichnet. Die Orientierung zweier Moleküle zueinander resultiert in drei rotatorische und drei kartesische Dimensionen. Zusätzlich treten noch weitere Freiheitsgrade für die Konformationen der Liganden auf. Die Kombination dieser Freiheitsgrade führt zu einer erheblichen Anzahl an Möglichkeiten. Die

meisten Dockingalgorithmen erzeugen deshalb eine große Anzahl an möglichen Strukturen und bewerten diese. Moleküldocking befasst sich somit mit der Erzeugung und Bewertung von Molekülkomplexen. Im strukturbasierten Wirkstoffdesign werden Dockingmethoden eingesetzt, um die Bindungsposition von Proteininhibitoren vorherzusagen. Dies macht Moleküldocking zu einer wichtigen Methode für strukturbasiertes VS.

1.1 Bestehende Ansätze für das Moleküldocking

Eines der ersten Dockingprogramme wurde im Jahr 1992 veröffentlicht und heißt DOCK [6]. Die Entwicklung der Algorithmen zur Vorhersage von Makromolekül-Ligand-Komplexen reicht noch mindestens 10 Jahre weiter zurück [7]. Der verwendete Algorithmus benutzte rigide Liganden und Proteine und bewertete den Komplex bezüglich der Oberflächen-Komplementarität.

Aus algorithmischer Sicht setzt sich das Dockingproblem aus zwei komplexen Bestandteilen zusammen. Zum Ersten handelt es sich bei der Erzeugung der möglichen Protein-Ligand-Komplexe (Pose) um ein kombinatorisches Problem mit exponentieller Komplexität. Diese Art von Problemen lässt sich in der Praxis kaum durch exakte Algorithmen lösen, da die Eingabedimension der Probleme in einer zu großen Anzahl an Möglichkeiten resultiert. Das zweite Problem stellt die Bewertung der Posen dar. Die verwendeten Bewertungsfunktionen sind vereinfachte Modelle und bilden die Realität näherungsweise ab. Die in den letzten Jahren veröffentlichten Dockingalgorithmen lassen sich in drei Klassen einteilen. Es gibt fragmentbasierte Methoden, heuristische Methoden und Multikonformer-Methoden. Diese Ansätze unterscheiden sich hinsichtlich des Algorithmus zur Erzeugung der Posen.

Im Fall der fragmentbasierten Ansätze wird der Ligand in rigide Fragmente zerlegt. Das größte Fragment wird Basisfragment genannt. Dieses Basisfragment wird in der Bindetasche des Proteins platziert und die besten Lösungen werden als Ausgangspunkt für einen sukzessiven Aufbau des Liganden aus seinen Fragmenten benutzt. Kriterium für die Güte der erzeugten Zwischenergebnisse stellt die Bewertungsfunktion dar. Zu dieser Gruppe gehören die Programme der FlexX-Familie [8], Surflex [9] und eHiTS [10]. Vorteil dieser Methode ist die Vereinfachung des kombinatorischen Suchproblems auf ein

1 Einleitung

lineares Suchproblem. Falls jedoch das Basisfragment falsch platziert wird, kann die fragmentbasierte Methode keine sinnvollen Lösungen finden.

Die zweite Klasse von Dockingalgorithmen beruht auf heuristischen Optimierungsverfahren. Heuristische Optimierungsverfahren können dann eingesetzt werden, wenn nicht ausschließlich die exakt richtige Lösung von Interesse ist, sondern eine sehr gute Lösung ausreichend ist. Diese Methoden versuchen das globale Optimum der Bewertungsfunktion zu finden und gehen davon aus, dass dieses Optimum mit dem tatsächlichen Komplex übereinstimmt. Der Suchraum des Algorithmus wird definiert durch die Freiheitsgrade von Ligand und Protein. Der in dieser Arbeit vorgestellte Dockingalgorithmus auf Basis eines Partikel-Schwarm Optimierer (PSO) [11] gehört zur Gruppe der heuristischen Optimierungsverfahren. Manche Programme nutzen problemspezifische Heuristiken wie z. B. Glide [12, 13]. In einem mehrstufigen Optimierungsprozess werden bei Glide spezielle Filter auf die Zwischenlösungen angewendet und im letzten Schritt mit einer Monte-Carlo-Simulation nach dem Endergebnis gesucht. Andere Programme benutzen universelle Optimierungsalgorithmen, die sich schon bei anderen hochdimensionalen, kombinatorischen Problemen als erfolgreich erwiesen haben. Die Programme GOLD (*Genetic Optimisation for Ligand Docking*) [14], AutoDock [15] und DARWIN [16] benutzen einen genetischen Algorithmus (GA). Ein GA ist ein populationsbasierter Optimierungsalgorithmus, der durch die evolutionären Prozesse Vererbung, Mutation, Rekombination und Selektion inspiriert wurde. Eine weitere erfolgreich angewandte populationsbasierte Optimierungsmethode ist der PSO in den Programmen SODOCK [17] und AutoDock mit ClustMPSO [18]. In PLANTS [19] wurde der Ameisenalgorithmus zur Lösung des Dockingproblems eingesetzt.

Die dritte Klasse der Dockingalgorithmen führen ein rigides Docking eines Ensembles von Ligandenkonformationen durch. Vertreter dieser Gruppe sind FRED (Fast Rigid Exhaustive Docking) [20] und DOCK, obwohl DOCK inzwischen auch flexibles Docking unterstützt. Vorteil dieser Methoden ist die hohe Geschwindigkeit mit der die rigiden Liganden gedockt werden. Die hohe Geschwindigkeit ergibt sich aus der geringeren Anzahl von Freiheitsgraden. Ein deutlicher Nachteil dieser Methode ist, dass die bioaktive Konformation in dem Ensemble von Ligandenkonformationen enthalten sein muss. Die

bioaktive Konformation muss jedoch nicht mit einer energetisch günstigen Konformation übereinstimmen und ist darum in manchen Fällen schwer zu identifizieren.

Von den vorgestellten Programmen sind nur DOCK und AutoDock als Quellcode verfügbar und nur für AutoDock ist eine Erweiterung oder Änderung des Programmes erlaubt (GPL seit Mitte 2007). Die anderen Programme sind zum Teil kostenlos für akademische Forschungsprojekte, z.B. DOCK, PLANTS oder eHiTS. Ein Großteil der Programme ist jedoch kostenpflichtig und für den Nutzer nicht transparent. Rein praktisch stellt sich die Erweiterung und Veränderung von AutoDock als kompliziert dar, da der Quellcode von AutoDock nur wenig dokumentiert ist und die gesamte Software als ein monolithisches System entworfen ist.

1.2 Optimieralgorithmen

Viele der Probleme, die auf dem Gebiet des computergestützten Wirkstoffdesigns zu lösen sind, gehören zur Gruppe der Optimierungsprobleme. Gesucht ist dabei entweder ein Minimum oder ein Maximum. Als Zielfunktion wird die zu optimierende Größe bezeichnet und als Parameter oder Variablen werden die zu variierenden Größen bezeichnet. Für die Oberfläche der Zielfunktion wird der Begriff Fitnesslandschaft verwendet. Besteht die Optimierung darin, von einem Startparameter ausgehend das nächste relative Optimum zu finden, ist es eine lokale Optimierung. Ist das absolute Optimum gesucht, ist es eine globale Optimierung. Als Nebenbedingungen können zusätzlich noch Gleichungen, Ungleichungen oder zulässige Mengen (z. B. nur ganzzahlige Parameter) für die Parameter definiert sein. Formal lässt sich eine Instanz I dieses Problem durch $I = (P, f, N)$ beschreiben, wobei P die möglichen Parameter darstellt. Die Zielfunktion $f : P \rightarrow \mathbb{R}$ weist jedem Parameter $p \in P$ einen Wert $f(p) \in \mathbb{R}$ zu. Die Nebenbedingung N definiert die Menge $P_N \subseteq P$ aller möglichen Parameter. Ziel der Optimierung ist es, den Parameter p^* zu finden, für den für alle anderen $p \in P_N$ im Falle einer Minimierung $f(p^*) \leq f(p)$ gilt. Im Falle einer Maximierung ist der Parameter p^* , für den $f(p^*) \geq f(p)$ gilt, von Interesse.

Alle hier vorgestellten Methoden zur nichtlinearen Optimierung sind iterative

1 Einleitung

Näherungsverfahren. Für lokale Optimierungsprobleme steht eine Reihe von Methoden zur Verfügung, die relativ robust ein nahgelegenes Optimum finden können. Der *hill climbing*-Algorithmus wählt, ausgehend von einer Startposition, solange eine bessere benachbarte Lösung bis keine weitere gefunden wird. Eine effektivere Variante dieser Methode ist das Downhill-Simplex-Verfahren [21]. Dabei wird im N-dimensionalen Raum ein Volumen mit $N+1$ Ecken (Simplex) aufgespannt und dieses Volumen wird so durch den Raum bewegt, dass immer die schlechteste der Ecken durch eine neue Ecke ersetzt wird. Die neue Ecke wird erzeugt, indem die schlechteste Ecke am Zentrum der verbleibenden Ecken gespiegelt wird. Obwohl bei dieser Methode kein echter Gradient durch die erste Ableitung der Zielfunktion berechnet wird, kann durch den Simplex eine Richtung abgeschätzt werden, in der nach besseren Lösungen gesucht werden sollte. Es existieren noch zahlreiche weitere Algorithmen zur lokalen Suche, die sich dahingehend unterscheiden, wie eine benachbarte Lösung erzeugt wird und unter welchen Bedingungen die benachbarte Lösung vom Algorithmus akzeptiert wird. Falls die Zielfunktion differenzierbar ist, können gradientenbasierte Verfahren eingesetzt werden, die in der Regel mit weniger Iterationen die Lösung finden. Da die Differenzierbarkeit der Zielfunktion im Falle von Moleküldocking Methoden oft nicht gegeben ist, werden diese Methoden hier nicht näher betrachtet.

Von größerer Bedeutung für das Dockingproblem ist die globale Optimierung. Diese muss in der Lage sein, lokale Optima zu überwinden, um das globale Optimum zu finden, und wird unter dem Begriff Metaheuristik zusammengefasst. Alle Algorithmen zur globalen Optimierung müssen ein ausgewogenes Verhältnis zwischen Diversifikation (globalen Suche) und Intensivierung (lokale Suche) aufweisen. Einfachster Ansatz zur globalen Optimierung ist die Wiederholung einer lokaler Optimierung, z. B. der *hill climbing*-Algorithmus, mit unterschiedlichen Startbedingungen. Eine Erweiterung dieses Ansatzes sind Algorithmen, die temporär auch schlechtere Lösungen akzeptieren und somit ein lokales Optimum überwinden können. Ein Vertreter dieser Gruppe von Algorithmen, die als Nachbarschaftssuche bezeichnet werden können, ist das *simulated annealing* (SA) [22]. Inspiriert wurde dieser Algorithmus von Methoden in der Metallurgie. Metalle werden langsam erhitzt und abgekühlt, um dem Material Zeit zu geben, gleichmäßige und span-

nungsarme Kristallgitter zu bilden. Während Phasen hoher Temperatur können die Atome auch energetisch ungünstige Zustände einnehmen und somit Spannungen lösen. Algorithmus 1 stellt den Pseudocode für das SA-Verfahren dar. Beim SA wird mit einer beliebigen Lösung p_0 gestartet. Außerdem

Algorithmus 1 *simulated annealing*

```

 $p \leftarrow \text{ZufallsStartZustand}()$ 
 $w \leftarrow f(p_0)$ 
 $T \leftarrow \text{StartTemperatur}$ 
while Abbruchkriterium nicht erfüllt und  $T > 0$  do
   $p^* \leftarrow \text{Nachbarschaft}(p)$ 
   $w^* \leftarrow f(p^*)$ 
   $A \leftarrow \exp\left(\frac{w-w^*}{T}\right)$ 
  if  $A \geq \text{Zufall}(0, 1)$  then
     $p \leftarrow p^*$ 
     $w \leftarrow w^*$ 
  end if
   $T \leftarrow \text{NächsteTemperatur}()$ 
end while
return  $p$ 

```

muss eine Temperaturabfolge definiert sein, die die Erhitzungs- und Abkühlungsperioden des Algorithmus festlegt und mit $T_N=0$ endet. p_i stellt den aktuellen Zustand des Systems dar. Ausgehend vom p_i wird in dessen Nachbarschaft ein beliebiges p^* gewählt. p^* ist eine mögliche Lösung der Zielfunktion und stellt einen möglichen neuen Zustand des Systems dar. Ob das System vom Zustand p_i in den neuen Zustand p^* übergeht wird durch die Akzeptanzwahrscheinlichkeits-Funktion $A(w, w^*, T)$ bestimmt. Dabei ist $w = f(p_i)$, $w^* = f(p^*)$ und T_i die sich stetig verändernde Temperatur. Im Fall einer Minimierung könnte $A = \exp\left(\frac{w-w^*}{T_i}\right)$ sein. Wenn $w^* < w$ ist, dann ist $A > 1$ und somit wird der neue Zustand vom System immer akzeptiert. Wenn $w^* > w$ ist, dann sinkt mit sinkender Temperatur auch die Akzeptanzwahrscheinlichkeit des neuen Zustands. Falls der neue Zustand vom System akzeptiert wird, ergibt sich $p_{i+1} = p^*$ und die nächste Iteration beginnt. Wird der neue Zustand nicht akzeptiert, ist $p_{i+1} = p_i$. Dieser Vorgang wird solange fortgesetzt, bis eine bestimmte Anzahl an Schritten durchgeführt wurde oder ein anderes Abbruchkriterium erreicht wurde. Auch andere Formen von A sind denkbar.

1 Einleitung

Allerdings muss immer gegeben sein, dass mit sinkender Temperatur die Akzeptanzwahrscheinlichkeit schlechterer Zustände abnimmt. Es existiert noch eine Anzahl weiterer Algorithmen, die auf einer Nachbarschaftssuche basieren, z. B. *taboo search* [23, 24], *quantum annealing* [25] und der Schwellenakzeptanz-Algorithmus [26].

Eine andere Gruppe von Methoden zur globalen Optimierung sind populationsbasierte Algorithmen. Eine mögliche Lösung $p_i \in P$ wird bei dieser Art von Algorithmen Individuum I genannt. Alle Individuen bilden eine Population, um das Optimum einer gegebenen Zielfunktion zu finden. Die Position oder die Eigenschaften jedes Individuums werden in Abhängigkeit von der eigenen Qualität (Fitness) und der Position oder den Eigenschaften und der Qualität der anderen Individuen in einem iterativen Prozess solange geändert, bis ein spezielles Abbruchkriterium erfüllt ist. Eine Untergruppe der populationsbasierten Algorithmen bilden die Evolutionären Algorithmen (EA). Sie wurden inspiriert von Vorgängen während der Evolution und beruhen auf den biologischen Prinzipien der Vererbung, der Mutation, der Rekombination und der Selektion. Ein Vertreter dieser Klasse ist der GA. Von großer Wichtigkeit bei der Anwendung eines GA ist, wie eine potenzielle Lösung (Phänotyp) abstrahiert als Chromosom (Genotyp) dargestellt wird. Das Chromosom muss die genetischen Operatoren Rekombination und Mutation sinnvoll implementieren und die Position in der Fitnesslandschaft muss durch das Chromosom sinnvoll dargestellt werden. Algorithmus 2 repräsentiert einen GA im Pseudocode. Zu Beginn wird eine Population P mit zufälligen Individuen erzeugt. Die Fitness eines jeden Individuums wird durch die Zielfunktion bestimmt. Dann erfolgt die iterative Optimierung (Evolution). Anhand der Fitness wird ein Teil der Population zur Replikation ausgewählt. Dabei werden Individuen mit besserer Fitness bevorzugt ausgewählt. Im Replikationsschritt werden durch die Anwendung genetischer Operatoren auf die ausgewählten Individuen Nachkommen erzeugt. Die Mutation erzeugt aus einem Eltern-Individuum einen Nachfahren, der an einer oder mehreren Stellen im Chromosom zufällig verändert ist. Die Rekombination erzeugt aus zwei Eltern-Individuen einen Nachfahren dessen Chromosom aus Teilen der Eltern-Chromosomen zusammengesetzt ist. Rekombination und Mutation können alternativ oder nacheinander angewendet werden. Anschließend wird die Fitness

Algorithmus 2 genetischer Algorithmus

```

for j=1 bis Anzahl der Individuen do
   $I \leftarrow \text{ZufallsStartZustand}()$ 
   $F(j) \leftarrow f(I)$ 
   $P(j) \leftarrow I$ 
end for
while Abbruchkriterium nicht erfüllt do
   $P \leftarrow \text{Selektion}(P, F)$ 
   $P \leftarrow \text{Replikation}(P)$ 
  for j=1 bis Anzahl der Individuen do
     $F(j) \leftarrow f(P(j))$  {Evaluation}
  end for
end while
return  $\text{BestesIndividuum}(P)$ 

```

der Nachfahren evaluiert und der nächste Schritt der Evolution beginnt. Das fitteste Individuum ist das Ergebnis der Optimierung. Erste Arbeiten auf dem Gebiet der künstlichen Evolution wurden schon in den 50er Jahren durchgeführt [27]. In den 80er Jahren entwickelte sich der GA zu einer universell einsetzbaren Optimierungsmethode und wurde für viele Probleme erfolgreich eingesetzt. Da der GA eine so große Verbreitung fand wurden auch sehr viele Varianten entwickelt. In der ursprünglichen Form wird für jede Position im Chromosom nur eine geringe Anzahl von Zuständen (Alphabet) zugelassen. Somit können prinzipbedingt nur diskrete Probleme optimiert werden. Es wurden jedoch auch Möglichkeiten zur Optimierung mit reellen Zahlen vorgestellt [28]. Eine weitere Variante ist der Lamarckian GA [29]. Diese Variante führt vor der Selektion einen Adaptionsschritt ein. Realisiert wird dies durch eine beliebige lokale Optimierung. Das Ergebnis der Adaption wird vom Phänotyp in den Genotyp übertragen. Das steht im Gegensatz zur echten Evolution, ist aber bei künstlicher Evolution im Computer möglich. Andere Varianten beschäftigen sich mit verschiedenen Selektionsverfahren oder der Teilung der Population in Teilpopulationen.

In die Gruppe der populationsbasierten Optimierungsalgorithmen gehören neben den evolutionären Algorithmen auch Schwarmintelligenz modellierende Algorithmen. Als Vertreter können der Ameisenalgorithmus (*ant colony optimization* – ACO) [30] und PSO genannt werden. ACO optimiert den Pfad zwischen

1 Einleitung

einem virtuellen Ameisennest und einer Futterquelle. Ameisen hinterlassen auf einem Pfad den sie begehen Pheromonmarkierungen. Hat eine Ameise mehrere Möglichkeiten so geht sie mit höherer Wahrscheinlichkeit den Weg mit höherer Pheromonkonzentration. Wenn eine bestimmte Wegalternative kürzer als eine andere ist, wird dieser Weg im selben Zeitintervall von mehr Ameisen benutzt, und hat somit eine höhere Pheromonkonzentration. Dies führt dazu, dass noch mehr Ameisen diesen Pfad benutzen. Wird ein Weg über eine gewisse Zeit nicht benutzt, verdunsten die vorhanden Pheromone und es wird unwahrscheinlicher, dass dieser Weg von weiteren Ameisen benutzt wird. Am Ende benutzt ein Großteil der Ameisenkolonie den kurzen Weg. Die Kolonie hat somit den Weg zur Futterquelle optimiert. Dieser Mechanismus erlaubt einer echten Ameisenkolonie eine schnelle Anpassung an veränderte Umweltbedingungen, wie z. B. einen Fressfeind oder ein Hindernis auf einem der Pfade. Falls eine bestimmte Problemstellung sinnvoll in alternativen Pfaden kodiert werden kann und der Funktionswert der Zielfunktion als Geschwindigkeit der Ameise auf einer speziellen Route dargestellt werden kann, kann dieser Algorithmus benutzen werden.

Weiterhin soll hier der PSO genauer vorgestellt werden. Erstmals beschrieben wurde dieser Algorithmus im Jahr 1995 und findet keine exakte Entsprechung in der Natur. Natürliche Systeme besitzen keine übergeordnete Schwarmintelligenz, die regulierend in das System eingreift. Inspiriert wurde der Algorithmus vom Verhalten von Vogel- und Fischschwärmen. Der Pseudocode in Algorithmus 3 stellt einen PSO dar. Jeder Partikel P besitzt eine Position P_X in der Fitnesslandschaft, eine Geschwindigkeit P_V und eine Fitness P_F , die der Funktionswert der Zielfunktion an der Stelle P_X ist. Weiterhin besitzt jeder Partikel ein Gedächtnis und merkt sich die Position P_{BX} mit der besten Fitness P_{BF} , die dieser Partikel bisher besucht hat. Zu Beginn wird der Schwarm mit zufälligen Partikeln mit zufälligen Geschwindigkeiten erzeugt. Die Geschwindigkeit P_V ist dabei jedoch mehr eine Schrittweite, die während jeder Iteration durch einen Partikel zurückgelegt wird, als eine echte Geschwindigkeit, da der Algorithmus keine Zeitkomponente besitzt. Während jeder Iteration wird die Geschwindigkeit der Partikel so modifiziert, dass zum Geschwindigkeitsvektor Komponenten in Richtung guter Lösungen addiert werden. Konstante c_0 ist ein Trägheitsfaktor und bestimmt wie viel vom

Algorithmus 3 *particle swarm optimization*

```

for j=1 bis Anzahl der Partikel do
   $P(j)_X \leftarrow \text{ZufallsPosition}()$ 
   $P(j)_F \leftarrow f(P(j)_X)$ 
   $P(j)_V \leftarrow \text{ZufallsGeschwindigkeit}()$ 
   $P(j)_{BX} \leftarrow P(j)_X$ 
   $P(j)_{BF} \leftarrow P(j)_F$ 
end for
while Abbruchkriterium nicht erfüllt do
  for j=1 bis Anzahl der Partikel do
     $N \leftarrow \text{Nachbarschaft}(P(j))$ 
     $B \leftarrow \text{Bester}(N)$ 
     $P(j)_V \leftarrow c_0 P(j)_V + c_1 r_1 (P(j)_{BX} - P(j)_X) + c_2 r_2 (B_{BX} - P(j)_X)$ 
     $P(j)_X \leftarrow P(j)_X + P(j)_V$ 
     $f^* \leftarrow f(P(j)_X)$ 
    if  $f^*$  besser  $P(j)_{BF}$  then
       $P(j)_{BX} \leftarrow P(j)_X$ 
       $P(j)_{BF} \leftarrow P(j)_F$ 
    end if
  end for
end while

```

aktuellen Geschwindigkeitsvektor des Partikels beibehalten wird und liegt zwischen 0 und 1. Die Parameter c_1 und c_2 werden kognitiver und sozialer Parameter genannt und bestimmen wie stark der Geschwindigkeitsvektor in Richtung des eigenen Optimums und des Optimums in der Nachbarschaft geändert wird. Der Begriff der Nachbarschaft kann in diesem Zusammenhang Verschiedenes bedeuten. Die Nachbarschaft eines Partikels kann der gesamte Rest des Schwarms sein. In diesem Fall ist das Optimum der Nachbarschaft das globale Optimum des Schwarms, und viele PSO benutzen den gesamten Schwarm als Nachbarschaft. Die Nachbarschaft kann aber auch nur ein Teil des Schwarms sein. Dieser Teil kann entweder im wörtlichen Sinn von Nachbarschaft die topologisch nächstgelegenen Partikel umfassen, aber auch eine Zufallsauswahl ist möglich. Die Zufallszahlen r_1 und r_2 liegen zwischen 0 und 1. Der Vektor $(P_{BX} - P_X)$ geht von der aktuellen Position des Partikels zu der eigenen besten Lösung und der Vektor $(B_{BX} - P_X)$ geht von der aktuellen Position des Partikels zur besten Position in der Nachbarschaft. Nachdem der Schrittvektor entsprechend der bisher gefundenen Optima aktualisiert wur-

1 Einleitung

de, wird die neue Position des Partikels durch Addieren des Schrittvektors zur aktuellen Position ermittelt. Die neue Position des Partikels in der Fitnesslandschaft wird bewertet und P_{BX} und P_{BF} aktualisiert, falls eine bessere Position gefunden wurde. Danach beginnt die nächste Iteration mit dem Anpassen der Geschwindigkeit, falls das Abbruchkriterium noch nicht erfüllt ist. Untersuchungen an verschiedenen Testsystemen haben ergeben, dass einfache Schwärme wie oben beschrieben zur Explosion oder zur Implosion neigen [31]. Explosion bedeutet, dass die Partikel sich um das Optimum bewegen aber ihre Geschwindigkeit immer größer wird und sie sich somit effektiv immer weiter vom Optimum entfernen. Implosion ist die Konvergenz des gesamten Schwarms in einem Punkt und somit die vorzeitige Beendigung der Optimierung. Ob einer der Effekte auftritt bestimmen der Trägheitsfaktor c_0 und der kognitive und soziale Parameter c_1 und c_2 . Je größer diese sind, um so höher ist die Wahrscheinlichkeit, dass der Schwarm explodiert. Kleinere Konstanten bedeuten eine höhere Wahrscheinlichkeit der Implosion. Implosion kann zum Ende einer Optimierung durchaus ein erwünschtes Verhalten sein, da die lokale Suche des Algorithmus dabei verbessert wird. Das kann durch eine kontinuierliche Verringerung von c_0 erreicht werden. Explosion ist jedoch völlig unerwünscht. Um diesen Effekt zu verhindern, kann der Algorithmus um ein maximales Geschwindigkeitslimit V_{max} [32] oder einen Skalierungsfaktor χ [31] erweitert werden.

Der große Vorteil populationsbasierter Algorithmen ist ihre einfache Parallelisierbarkeit. Dies ist heutzutage besonders interessant, da die aktuelle Hardwareentwicklung immer mehr in Richtung paralleler Systeme, z. B. *multi core* CPU, geht. Bei dem PSO kann die Abfolge der einzelnen Schritte synchron oder asynchron sein. Bei synchroner Ausführung wird jede Operation für alle Partikel beendet, bevor die nächste begonnen wird, d.h. es wird der Geschwindigkeitsvektor aller Partikel aktualisiert und erst danach die Position aller Partikel geändert. Synchrone parallele PSO sind sehr leicht zu implementieren, in dem die Bewertung für mehrere Partikel gleichzeitig durchgeführt wird. Bei asynchroner Ausführung werden alle Operationen für einen Partikel durchgeführt, bevor mit dem nächsten Partikel begonnen wird. Parallele asynchrone Ausführung kann dazu führen, dass Partikel unterschiedlich viele Schritte zurückgelegt haben und sich in verschiedenen Zuständen

befinden. Dieser Ansatz ist etwas komplizierter umzusetzen, führt aber zu sehr guter paralleler Effizienz. Die Programmausführung muss nicht mehr an verschiedenen Stellen des Algorithmus synchronisiert werden. Somit entfallen auch die Wartezeiten an diesen Synchronisationsstellen. Die Qualität der Optimierung leidet unter asynchroner Ausführung nicht [33]. Es ist eine Vielzahl universeller Optimierungsalgorithmen verfügbar und viele sind noch nicht auf ihre Nutzbarkeit für das Moleküldocking untersucht worden. Da es schwer vorherzusagen ist, welcher Algorithmus für welches Problem gut funktioniert, erscheint es wünschenswert ein System zu haben, das die einfache Integration und Testung verschiedenster Algorithmen ermöglicht. Auch zukünftige Neuentwicklungen könnten so einfach auf ihre Eignung für das Moleküldocking untersucht werden.

1.3 Bewertungsfunktionen

Neben der Generierung von Posen und dem Sampling des Suchraums ist die Bewertung der möglichen Lösungen von größter Wichtigkeit für ein erfolgreiches Dockingexperiment. Diese Aufgabe wird durch die Bewertungsfunktion, auch Fitness- oder Scoringfunktion genannt, erfüllt. Diese werden auch in VS-Experimenten benutzt und dienen dort der Sortierung unterschiedlicher Komplexe. Im Gegensatz dazu muss die Bewertungsfunktion im Fall des Moleküldocking die unterschiedlichen Posen eines Komplexes anhand der Bindungsenergie richtig sortieren. Die Bindungsenergie ist nicht exakt berechenbar, kann aber durch ein empirisches Kraftfeld dargestellt werden. Wichtige Terme sind die Solvatationsenergie des Komplexes, des Proteins und des Liganden $\Delta G_{bind}^{complex}$, ΔG_{bind}^{prot} und ΔG_{bind}^{lig} . Auch die Entropieänderung ΔS für Protein und Ligand zwischen gebundenem und ungebundenem Zustand spielt eine Rolle. Und natürlich spielen auch die Wechselwirkungsenergie ΔG_{int} und die Energieänderung von Protein und Ligand im Verlauf der Bindung $\Delta \lambda$ eine Rolle. Alles zusammengefasst ergibt

$$\Delta G_{bind} = \Delta G_{bind}^{complex} - \Delta G_{bind}^{prot} - \Delta G_{bind}^{lig} + \Delta G_{int} - T\Delta S + \Delta \lambda.$$

Die praktische Durchführung dieser Berechnungen erweist sich jedoch als äußerst kompliziert. Ein Problem entsteht dadurch, dass sehr große Werte ad-

1 Einleitung

diert und subtrahiert werden, um am Ende einen sehr kleinen Wert zu erhalten. Somit führen verhältnismäßig kleine Fehler der Zwischenergebnisse zu sehr großen Fehlern im Endergebnis. Ein weiteres Problem ist, dass die Entropieänderung nur grob geschätzt werden kann. Und als drittes Problem kann genannt werden, dass im Prinzip der gesamte Konformationsraum von Protein, Ligand und Komplex berechnet werden müsste. Die dafür eingesetzten Methoden, z. B. *free energy perturbation*, sind extrem rechenaufwendig und können weder für Docking- noch für VS-Experimente benutzt werden. Für ausführliche Übersichtsartikel zu diesem Thema siehe [34, 35].

Im Kontext eines Dockingprogrammes muss die Bewertungsfunktion zwischen artifiziellen und natürlichen Komplexen unterscheiden können. Das Optimum einer Bewertungsfunktion sollte deshalb möglichst gut mit dem tatsächlichen Komplex übereinstimmen. Falsche oder schlechte Lösungen sollten auch schlechtere Bewertungen bekommen. Weiterhin sollte die Bewertungsfunktion schnell sein, da im Verlauf eines Dockingexperiments viele Komplexe zu bewerten sind. Bewertungsfunktionen können in die Gruppen Kraftfeld-Bewertungsfunktionen, empirische Bewertungsfunktionen und wissenbasierte Bewertungsfunktionen eingeteilt werden.

Eine Kraftfeld-Bewertungsfunktion (*Force Field Scoring* – FFS) benutzt die Terme der nichtbindenden Wechselwirkungen (WW) eines Kraftfelds. Ein Vorteil des FFS ist, dass Kraftfelder gut untersucht sind und auf physikalischen Eigenschaften beruhen. Nachteil ist, dass nur die enthalpischen Effekte in die Berechnungen einfließen. Dies führt zu einer Überbewertung elektrostatischer Effekte. Für die Vorhersage der Pose stellt das kein großes Problem dar. Bei der abschließenden Sortierung der Posen treten jedoch systematische Fehler auf, die zu falschen Vorhersagen in VS-Experimenten führen. Die allgemeine Form einer FFS-Funktion ist

$$E_{nonbond} = \sum_i^{lig} \sum_j^{prot} \left(\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^6} + \frac{q_i q_j}{D r_{ij}} \right)$$

und wird oft direkt aus einem Standardkraftfeld übernommen. A_{ij} und B_{ij} sind die *van der Waals*-Parameter (*van der Waals* – vdW), r_{ij} ist der Abstand zwischen Ligandatomben und Proteinatomben, q ist die Ladung eines Atoms und D ist die Dielektrizitätskonstante. Die Überbewertung elektrostatischer Wech-

selwirkung kann durch abstandsabhängige Dielektrizitätsfunktionen [36] oder die Lösung der Poisson-Boltzmann Gleichung [37] verringert werden. Unter der Voraussetzung eines rigiden Proteins kann die Berechnung mit Hilfe eines 3D-Gitters approximiert und die Ausführungsgeschwindigkeit um 1-2 Größenordnungen verbessert werden. Zusätzlich zu den nichtbindenden WW werden noch ligandinterne WW benötigt. Diese können ebenfalls aus einem Standardkraftfeld stammen. Das Programm DOCK benutzt Teile der AMBER-Energiefunktion [38] und einen expliziten Wasserstoffbrückenterm [39]. Als Änderung ist die Verwendung eines 8-4-Lennard-Jones-Potentials möglich. Dadurch wird die Oberfläche der Bewertungsfunktion weniger steil und leichter zu optimieren. GOLDScore, die Bewertungsfunktion des Programmes GOLD, benutzt diese weicheren vdW-Potentiale, gehört jedoch in die Gruppe der empirischen Bewertungsfunktionen, da es noch eine Reihe anderer Terme enthält.

Die Gruppe der empirischen Bewertungsfunktionen hat die meisten Vertreter. Diese Funktionen bestehen aus der Summe verschiedener physikalisch motivierter Terme, die durch multiple Regression so gewichtet werden, dass experimentelle Daten möglichst gut wiedergegeben werden. Hauptsächlich geschieht die Regression auf die Bindungsenergie eines Trainingsdatensatzes von Protein-Ligand-Komplexen. Die Fitness F kann durch $F \approx \sum_i \Delta G_i f_i(r_l, r_p)$ beschrieben werden. G_i ist der Regressionsfaktor der Teilfunktion $f_i(r_l, r_p)$. f_i ist eine Funktion der Position der Ligandenatome r_l und der Proteinatome r_p . Empirische Bewertungsfunktionen können Terme für Wasserstoffbrücken, ionische WW, hydrophobe WW, Entropie, π -Stacking, π -Kationen-WW und andere enthalten. Wasserstoffbrücken werden bewertet, indem alle Donor-Akzeptor Paare gezählt werden, die entsprechende Abstands- und Winkelkriterien erfüllen. Die Beiträge jeder einzelnen Wasserstoffbrücke werden oft gewichtet nach ihrer Abweichung von der idealen Geometrie gewertet. Die Bewertung ionischer WW erfolgt, ähnlich wie die der Wasserstoffbrücken, ohne Winkelkriterien. Ionische WW über große Distanzen werden normalerweise vernachlässigt. Hydrophobe WW werden meist durch die Größe der Protein-Ligand-Kontaktfläche abgeschätzt. Dabei wird unterschieden ob eine hydrophobe oder hydrophile Ligandenoberfläche mit einer hydrophoben oder hydrophilen Proteinoberfläche in Kontakt ist.

1 Einleitung

Hydrophob-hydrophobe Kontakte und hydrophil-hydrophile Kontakte sind günstig, während hydrophob-hydrophile Kontakte ungünstig sind. Der Einfluss der Entropie kann durch die Differenz der frei rotierbaren Bindungen des Liganden im gebundenen und ungebundenen Zustand abgeschätzt werden. π -Stacking und π -Kationen-WW können durch Zählen und Wichten bezüglich der Abweichung von idealen Geometrien ermittelt werden. Der bekannteste Vertreter dieser Gruppe ist SCORE1 [40], die LUDI [41] Bewertungsfunktion. In leicht veränderter Form kommt diese auch in FlexX zum Einsatz. GOLDScore und ChemScore, beide werden in GOLD benutzt, gehören ebenso in diese Gruppe wie X-SCORE [42] und PLP [43]. Vorteile der empirischen Bewertungsfunktionen sind die gute Ausführungsgeschwindigkeit und die leichte Interpretierbarkeit der Ergebnisse. Nachteil ist, dass diese Funktionen durch Regression an mittel- bis hochaffinen Komplexen gewonnen werden, und somit eine Tendenz für WW in dieser Art von Komplexen aufweisen. In den meisten Fällen erfolgt die Regression der Terme mit der Bindungsenergie. Da Dockingprogramme aber die korrekten Molekül-Komplexe vorhersagen sollen, sollten Bewertungsfunktionen für diesen Zweck auf die Vorhersage der richtigen Geometrie optimiert sein. Dies ist z. B. bei GOLDScore und PLP von PLANTS erfolgt.

Die dritte Gruppe bilden die wissensbasierten Bewertungsfunktionen. Diese Funktionen werden durch statistische Auswertung struktureller Daten aus der *Protein Data Bank* (PDB) [44] erzeugt. Empirische Bewertungsfunktionen können nur WW bewerten, die im Modell vorhanden sind. Da aber alle Modelle unvollständig sind, werden einige WW ignoriert. Im Gegensatz dazu versuchen wissensbasierte Bewertungsfunktionen nicht einzelne Effekte zu modellieren, um in der Summe eine möglichst gute Vorhersage zu treffen, sondern betrachten das Ergebnis aller WW gleichzeitig. Experimentell bestimmte Strukturen stellen den optimalen Zustand eines Komplexes dar und ermöglichen die Ableitung eines Potentials. Die Häufigkeit eines Kontaktes zwischen Ligand- und Proteinatomen in einem bestimmten Abstand kann als Maß für ihren energetischen Beitrag zur Bindung angesehen werden. Die erste universell einsetzbare wissensbasierte Bewertungsfunktion für Protein-Ligand-Komplexe ist BLEEP [45, 46]. BLEEP wurde von 351 Komplexen der PDB abgeleitet. Aktuellere Potentiale, z. B. PMF04 [47] mit 7152 oder DrugSco-

re [48] mit 6026 Komplexen, benutzen wesentlich größere Datensätze. PMF04 setzt sich aus abstandsabhängigen Paarpotentialen $E_{ij}(r)$, die nach

$$E_{ij}(r) = -kT \ln \left[f_j(r) \frac{\rho^{ij}(r)}{\rho^{ij}} \right]$$

gebildet werden, zusammen. Dabei ist ρ^{ij} die Dichte des Atompaars ij im Datensatz, $\rho^{ij}(r)$ die Dichte des Atompaars ij im Abstand r und $f_j(r)$ ist ein Volumenkorrekturfaktor. Der Volumenkorrekturfaktor soll dafür sorgen, dass nur effektiv belegtes Volumen in die Ableitung des Potentials einfließen kann. Unterscheiden können sich die verschiedenen wissenschaftlichen Bewertungsfunktionen im Datensatz, aus dem sie abgeleitet werden, der Einteilung in Atomtypen und der Art der Volumenkorrektur. Ein Vorteil der wissenschaftlichen Bewertungsfunktionen ist ihr allgemeiner Charakter. Alle möglichen Effekte sind bereits in diesen Potentialen enthalten. Ein Nachteil ist, dass einige Atompaare in den Datensätzen unterrepräsentiert sind und somit kein statistisch abgesichertes Potential abgeleitet werden kann. Prinzipbedingt ebenfalls unterrepräsentiert sind kurze Entfernungen. Für beide Fälle muss ein Ersatz, z. B. ein vdW-Potential, in das statistische Potential eingefügt werden, um trotzdem sinnvolle Ergebnisse zu ermöglichen.

Keine der vorgestellten Bewertungsfunktionen kann als allgemeingültige Bewertungsfunktion verwendet werden. Leider sind die Funktionen noch zu ungenau. Verschieden Bewertungsfunktionen funktionieren für verschiedene Probleme unterschiedlich gut. Einer der Gründe für die geringe Genauigkeit der Funktionen ist, dass diese an experimentelle Daten per Regression angepasst oder von experimentellen Daten abgeleitet werden. Darum können die Funktionen auch nur so genau sein wie die zugrundeliegenden experimentellen Resultate. Die Daten stammen aus verschiedenen Laboratorien mit unterschiedlichen Experimentatoren und werden in verschiedenen Testsystemen gemessen. Dies führt zu systematischen Fehlern im Datensatz. Informationen über den Protonierungszustand sind in Kristallstrukturen nicht enthalten, spielen jedoch bei der Ableitung der Parameter und der Anwendung der Bewertungsfunktion eine Rolle. Ebenfalls wichtig sind die Positionen von Wassermolekülen. In Kristallstrukturen sind jedoch nur konservierte Kristallwasser-Moleküle aufgelöst. Ein weiteres Problem ist die Art der Daten. Es

1 Einleitung

handelt sich meist um Daten hochaffiner Komplexe und somit sind mittel- und niedrigaffine Komplexe in den Datensätzen unterrepräsentiert. Schlechte oder ungünstige WW sind in diesen Datensätzen kaum enthalten. Die meisten Bewertungsfunktionen beruhen demzufolge auch ausschließlich auf Termen für günstige WW. Speziell für Bewertungsfunktionen in Dockingexperimenten ist dies von Nachteil, da zwischen günstigen und ungünstigen Komplexen unterschieden werden muss. Aber auch VS-Experimente leiden unter diesem Problem, da durch den einfachen additiven Charakter der Bewertungsfunktion große Liganden systematisch zu gut bewertet werden.

1.4 Zielsetzung

Obwohl heutzutage eine Reihe kommerzieller und frei verfügbarer Anwendungen für das Moleküldocking existieren, gibt es noch Raum für Verbesserungen. Kommerzielle Lösungen können aufgrund ihres geschlossenen Charakters nur in sehr geringem Maße angepasst werden. Einzige Möglichkeit ist die Modifikation von Parameterdateien. Bestehende quelloffene Lösungen sind monolithisch aufgebaut, benutzen einen antiquierten Programmierstil und machen deshalb einen Einstieg schwer. Mit fortschreitender Hardwareentwicklung ist es möglich komplexere und genauere Modelle in der Vorhersage von Komplexstrukturen zu verwenden und eine Anpassung der Vorhersagemodelle an das bearbeitete Problem kann die Genauigkeit erhöhen. Auch die Einführung von Nebenbedingungen, z. B. aus experimentellen Untersuchungen wie NMR-Spektroskopie oder FRET-Spektroskopie (*fluorescence resonance energy transfer*), kann die Genauigkeit der Vorhersagen erhöhen. Deshalb stellt eine leicht erweiterbare und anpassbare Lösung für das Moleküldocking eine wertvolle Bereicherung auf dem Gebiet der computergestützten Wirkstoffentwicklung dar.

In dieser Arbeit sollte ein flexibles und erweiterbares Dockingprogramm auf Basis populationsbasierter Algorithmen entwickelt werden. Es wird die Anwendung eines PSO zur Lösung des Dockingproblems vorgestellt. Zur Bewertung der Komplexe kommt eine empirische Bewertungsfunktion zum Einsatz. Die vorgestellte Lösung soll hinsichtlich ihrer Leistungsfähigkeit für die Vorhersage von Protein-Ligand-Komplexen und ihrer Eignung für das VS

überprüft werden.

1 Einleitung

2 Material und Methoden

2.1 Datensätze

2.1.1 *Astex Diverse Set*

Das *Astex Diverse Set* [49] ist eine Auswahl an Protein-Ligand-Komplexen, die speziell zur Untersuchung von Dockingprogrammen zusammengestellt wurde. Alle Komplexe stammen aus der PDB und wurden nach folgenden Kriterien ausgewählt:

1. Die Proteine müssen für das Wirkstoffdesign wichtig sein und die Liganden müssen *drug-like* [50] sein.
2. Sowohl Proteine als auch Liganden sollen möglichst divers sein, um ein breites Problemspektrum abzubilden.
3. Die experimentell bestimmten Strukturen sollen von hoher Qualität sein und keine Ungenauigkeiten aufweisen. Alle Bereiche der Bindetasche und der Ligand müssen in der Kristallstruktur aufgelöst sein.
4. Packungseffekte sollen bei den Komplexen keine Rolle spielen. Der Ligand hat nur mit einer Einheitszelle des Kristalls Kontakt.
5. Um Überschneidungen mit älteren Validierungsdatensätzen zu vermeiden und somit eine unabhängige Auswahl zu erhalten, werden nur relativ neue Strukturen (ab 2000) in die Auswahl genommen.

Die Auswahl wurde mit der gesamten PDB begonnen, die am 27. Juni 2006 31875 Kristallstrukturen enthielt. Eine Gruppierung der Sequenzen nach ihrer Homologie ergab 9188 Proteingruppen. Nach einer Analyse der in den Proteinen enthaltenen Liganden hinsichtlich ihrer *drug-likeness* verblieben 1310 Komplexe. Komplexe mit zu geringen Abständen zwischen Protein- und Ligandatomen (*vdW-clashes*) wurden nicht berücksichtigt. Falls der Ligand

2 Material und Methoden

große Spannungen aufwies, wurde dieser Komplex ebenfalls aus der Auswahl entfernt. Entfernt wurden auch Komplexe, in denen die experimentell bestimmten Elektronendichten in der Bindetasche nicht gut mit der Position der Atome übereinstimmten. Die resultierenden 836 Strukturen belegen 427 Gruppen und jede dieser Gruppen repräsentiert eine aktuelle, qualitativ hochwertige Struktur mit einem *drug-like* Liganden und hinterlegten Elektronendichten. Die verbleibenden Strukturen wurden einer manuellen Inspektion unterzogen, um die Korrektheit der Komplexe zu überprüfen, und es wurden vorzugsweise Komplexe ausgewählt, die von pharmazeutischem oder agrochemischem Interesse sind. Das Resultat ist eine Auswahl von 85 Protein-Ligand-Komplexen (s. Anhang A.1) mit folgenden Eigenschaften:

- Die Proteine weisen nur geringe Sequenzhomologie auf.
- 90 % der Proteine sind Wirkstoffziele oder Ziele für Agrochemikalien.
- Die Auflösung der Kristallstrukturen ist besser als 2,5 Å.
- 75 % der Liganden stammen aus Projekten der Wirkstoffforschung.
- Die experimentell bestimmten Elektronendichten stimmen mit der Lage der Ligandenatome überein.
- Es gibt keine *vdW-clashes*.
- Für 87 % der Komplexe sind Aktivitätsdaten vorhanden.
- Die Liganden haben weniger als 13 frei rotierbare Bindungen und weniger als 60 Schweratome.

2.1.2 *PDBbind*

Die *PDBbind* Datenbank [51, 52] ist ebenfalls eine Auswahl von Protein-Ligand-Komplexen aus der PDB. Die Auswahl umfasst nur Komplexe für die Bindungsaffinitäten vorhanden sind. Die Bindungsaffinitäten wurden von den Autoren aus der Primärliteratur oder deren Referenzen entnommen. Die *PDBbind* existiert seit 2003 und wurde kontinuierlich gepflegt. Die erste Ausgabe enthielt 1446 Komplexe, die aus insgesamt 19621 Strukturen ausgewählt wurden. In dieser Arbeit wurde die Version 2008 verwendet, die 4300 Komplexe aus 48092 Strukturen umfasst und für jeden Eintrag einen IC_{50} , einen

K_d - oder einen K_i -Wert aus der Literatur enthält. Die Autoren nennen diese 4300 Komplexe *general set*. Für die Ableitung und Validierung von Bewertungsfunktionen wurde, ausgehend von diesem Datensatz, ein *refined set* mit folgenden Kriterien ausgewählt:

- Die Auflösung der Kristallstrukturen ist besser als 2,5 Å.
- Der Ligand ist nicht kovalent gebunden.
- Nur binäre Komplexe sind enthalten.
- Für alle Komplexe sind K_i - oder K_d -Werten bekannt.
- Der Ligand muss ein organisches Molekül sein (nur C, N, O, P, S, F, Cl, Br, I und H).
- Das Protein muss aus proteinogenen Aminosäuren bestehen.

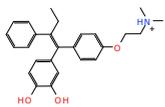
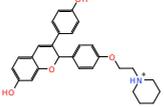
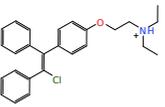
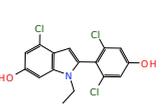
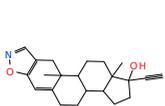
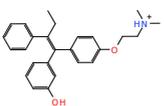
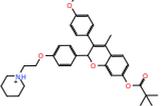
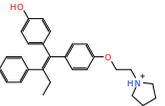
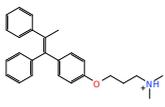
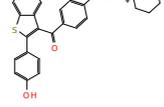
Ausgehend vom *refined set* wurde eine weitere Unterauswahl mit 210 Komplexen zusammengestellt, wobei besonderer Wert auf die Diversität der Proteine gelegt wurde. Die Sequenzen des *refined set* wurden anhand ihrer Sequenzhomologie in Gruppen eingeteilt. Falls eine Gruppe mindestens 4 Mitglieder hat, wurde der Komplex mit der höchsten Affinität, der niedrigsten Affinität und einer mittleren Aktivität ausgewählt. Die resultierenden 70 Gruppen enthalten somit 210 Komplexe. Die Autoren nennen diese Auswahl *core set*. Das *core set* stellt einen guten Testdatensatz zur Untersuchung von Dockingprogrammen dar, da keine Proteingruppe übermäßig häufig vorkommt und die Aktivitäten der Liganden über einen größeren Bereich verteilt sind. In dieser Arbeit wurde nur das *core set* dieser Datenbank verwendet. Die enthaltenen Komplexe sind in Anhang A.2 aufgeführt.

2.1.3 Estrogenrezeptor

Der Estrogenrezeptor (ER) ist ein Steroidrezeptor und gehört zur Gruppe der Kernrezeptoren. Kernrezeptoren [53] besitzen eine Ligandenbindungsdomäne (LBD) und eine DNA-Bindungsdomäne (DBD), eine variable Domäne, die LBD und DBD verbindet, und eine C- und N-terminale Domäne. Die hochkonservierte DBD besteht aus zwei Zinkfingermotiven und bindet spezifische DNA-Sequenzen. Diese DNA-Sequenzen nennt man *hormone response element*

2 Material und Methoden

Tab. 2.1: Estrogenrezeptor-Antagonisten im Testdatensatz.

 <p>3,4-Dihydroxytamoxifen</p>	 <p>CDRI86333</p>	 <p>Clomifen</p>	 <p>D15413</p>
 <p>Danazol</p>	 <p>Droloxifen</p>	 <p>EM762</p>	 <p>ICI129351</p>
 <p>ICI46414</p>	 <p>Raloxifen</p>		

(HRE). Die LBD ist nicht so hoch konserviert wie die DBD und besitzt eine spezifische Ligandenbindungsstelle für den entsprechenden Modulator. Nach Bindung des Liganden formen die Rezeptoren oft Dimere und binden dann an die HRE der DNA, wo sie als Transkriptionsfaktoren entweder die Transkription aktivieren oder hemmen.

Der ER gehört zur Kernrezeptor-Unterfamilie 3. Die Rezeptoren dieser Familie liegen im nichtaktivierten Zustand im Cytosol der Zelle, stabilisiert durch Hitzeschockproteine (HSP), vor. Durch Bindung eines Modulators dissoziiert der Rezeptor-HSP-Komplex. Die Rezeptor-Ligand-Komplexe bilden Dimere und werden in den Zellkern transportiert. In dieser Untersuchung wird die Struktur des Estrogenrezeptor- α aus einem Komplex von Estrogenrezeptor- α mit 4-Hydroxytamoxifen (PDB-ID 3ERT) verwendet. 4-Hydroxytamoxifen ist ein ER-Antagonist und inhibiert die Transkription von *estrogen-responsive genes*. Die verwendete Rezeptorstruktur liegt deshalb in der inaktivierten Form vor.

Der Testdatensatz besteht insgesamt aus 500 Molekülen, wovon zehn bekannte Antagonisten sind. Die aktiven Verbindungen (s. Tabelle 2.1) wurden aus

dem *world drug index* (WDI) [54] entnommen und sind dort mit Estrogen-Antagonist annotiert.

Die 490 *decoys* sind eine zufällige Auswahl von Molekülen aus der NCI-Datenbank [55] und wurden von der ZINC-Datenbank [56] heruntergeladen. Um eine Anreicherung aktiver Liganden in einem VS anhand einfacher physikochemischer Eigenschaften zu vermeiden, wurde die Auswahl der Liganden entsprechend der Eigenschaften der aktiven Verbindungen eingeschränkt. Die Molmasse beträgt 350-600 Da und die Moleküle haben zwischen 17 und 51 Schweratome. Die Anzahl der Wasserstoffbrückenakzeptoren, die polare Oberfläche und der $\log P(o/w)$ -Wert wurden entsprechend der Eigenschaften der aktiven Verbindungen gewählt. Die *decoys* befinden sich auf der DVD im Verzeichnis TESTSET/ER.

2.1.4 Acetylcholinesterase

Die Acetylcholinesterase (AChE) katalysiert den Abbau des Neurotransmitters Acetylcholin. Somit sorgt die AChE durch die schnelle Inaktivierung des ausgeschütteten Acetylcholins für eine Unterbrechung der Reizübertragung an cholinergen Synapsen. Das Enzym ist an neuromuskulären Synapsen im Muskelgewebe und an cholinergen Synapsen im ZNS lokalisiert.

Verschiedene AChE-Inhibitoren sind zur symptomatischen Behandlung der Alzheimer-Krankheit zugelassen. Die Alzheimer-Krankheit ist eine neurodegenerative Erkrankung, die durch extrazelluläre Ablagerungen von β -Amyloid-Protein, Neurofibrillen und die Degeneration cholinergischer Synapsen [59] gekennzeichnet ist. Durch den Einsatz von AChE-Inhibitoren [60] wird der Abbau von Acetylcholin verlangsamt und die Acetylcholinkonzentration im ZNS steigt an. Durch die höhere Acetylcholinkonzentration muss weniger Acetylcholin in den synaptischen Spalt freigesetzt werden um eine Reizweiterleitung zu ermöglichen. Somit wird trotz geringerer Acetylcholin-Ausschüttung eine normale Reizweiterleitung ermöglicht und die Abnahme der kognitiven Leistungsfähigkeit wird verlangsamt. Eine Heilung der Alzheimer-Krankheit ist jedoch nicht möglich, da mit dieser Therapie nur die Funktion der cholinergen Synapsen unterstützt werden kann. Eine Wiederherstellung der Funktion oder eine Verhinderung der fortschreitenden Degeneration ist nicht möglich.

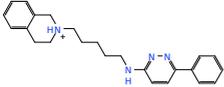
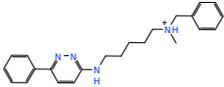
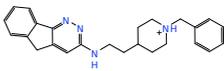
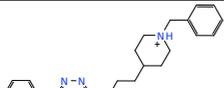
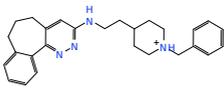
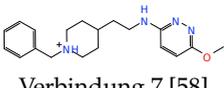
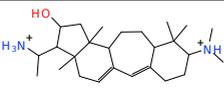
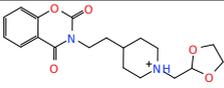
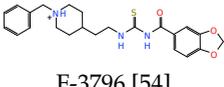
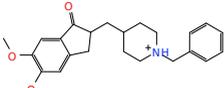
2 Material und Methoden

Als Zielstruktur des VS wurde eine AChE-Struktur in Komplex mit Donepezil (PDB-ID 1EVE) ausgewählt. Die zehn aktiven Verbindungen in Tabelle 2.2 sind aus zwei Veröffentlichungen [57, 58] und dem WDI entnommen. Die 490 *decoys* stammen aus der NCI-Datenbank und stellen eine zufällige Auswahl an Molekülen dar. Vor der Zufallsauswahl wurde die Datenbank auf folgende Eigenschaften eingeschränkt:

- Die Molmasse liegt zwischen 250 Da und 500 Da.
- Das Molekül enthält mindestens ein Stickstoffatom.
- Die polare Oberfläche, die Anzahl der Schweratome und der $\log P(o/w)$ -Wert entsprechen dem Wertebereich der aktiven Verbindungen.

Die *decoys* befinden sich auf der DVD im Verzeichnis TESTSET/AChE.

Tab. 2.2: AChE-Inhibitoren im Testdatensatz.

 <p>Verbindung 3r [57]</p>	 <p>Verbindung 3v [57]</p>	 <p>Verbindung 4e [58]</p>
 <p>Verbindung 4g [58]</p>	 <p>Verbindung 4j [58]</p>	 <p>Verbindung 7 [58]</p>
 <p>Buxaminol E [54]</p>	 <p>E-2030 [54]</p>	 <p>F-3796 [54]</p>
 <p>Donepezil [54]</p>		

2.1.5 Protein-Arginin Methyltransferase 1

Protein-Arginin Methyltransferase 1 (PRMT1) gehört zur Gruppe der Typ I Methyltransferasen. Typ I Methyltransferasen katalysieren die Methylierung der Argininseitenketten von Histonen zu asymmetrischem Dimethylarginin, während Typ II Methyltransferasen symmetrisches Dimethylarginin erzeugen. PRMT1 wird in vielen Geweben exprimiert und liegt sowohl im Cytoplasma als auch im Zellkern vor. PRMT1 besitzt zwei Bindetaschen, eine für das Substrat-Arginin und eine für das Kosubstrat S-Adenosylmethionin, welches als Methylgruppen-Donor fungiert. Histonmethylierung spielt, im Zusammenhang mit einer Vielzahl andere Histonmodifikationen, eine wichtige Rolle bei der Steuerung der DNA-Transkription. Es wurde z. B. nachgewiesen, dass die Methylierung von Histon H4 an R3 durch PRMT1 zu einer Reihe nachgeschalteter Prozesse führt, an deren Ende eine transkriptionelle Aktivierung steht [61].

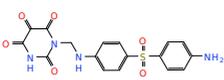
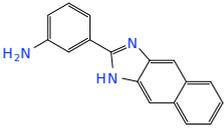
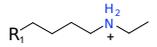
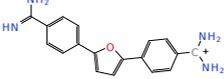
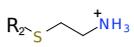
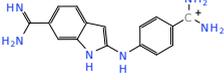
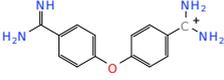
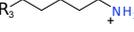
Als Zielstruktur kommt bei diesem Datensatz ein Homologiemodell zum Einsatz, dass aus einer PRMT3-Struktur (PDB-ID 1F3L) und mehreren PRMT1-Strukturen (PDB-ID 1ORI, 1ORH, 1OR8) der Ratte und der humanen PRMT1-Sequenz erstellt wurde [62]. Die aktiven Verbindungen sind in Tabelle 2.3 aufgeführt und stammen aus den angegebenen Veröffentlichungen oder aus unveröffentlichten Testdaten des Arbeitskreises Prof. Jung der Universität Freiburg. Das VS erfolgte an der Substrat-Bindetasche in Anwesenheit des Kosubstratanalogons S-Adenosylhomocystein. Die *decoys* wurden entsprechend der Anzahl der Stickstoffatome, der polaren Oberfläche und des $\log P(o/w)$ -Wertes der aktiven Verbindungen aus der NCI-Datenbank zufällig ausgewählt und befinden sich auf der DVD im Verzeichnis TESTSET/PRMT1.

2.2 Bibliotheken und Software

2.2.1 Xerces-C++

Xerces-C++ [64] ist ein XML Parser für C++. XML, eine Abkürzung für *Extensible Markup Language*, ist ein Textformat zum Darstellen und Austauschen hierarchisch strukturierter Daten. Ein XML Parser liest den Inhalt einer XML-Datei und wandelt diesen in eine Datenstruktur um, die von einem Compu-

Tab. 2.3: PRMT1-Inhibitoren im Testdatensatz.

 <p>Allantodapson [62]</p>	 <p>5756663 [63]</p>	 <p>7155176 [63]</p>
 <p>1</p>	 <p>NCI305831</p>	 <p>2</p>
 <p>NCI377363</p>	 <p>NCI9919</p>	 <p>3</p>

terprogramm benutzt werden kann. Mit Hilfe der Programmbibliotheken von Xerces-C++ kann ein Programm, um die Fähigkeit XML-Dateien zu lesen und zu schreiben, erweitert werden. Das Programm ist in portablen C++ geschrieben und kann somit auf vielen verschiedenen Systemen eingesetzt werden. Der Parser bietet hohe Geschwindigkeit, Modularität, Skalierbarkeit und eine genaue Umsetzung der XML-Spezifikationen. Die Programmierschnittstelle ist dokumentiert und das Programm wird aktiv gewartet. Xerces-C++ steht unter der *Apache Software License, Version 2.0* und erlaubt die Verwendung des Programmes in allen Arten von Projekten. In dieser Arbeit wurde die Version 2.8 des Programmes verwendet.

2.2.2 OpenSceneGraph

OpenSceneGraph [65] ist eine freie 3D-Grafik Bibliothek für C++, basierend auf der OpenGL Schnittstelle. Hauptsächlich wird diese Bibliothek zur Visualisierung in Spielen, zur Visualisierung in wissenschaftlichen Projekten, für *virtual reality*, für bildgebende Diagnoseverfahren in der Medizin und anderen Projekten, in denen große Datenmengen dreidimensional dargestellt werden müssen, eingesetzt. Die Funktionalität zur Visualisierung von Daten fand jedoch in dieser Arbeit keine Verwendung. Die OpenSceneGraph-Bibliothek

hat zusätzlich ein breites Sortiment an Funktionen zur Berechnung von Vektoren, Matrizen und Quaternionen. Diese Funktionalität fand in dieser Arbeit Verwendung. OpenSceneGraph ist unter der *OpenSceneGraph Public License* veröffentlicht und erlaubt die Verwendung in allen Arten von Projekten. In dieser Arbeit wurde die Version 2.X verwendet.

2.2.3 OpenMPI

OpenMPI [66] ist eine freie Implementierung des MPI-2-Standards [67]. MPI, die Abkürzung für *Message Passing Interface*, ist eine Programmierschnittstelle für C und Fortran, die eine Reihe von Funktionen für den Nachrichtenaustausch paralleler Berechnungen auf verteilten Computersystemen und in Multiprozessor-Umgebungen umfasst. MPI-Programme bestehen aus mehreren parallel laufenden Prozessen, die entweder auf einem oder mehreren Systemen laufen. MPI übernimmt den Datenaustausch zwischen diesen Prozessen und OpenMPI nutzt immer den schnellsten verfügbaren Kommunikationsweg. Falls die kommunizierenden Prozesse auf dem selben Computer ausgeführt werden, kann die Kommunikation über gemeinsam genutzten Hauptspeicher erfolgen. Laufen die Prozesse auf verschiedenen Computern wählt OpenMPI immer die schnellste verfügbare Netzwerkkommunikation, z. B. Infiniband statt TCP. Der Standard definiert Funktionen für Punkt-zu-Punkt, Punkt-zu-Gruppe und Gruppe-zu-Gruppe Kommunikation. Es gibt sowohl blockierende Kommunikation, d.h. die beteiligten Prozesse warten auf die Beendigung des Kommunikationsvorgangs, als auch nichtblockierende Kommunikation, d.h. die Prozesse fahren nach dem Starten der Kommunikation mit anderen Aufgaben fort. Weiterhin bietet der Standard Haltepunkte zur Synchronisation der Prozesse und kooperative IO-Routinen an. Der MPI Standard wurde 1994 vom MPI Forum (ein Komitee mit Mitgliedern aus Forschung und Industrie) als MPI-1 veröffentlicht. Die Erweiterung MPI-2 erschien 1996. OpenMPI ist *open source* und umfaßt MPI-1 und MPI-2. Die Lizenzierung unter der *New BSD license* ermöglicht eine Verwendung in allen Arten von Projekten. OpenMPI wurde von einem Konsortium aus kommerziellen Anbietern und verschiedenen Hochschulen entwickelt und wird aktiv gepflegt.

2.2.4 Boost C++ Libraries

Boost [68] ist eine Sammlung von C++ Bibliotheken für verschiedene Aufgaben, die nicht von der C++ Standard Library abgedeckt werden. Bei der Entwicklung der Routinen wird viel Wert auf die Portabilität des Codes gelegt, und deshalb sind die Bibliotheken für ein breites Spektrum an Architekturen und Betriebssystemen verfügbar. Oftmals werden Teile von Boost später in die C++ Standard Library übernommen. Vorteil der Verwendung von Boost ist eine höhere Produktivität beim Programmieren, da für viele Szenarien bereits Routinen vorhanden sind, und schnellere und fehlerfreiere Programme, da die große Nutzerbasis eine hohe Qualität des Codes sicherstellt. Die gesamte Boost Bibliothek steht unter der sehr liberalen *Boost Software License* und erlaubt uneingeschränkte Nutzung in kommerziellen und nichtkommerziellen Projekten. Boost existiert seit ca. 2000 und wird auch in großen Softwareprojekten, z. B. OpenOffice, eingesetzt. Die vielen kleinen verwendeten Routinen sollen nicht im Detail vorgestellt werden. Im folgenden soll jedoch auf die Boost.Serialization Bibliothek und Spirit, einen Textparser-Generator, explizit eingegangen werden.

Boost.Serialization

Unter Serialisierung wird, im Zusammenhang mit verteiltem Rechnen, die Umwandlung von Datenobjekten im Speicher in eine externe sequenzielle Darstellungsform verstanden. Moderne Programmierung beinhaltet dynamisch wachsende Container, intelligente Zeiger und polymorphe Klassen. Diese Datenobjekte sind nicht Byteweise kopierbar und können somit nicht mit den Standard-Mechanismen von MPI übertragen werden. Standardmäßig werden vom MPI-Standard nur die integralen Datentypen und Fließkomma-Datentypen unterstützt. Die Boost.Serialization Bibliothek bietet automatische Dekomposition komplexer Datentypen in serielle Datenströme und den Aufbau komplexer Datentypen aus seriellen Datenströmen. Die Anwendung ist sehr einfach, erfordert nur sehr wenig Quellcode und es existiert bereits eine transparente Schnittstelle zu MPI. Immer, wenn in diesem Projekt komplexe Datentypen zwischen Prozessen ausgetauscht werden müssen, kommt diese Bibliothek zum Einsatz.

Spirit

Spirit ist eine objekt-orientierte Umgebung zur Erzeugung statischer, rekursiv absteigender Textparser. Ein Textparser ist eine Programmkomponente, die Informationen aus Text interpretiert, in die benötigten Datentypen konvertiert und diese in den internen Strukturen des Programmes ablegt. Spirit beruht auf C++-Template Programmierung und unterstützt eine leicht abgewandelte Variante der erweiterten Backus-Normalform (EBNF) [69]. EBNF ist eine Meta-Sprache, mit der die Grammatik einer anderen Sprache beschrieben werden kann. Diese Grammatiken werden zum Beispiel bei der Herstellung von Compilern für Programmiersprachen benutzt und mit Hilfe von Parsergeneratoren in Textparser umgewandelt. In dieser Arbeit wird Spirit benutzt, um einen Parser für MOL2 Dateien [70] zu erzeugen. Diese Dateien enthalten die Strukturen der Moleküle, die in den Simulationen verwendet werden. MOL2 nutzt eine Grammatik, die leicht in EBNF Notierung formuliert werden kann. Vorteile dieser Methode sind der kompakte Quellcode des Parsers und ein fehlerarmes Programm im Vergleich zu selbstimplementierten Textparsern. Auch für andere Formate zum Speichern von Molekülinformationen sollte sich relativ leicht in eine EBNF-Grammatik erstellen lassen, wodurch das Programm an dieser Stelle leicht erweitert werden kann.

2.3 RMSD-Wert Bestimmung kleiner Moleküle

Der *root mean square deviation*-Wert (RMSD) gibt den durchschnittlichen Abstand zweier äquivalenter Atome in verschiedenen Konformationen eines Moleküls an und ist somit ein Maß für den Unterschied oder Abstand zweier Konformationen eines Moleküls. Da der RMSD als Bewertung für viele verschiedene Fragestellungen zum Einsatz kommt, gibt es keine allgemeingültige Formel zur Berechnung. Bei der Modellierung von Proteinen wird der RMSD oft aus der Superpositionierung zweier Proteine errechnet. In diesem Fall werden oft nur die C_{α} -Atome oder das Proteinrückgrat benutzt. Bei der Evaluierung von Dockingprogrammen werden vorhergesagte Protein-Ligand-Komplexe mit bekannten Kristallstrukturen verglichen. Dazu wird der RMSD-Wert zwischen vorhergesagtem Liganden und experimentell be-

stimmten Liganden berechnet. Die Formel zur Berechnung lautet

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

wobei δ den Abstand zwischen N Paaren äquivalenter Atome darstellt. Die Bestimmung der Paare äquivalenter Atome für die RMSD Berechnung kleiner Moleküle ist nicht trivial, falls die Moleküle symmetrische Molekülteile aufweisen. In diesem Fall muss das Subgraph-Isomorphismus Problem gelöst werden und alle äquivalenten Pfade in den Molekülen müssen miteinander verglichen werden. Die zur Verfügung stehenden Werkzeuge der RMSD Berechnung in MOE [71] beachten diesen Sachverhalt nicht. Weiterhin stand das Programm `smart_rms` aus der GOLD-Suite [72] zur Verfügung. Ob dieses Programm alle äquivalenten Pfade miteinander vergleicht, kann nicht abschließend geklärt werden, da Testberechnungen nicht eindeutig waren und die Dokumentation keine Informationen zu diesem Punkt enthält. Deshalb wurde ein eigenes Programm zur Berechnung des RMSD entwickelt. Nur so kann sichergestellt werden, dass die Auswertung der Dockingexperimente korrekt erfolgt.

Der Algorithmus beruht auf der Erzeugung aller topologisch äquivalenter Pfade in einem der beiden Moleküle. Diese können dann mit einem topologisch äquivalenten Pfad des anderen Moleküls verglichen werden, und der kleinste berechnete RMSD ist die korrekte Lösung. Für die Bestimmung dieser Pfade wird ein Deskriptor benötigt, der die Atome exakt topologisch klassifiziert. Diese Klassifizierung erfolgt durch Aneinanderreihung des Ergebnisses einer Breitensuche im Molekülgraph wie in Abb. 2.1 illustriert. In diesem Beispiel ergibt der erste Schritt der Breitensuche, ausgehend vom hellblau markierten Kohlenstoff mit roter Umrandung, 3 mal C (rosa unterlegt). Die gefundenen Atome werden entsprechend ihrer Ordnungszahl sortiert und gespeichert und der nächste Schritt der Breitensuche wird durchgeführt. Dieser ergibt 5 mal C und H (hellgrün unterlegt). Der fertige Deskriptor ist demnach `[[3C][5CH][5C6H][C7H][O2C][C3H][O][H]]`. In allen Molekülen mit der gleichen Topologie haben alle Atome, die äquivalent sind, den gleichen Deskriptor. Dieser ist unabhängig von der Art des Dateiformats in dem die Moleküle

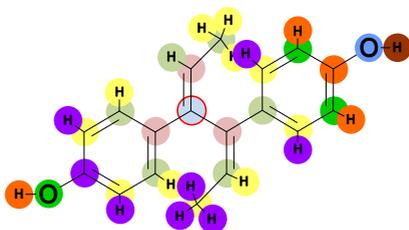


Abb. 2.1: Erstellung eines Topologie-Deskriptors mittels einer Breitensuche am Beispiel von Dienestrol. Atome mit gleichfarbiger Markierung besitzen die gleiche Tiefe in einer Breitensuche ausgehend von dem hellblau markierten Kohlenstoff mit roter Umrandung.

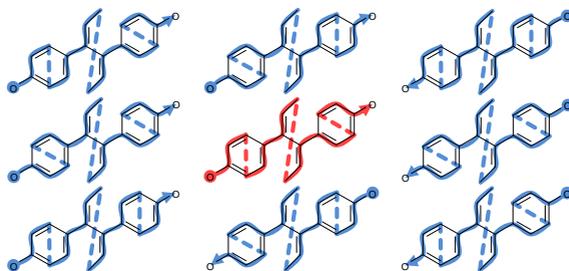


Abb. 2.2: Die RMSD-Werte müssen zwischen allen äquivalenten Pfaden des einen Moleküls (blau) und einem Pfad des anderen Moleküls (rot) berechnet werden.

gespeichert sind und der Sortierung der Atome in der Datei. Die Deskriptoren lassen sich lexikographisch vergleichen und somit sortieren. Mit Hilfe der topologischen Deskriptoren werden anschließend in einer rekursiven Tiefensuche alle äquivalenten Pfade aufgebaut. Die Tiefensuche wird so durchgeführt, dass immer zuerst der Pfad mit den größten Deskriptoren beschriftet wird. Somit ergeben sich für 2 Moleküle gleiche Pfade, falls die Topologie gleich ist. Sind die Pfade verschieden, kann kein RMSD berechnet werden, da die Moleküle unterschiedlich sind. Werden die Wasserstoffatome bei der RMSD Berechnung nicht mit verwendet, ergeben sich für Dienestrol acht mögliche Lösungen von äquivalenten Pfaden wie in Abb. 2.2.

Das Ignorieren der Wasserstoffe ist üblich und ergibt speziell beim Vergleich von Simulationen mit Kristallstrukturen Sinn, da diese keine Wasserstoffkoordinaten enthalten. Alle RMSD-Werte dieser Arbeit wurden ohne die Verwendung von Wasserstoffatomen berechnet. Der Quellcode zum Programm befindet sich auf der beigelegten DVD unter `paradocks/rmsd/rmsd.cpp`.

2.4 Quaternionen

Quaternionen sind eine Erweiterung komplexer Zahlen und können in Analogie zu komplexen Zahlen als Summe aus Realteil und Imaginärteil geschrieben werden.

$$\mathbb{H} = x_1 + x_2i + x_3j + x_4k$$

Entwickelt wurden sie von Sir William Rowan Hamilton und nach ihm wurden auch die Hamilton-Regeln für Quaternionen benannt.

$$i^2 = j^2 = k^2 = ijk = -1$$

In der angewandten Mathematik finden Quaternionen in der Berechnung dreidimensionaler Rotationen [73] Verwendung und auch in dieser Arbeit werden alle Rotationen, Orientierungen und Rotationsgeschwindigkeiten mit ihrer Hilfe beschrieben. Sie besitzen eine Reihe von Eigenschaften, die sie gegenüber anderen Repräsentationsarten dreidimensionaler Rotationen auszeichnen. Gegenüber Eulerschen Winkeln (EW) besitzen sie den Vorteil, nicht unter dem Problem des *gimbal lock* zu leiden. EW sind die Aneinanderreihung von drei Rotationen um die orthogonalen Axen im dreidimensionalen Raum. Wird beispielsweise um die X-Achse 90° gedreht, sodass die Y-Achse auf der Z-Achse liegt, ist die Rotation um die Y-Achse blockiert, was als *gimbal lock* bezeichnet wird. Eine andere Möglichkeit Rotationen darzustellen, ist eine Rotationsmatrix, jedoch ist die Berechnung von Matrizen im Vergleich zu Quaternionen aufwendiger.

Nur Einheitsquaternionen beschreiben eine Drehung im dreidimensionalen Raum. Für Einheitsquaternionen gilt

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 = 1.$$

Für jedes Einheitsquaternion q definieren q und $-q$ dieselbe Drehung. Im Unterschied zur Beschreibung von Drehungen durch Rotationsmatrizen gibt es also genau zwei Einheitsquaternionen, die eine Rotation beschreiben. Die Hintereinanderausführung von Drehungen entspricht der Multiplikation der Quaternionen. Für die Quaternionen $x = x_0 + x_1i + x_2j + x_3k$ und $y = y_0 + y_1i + y_2j + y_3k$ gilt

$$\begin{aligned} x \cdot y = & (x_0 \cdot y_0 - x_1 \cdot y_1 - x_2 \cdot y_2 - x_3 \cdot y_3) \\ & + (x_0 \cdot y_1 + x_1 \cdot y_0 + x_2 \cdot y_3 - x_3 \cdot y_2) \cdot i \\ & + (x_0 \cdot y_2 - x_1 \cdot y_3 + x_2 \cdot y_0 + x_3 \cdot y_1) \cdot j \\ & + (x_0 \cdot y_3 + x_1 \cdot y_2 - x_2 \cdot y_1 + x_3 \cdot y_0) \cdot k. \end{aligned}$$

Die Umkehr einer Rotation ist die Konjugation des Quaternion und ist definiert als

$$\bar{x} = x_0 - x_1 \cdot i - x_2 \cdot j - x_3 \cdot k.$$

Die einem Einheitsquaternion $x = x_0 + x_1i + x_2j + x_3k$ äquivalente Rotationsmatrix ist

$$\begin{pmatrix} 1 - 2(x_2^2 + x_3^2) & -2x_0x_3 + 2x_1x_2 & 2x_0x_2 + 2x_1x_3 \\ 2x_0x_3 + 2x_1x_2 & 1 - 2(x_1^2 + x_3^2) & -2x_0x_1 + 2x_2x_3 \\ -2x_0x_2 + 2x_1x_3 & 2x_0x_1 + 2x_2x_3 & 1 - 2(x_1^2 + x_2^2) \end{pmatrix}.$$

Eine zufällige, gleichmäßige Verteilung von Quaternionen, im Sinne einer sphärischen Gleichverteilung, erhält man nach Shoemake [74] aus drei Zufallszahlen X_1 , X_2 und X_3 mit Gleichverteilung zwischen 0 und 1 gemäß folgender Gleichung

$$\begin{aligned} \mathbb{H} = & \sin(2\pi X_2) \cdot \sqrt{1 - X_1} \\ & + \cos(2\pi X_2) \cdot \sqrt{1 - X_1} \cdot i \\ & + \sin(2\pi X_3) \cdot \sqrt{X_1} \cdot j \\ & + \cos(2\pi X_3) \cdot \sqrt{X_1} \cdot k. \end{aligned}$$

Korrekte Zufallszahlen sind für heuristische Verfahren von größter Wichtigkeit, denn zu kurze Zufallszahlenfolgen oder nicht normalverteilte Zufallszahlen führen zu künstlichen Trends. Falls eine lineare Normalverteilung

nicht direkt auf die Dimension der gesuchten Zufallszahl projiziert werden kann, muss eine entsprechende Transformation wie oben erfolgen.

2.5 X-SCORE

X-SCORE [42] ist eine empirische Bewertungsfunktion zur Vorhersage der Bindungsaffinität von Protein-Ligand-Komplexen. Wie alle empirischen Funktionen basiert auch X-SCORE auf der Regressionsanalyse der Funktionsterme an einem Trainingsdatensatz. Dieser Datensatz umfasst 200 Komplexe mit 70 verschiedenen Proteinen [42] aus der PDB-Datenbank. Für alle Komplexe wurden K_i oder K_d -Werte aus Publikationen zusammengestellt. Die allgemeine Form der Funktion ist

$$\Delta G_{bind} = \Delta G_{vdW} + \Delta G_{H-bond} + \Delta G_{deformation} + \Delta G_{hydrophobic} + G_0.$$

ΔG_{vdW} steht für die vdW-WW zwischen Protein und Ligand und wird durch ein 8-4-Lennard-Jones-Potential der Form

$$\Delta G_{vdW} = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{L}} \left[\left(\frac{d_{0ij}}{d_{ij}} \right)^8 - 2 \left(\frac{d_{0ij}}{d_{ij}} \right)^4 \right]$$

beschrieben. \mathcal{L} ist die Menge alle Ligandenatome und \mathcal{P} ist die Menge alle Proteinatome. d_{0ij} gibt den optimalen Abstand (die Summe der vdW-Radien) zwischen den Atomen vom Typ i und j an und d_{ij} ist der zu bewertende Abstand. Die Stärke der WW wird für alle Atomtypen gleich angenommen. Wasserstoffbrücken werden mit den geometrischen Parametern

- d , der Abstand zwischen Donor und Akzeptor,
- θ_1 , der Winkel zwischen Akzeptor, Donor und einem an den Donor gebundenen Schweratom und
- θ_2 , der Winkel zwischen Donor, Akzeptor und einem an den Akzeptor gebundenen Schweratom

durch die Funktion

$$HB_{ij} = f(d_{ij})f(\theta_{1ij})f(\theta_{2ij})$$

beschrieben. Alle Einzelfunktionen liefern Werte zwischen 0 und 1, wobei 0 einer Geometrie entspricht, die nicht in Wasserstoffbrücken gefunden wird,

und 1 stellt eine optimale Geometrie dar. Die Gesamtqualität einer Wasserstoffbrücke wird somit maßgebend durch den schlechtesten Teilwert bestimmt. Der Gesamtwert von ΔG_{H-bond} ergibt sich aus der Summe der Wasserstoffbrückenterme aller Paare von Protein- und Ligandenatomen

$$\Delta G_{H-bond} = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{L}} HB_{ij}.$$

Die Teilfunktionen wurden durch statistische Analyse des Trainingsdatensatzes abgeleitet und lauten

$$f(d_{ij}) = \begin{cases} 1 & d_{ij} \leq (d_{0ij} - 0,7\text{\AA}) \\ (1/0,7) \cdot (d_{0ij} - 0,7) & (d_{0ij} - 0,7\text{\AA}) < d_{ij} \leq d_0 \\ 0 & d_{ij} > d_0 \end{cases}$$

$$f(\theta_1) = \begin{cases} 1 & \theta_1 \geq 120^\circ \\ (1/60) \cdot (\theta_1 - 60) & 120^\circ > \theta_1 \geq 60^\circ \\ 0 & \theta_1 < 60^\circ \end{cases}$$

$$f(\theta_2) = \begin{cases} 1 & \theta_2 \geq 120^\circ \\ (1/60) \cdot (\theta_2 - 60) & 120^\circ > \theta_2 \geq 60^\circ \\ 0 & \theta_2 < 60^\circ. \end{cases}$$

$\Delta G_{deformation}$ dient der Abschätzung entropischer Effekte und wird aus der Anzahl der freien Rotoren im Molekül berechnet. Als Rotor werden alle azyklischen Einfachbindungen zwischen Schweratomen gewertet, jedoch nicht wenn eines der Atome keine weiteren Schweratome als Nachbar hat und somit eine terminale Gruppe ist. Das Ergebnis ist die Summe aller gewichteten Rotoren des Liganden

$$\Delta G_{deformation} = \sum_{i \in \mathcal{L}} RT_i$$

2 Material und Methoden

mit

$$RT_i = \begin{cases} 0 & \text{falls Atom } i \text{ keinen Rotor hat} \\ 0,5 & \text{falls Atom } i \text{ einen Rotor hat} \\ 1 & \text{falls Atom } i \text{ zwei Rotoren hat} \\ 0,5 & \text{falls Atom } i \text{ mehr als zwei Rotoren hat.} \end{cases}$$

$\Delta G_{hydrophobic}$ ist die Abschätzung des hydrophoben Effekts und wird durch drei verschiedene Verfahren berechnet. Der erste Algorithmus berechnet die vergrabene hydrophobe Oberfläche des Liganden. Ein Bereich der Ligandenoberfläche gilt dann als vergraben, wenn er innerhalb der lösungsmittelzugängigen Oberfläche des Proteins liegt. Die Berechnung erfolgt durch Summieren aller vergrabenen Oberflächen hydrophober Ligandenatome

$$HS = \sum_{i \in \mathcal{L}_H} SAS_i.$$

Der zweite Algorithmus summiert die Anzahl hydrophober Kontakte zwischen Protein und Ligand. Die Berechnung erfolgt nach

$$HC = \sum_{i \in \mathcal{P}_H} \sum_{j \in \mathcal{L}_H} f(d_{ij})$$

mit

$$f(d_{ij}) = \begin{cases} 1,0 & d_{ij} \leq (d_{0ij} + 0,5\text{\AA}) \\ (1/1,5) \cdot (d_{0ij} + 2 - d_{ij}) & (d_{0ij} + 0,5\text{\AA}) < d_{ij} \leq (d_{0ij} + 2\text{\AA}) \\ 0 & d_{ij} > (d_{0ij} + 2\text{\AA}). \end{cases}$$

Der dritte Algorithmus bezieht den Anteil eines Ligandenatoms an der Hydrophobizität des Liganden und die Hydrophobizität der Proteinumgebung mit ein. Die Funktion lautet

$$HM = \sum_{i \in \mathcal{L}_H} \log P_i \cdot HM_i$$

wobei $\log P_i$ der Anteil eines Atoms am $\log P$ des Liganden und HM_i ein Indikator für die Hydrophobizität der Proteinumgebung ist. HM_i ist 1 falls die Proteinumgebung hydrophob ist und 0 falls die Proteinumgebung hydrophil

ist.

Die Regressionsanalyse der Funktion ergibt einen Bestimmtheitsmaß von $R^2 = 0,591$ und eine Standardabweichung von $S = 1,47 \text{ p}K_d$ -Einheiten. Ein externer Datensatz von 30 Protein-Ligand-Komplexen zeigt eine Korrelation von $R^2_{pred} = 0,356$ und eine Standardabweichung von $S_{pred} = 1,58 \text{ p}K_d$ -Einheiten.

3 Ergebnisse und Diskussion

3.1 Design des Dockingprogrammes

Bei der Entwicklung des Dockingprogrammes wurde viel Wert auf ein modernes Design mit der Möglichkeit der Weiterentwicklung als *community project* gelegt. Obwohl mit AUTODOCK4 bereits ein quelloffenes Dockingprogramm unter der GPL existiert, zeigt sich an der langsamen Weiterentwicklung von AUTODOCK, dass der über lange Zeit unter einer sehr restriktiven nicht freien Lizenz gewachsene, monolithische Quellcode eine Weiterentwicklung durch Neueinsteiger sehr kompliziert macht. Erschwert werden könnte die Entwicklung als offene Software durch die kleine Nutzerbasis für Software aus dem Bereich *Molecular Modeling / Chemical Computing*, jedoch hat z. B. das Moleküldynamik-Paket GROMACS [75] bewiesen, dass eine von Beginn an offene Entwicklung ein erfolgreiches Entwicklungsmodell sein kann. Erstmals wurde GROMACS im Jahr 1995 veröffentlicht und hat im Laufe der Zeit eine Reihe von Weiterentwicklungen erfahren. Es ist eines der schnellsten Moleküldynamik-Pakete und die Homepage des Programmes führt über 1000 Publikationen auf, in denen mit GROMACS gearbeitet wurde. Vor allem bei der Entwicklung von Softwaremethoden im akademischen Bereich ist die Anzahl der Entwickler, die gleichzeitig ein Projekt bearbeiten können, limitiert und die Fluktuation ist verglichen mit kommerzieller Softwareentwicklung hoch. Folgende Ziele wurden deshalb als Grundlagen für das Design des Programmes festgelegt:

1. Falls ein aktiv gepflegtes externes Programm oder eine aktiv gepflegte externe Bibliothek eine benötigte Funktionalität bereitstellt, soll diese benutzt werden.
2. Falls möglich, sollen die verwendeten externen Bestandteile eine kommerzielle Lizenzierung nicht ausschließen.

3 Ergebnisse und Diskussion

3. Das Programm soll in viele eigenständige Komponenten (Klassen) unterteilt werden, die über definierte Schnittstellen miteinander kommunizieren.
4. Die Komponenten sollen leicht austauschbar sein.
5. Durch die Verwendung eines automatischen Dokumentationswerkzeuges soll eine umfangreiche Dokumentation der einzelnen Bestandteile erreicht werden.
6. Es sollen keine system- oder architekturenspezifischen Funktionen verwendet werden, um eine möglichst große Portabilität zu gewährleisten.
7. Um die Möglichkeiten populationsbasierter Metaheuristiken ausnutzen zu können, sollen alle Komponenten so erstellt werden, dass eine Parallelisierung möglich ist.

Durch den Einsatz externer Komponenten erhöht sich die Qualität einer Software im Bereich Fehlerfreiheit und Geschwindigkeit. Da die externen Komponenten auch in anderer Software benutzt werden, besitzen diese eine breitere Nutzer- und Entwicklerbasis als eine Eigenentwicklung, wodurch Fehler schneller gefunden und beseitigt werden. Kleine eigenständige Komponenten in Verbindung mit einer guten Dokumentation der Schnittstelle zu den anderen Komponenten ermöglichen einem Entwickler einen schnelleren Einstieg in die Zusammenhänge und Abläufe des Programmes. Somit steigen die Chancen, dass Nutzer mit den entsprechenden Kenntnissen gefundene Fehler selbst beheben oder möglicherweise das Programm erweitern. Eine große Portabilität führt zu mehr Nutzern und damit auch zu mehr gefundenen und beseitigten Fehlern. Als Name für das Programm wurde PARADOCKS gewählt. PARADOCKS ist ein Akronym für *Parallel Docking Suite* und soll die Funktion (Moleküldocking) und die Möglichkeit der parallelen Berechnung zum Ausdruck bringen

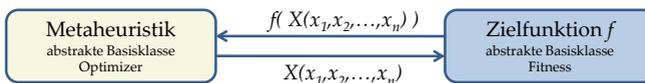


Abb. 3.1: Zusammenspiel von Metaheuristik und Zielfunktion.

3.1 Design des Dockingprogrammes

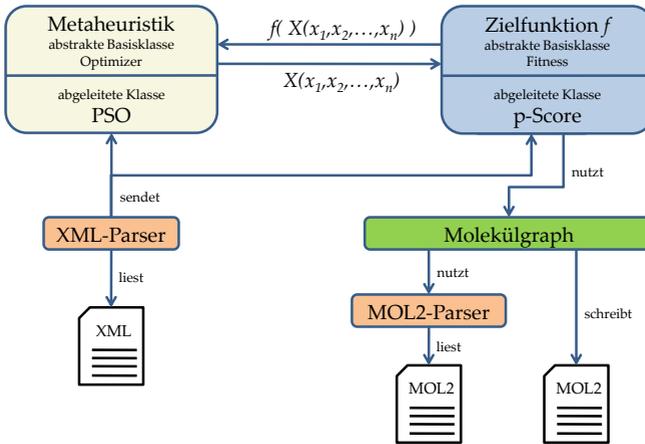


Abb. 3.2: Schematische Darstellung aller Komponenten des Dockingprogrammes mit PSO und p-Score als Beispiele für Metaheuristik und Fitnessfunktion.

Im Folgenden soll die Entwicklung des Designs beschrieben werden. Den Kern des Dockingprogrammes bildet die Optimierung einer Zielfunktion durch eine Metaheuristik wie in Abb. 3.1. Metaheuristiken optimieren beliebige n -dimensionale Funktionen und benötigen keine Informationen über die Art der Zielfunktion. In einem iterativen Prozess wird ein n -dimensionaler Vektor X (Position in der Fitnesslandschaft) von der Zielfunktion evaluiert und der Funktionswert $f(X)$ an die Metaheuristik zurückgegeben (Funktion `eval`). Um in diesem Kernbereich des Programmes die gewünschte Modularität zu erreichen sind beide Komponenten als abstrakte Basisklassen implementiert. Diese enthalten keine Funktionalität, abgesehen von kleinen Hilfsfunktionen, und sollen nur als Hüllen für konkrete Implementierungen dienen. Eine konkrete Implementierung erbt von der Basisklasse die Schnittstelle, so dass eine Metaheuristik mit beliebigen Fitnessklassen funktioniert und umgekehrt. Der Quellcode der zwei abstrakten Basisklassen befindet sich auf der beigelegten DVD in `paradocks/fitness/fitness.[hc]pp` und `paradocks/optimizer/optimizer.[hc]pp`.

Als konkrete Beispiele wurden in dieser Arbeit ein PSO als Metaheuristik und die empirische Bewertungsfunktion p-Score (siehe Abschnitt 3.1.4) als Fitnessfunktion entwickelt. Neben diesen beiden Bestandteilen, die den Kern des Programmes bilden, wird noch eine Reihe anderer Komponenten zur Verarbeitung von Eingabedateien und Molekülgraphen benötigt (siehe Abb. 3.2).

3.1.1 XML-Parser für die Konfigurationsdatei

Die Verarbeitung der Parameter aus der Konfigurationsdatei muss, unabhängig vom Typ der Kernkomponenten, immer durchgeführt werden und der Inhalt muss im gesamten Programm bereitgestellt werden. Deshalb wurden diese Routinen im zentralen Teil des Programmes angelegt. Da die Art und Anzahl der Optionen für die konkreten Implementierungen von Metaheuristiken oder Fitnessfunktionen verschieden sind, wurde ein flexibles Format gewählt. Das gewählte Eingabeformat XML ermöglicht die Darstellung hierarchisch strukturierter Daten in Textdateien. Listing 3.1 zeigt das Beispiel einer Konfigurationsdatei mit den 5 Abschnitten `optimizer`, `fitness`, `input`, `output` und `configuration`. Der Abschnitt `input` enthält die Strukturen für das Dockingexperiment sowie die Koordinaten und Größe der Bindetasche. Die Anzahl der Ligandeneinträge ist flexibel und ermöglicht somit das Verarbeiten mehrere Liganden nacheinander. Der Abschnitt `output` enthält den Pfad unter dem die Ergebnisse gespeichert werden sollen. Im Abschnitt `configuration` kann die Anzahl der Optimierdurchläufe pro Ligand angegeben werden und der Zufallszahlengenerator mit einer bestimmten Zahl gestartet werden. Dieser sollte jedoch nur zum Zweck der Fehleranalyse mit einem festen Wert initialisiert werden, da sonst immer wieder identische Zufallszahlenfolgen produziert werden und somit immer wieder die gleichen Ergebnisse berechnet werden. Die Abschnitte `optimizer` und `fitness` sind ähnlich strukturiert und bestimmen welche Metaheuristik und welcher Optimierer verwendet werden. Zusätzlich wird eine flexible Anzahl an Parametern an die Komponenten übergeben. Die Überprüfung von Parameteranzahl und Art der Parameter muss durch die jeweiligen Komponenten durchgeführt werden, da nur dort bekannt ist, welche Parameter benötigt werden. Die Vorteile von XML sind:

- XML ist eine durch das *World Wide Web Consortium* standardisierte

```

<paradocks> <!-- root node -->
<!-- type of the optimizer -->
<optimizer type="pso">
  <par val="100000"/> <!-- iterations -->
  <par val="30"/> <!-- particle count -->
  <par val="30"/> <!-- neighborhood size -->
  <par val="1"/> <!-- inertia weight start -->
  <par val="0.7"/> <!-- inertia weight end -->
  <par val="1.4"/> <!-- cognitive weight -->
  <par val="1.4"/> <!-- social weight -->
  <par val="5"/> <!-- maximum velocity -->
  <par val="0.79"/> <!-- maximum angle velocity for quaternions -->
  <par val="0.79"/> <!-- maximum angle velocity for angles -->
</optimizer>
<!-- type of the fitness function -->
<fitness type="pscore">
  <par val="1"/> <!-- example parameter -->
</fitness>
<input>
  <!-- protein input with coordinates and
  radius of the active site -->
  <protein file="protein.mol2"
    x="47.5387" y="27.5443" z="13.7082" rad="15"/>
  <!-- arbitrary number of ligand files with index for
  multi mol2 files -->
  <ligand file="ligand.mol2" idx="1"/>
</input>
<configuration>
  <!-- random seed, 0 seeds with system time -->
  <random seed="0"/>
  <!-- number of consecutive runs -->
  <runs val="10"/>
</configuration>
<output>
  <!-- every run will create a outputfile -->
  <!-- the name is prefix_protein_ligand_idx_run.xml -->
  <!-- none for production runs; best, all only for debug -->
  <iteration val="none"/>
  <!-- structure output at the end,
  possible values are: best, all-->
  <end val="best"/>
  <!--prefix for the output files-->
  <prefix val="./result"/>
</output>
</paradocks>

```

Listing 3.1: Beispiel für das Format der XML-Konfigurationsdatei

Grammatik.

- XML ist im Internet weit verbreitet und wird von vielen Programmen, z. B. auch OpenOffice, benutzt. Es existieren viele Werkzeuge zur Verarbeitung von XML-Dokumenten.
- XML ist normaler Text und kann somit gelesen werden.
- XML kann durch Computer einfach verarbeitet und direkt in Datenbanken benutzt werden.
- Die strenge Grammatik von XML verhindert Fehler.
- XML bietet die nötige Flexibilität für zukünftige Erweiterungen.

In dieser Arbeit fand der XML-Parser des Apache Projekts wie in Abschnitt 2.2.1 beschrieben Verwendung. Der Quellcode der Routinen zur Verarbeitung der Konfigurationsdatei ist auf der beigelegten DVD in den Dateien `paradocks/framework/ParadocksConfig.[hc].pp` zu finden.

3.1.2 Molekülgraph

Sollen in Computerprogrammen Moleküle verarbeitet werden, wird eine Datenstruktur benötigt, die deren topologische Eigenschaften widerspiegelt. Dabei handelt es sich um einen Graphen dessen Ecken die Atome und dessen Kanten die Bindungen zwischen Atomen darstellen. Einfache Listen- oder Feldstrukturen können diese Zusammenhänge nicht abbilden. Da wahrscheinlich jede Fitnessfunktion mit Molekülen umgehen muss, wurde eine zentrale Molekülgraph-Bibliothek entwickelt. Die Bibliothek gliedert sich in eine Baumstruktur und einen Graphen wie in Abb. 3.3 gezeigt. Die Baumstruktur wird gebildet aus Kette, Substruktur und Atom. Diese Elemente stehen in einer Eltern-Kind-Beziehung und dienen vorrangig der Organisation der Daten. Der eigentliche Molekülgraph wird durch Atom und Bindung gebildet. Die Bibliothek bietet Funktionalität zum Anlegen und Entfernen aller Elemente (Kette, Substruktur, Atom und Bindung), zum Abfragen von Nachbarschafts- und Eltern-Kind-Beziehungen und zum Abfragen von Eigenschaften der Elemente. Diese primitiven Funktionen finden Verwendung in verschiedenen Graph-Suchen. Zur Unterstützung von verteiltem Rechnen ist die Serialisierung der gesamten Molekülgraph-Struktur implementiert.

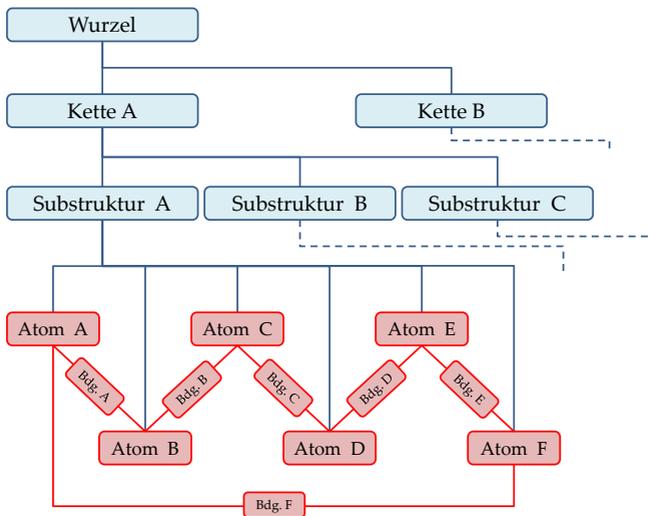


Abb. 3.3: Aufbau der Molekülgraph-Bibliothek. Die Baumstrukturelemente sind in blau und die Graphteile in rot markiert.

Durch Serialisierung kann der Graph in einen Datenstrom umgewandelt, an einen anderen Prozess gesendet und von diesem wieder in einen Graphen zurückverwandelt werden. Die Topologie der Graphen ist anschließend identisch.

Weiterhin bietet die Bibliothek einen Textparser für MOL2-Dateien zum Einlesen von Strukturen in den Molekülgraph. Die Implementierung erfolgte mit Hilfe des in Abschnitt 2.2.4 beschriebenen Parsergenerators. Dessen Vorteil gegenüber einem selbst erstellten Textparser ist die strikte Einhaltung des MOL2-Syntax. Der Quellcode ist in der Datei `paradocks/mgraph/MOL2.cpp` enthalten. Das MOL2-Format wurde in dieser Arbeit als Eingabeformat für Strukturdaten gewählt, da es detaillierte Atom- und Bindungstypisierungen enthält und von vielen Programmen erzeugt werden kann. Die Verwendung von PDB-Dateien würde die automatische Erkennung von Hybridisierungszuständen und Bindungsordnungen benötigen. Die Implementierung eines solchen Verfahrens ist nicht unmittelbares Ziel dieser Arbeit und bestehende Lösungen sind komplex. Um eine zuverlässige Typisierung zu ermöglichen,

benötigt das Programm MOL2-Dateien mit allen Wasserstoffen. Das Verzeichnis `paradocks/mgraph` der beigelegten DVD enthält alle Komponenten der Molekülgraph-Bibliothek.

3.1.3 Metaheuristik/PSO

Die abstrakte Basisklasse `Optimizer` nimmt im Programm den Platz der allgemeinen Metaheuristik ein. `Optimizer` beschreibt ein unspezifisches Optimierungsverfahren, besitzt aber selbst keine eigenen Funktionen. Es enthält nur die Informationen (Schnittstelle), die andere Programmteile benötigen, um ein beliebiges Optimierungsverfahren zu benutzen und die jede abgeleitete Klasse beinhalten muss. Diese Schnittstelle besteht aus der Initialisierungsfunktion und der Funktion zum Starten der Optimierung. Die Initialisierung erfolgt mit den Informationen des `optimizer`-Abschnitts aus der Konfigurationsdatei, der Fitnessfunktion als fertiges Objekt, der Position und dem Radius der Bindetasche aus der Konfigurationsdatei und einem Pseudozufallszahlengenerator. Die Erzeugung von Zufallszahlen ist wichtig für eine Metaheuristik, denn eine zu kurze Zufallszahlenfolge kann zu einer Verzerrung der Ergebnisse führen. Deshalb wurde in dieser Arbeit mit dem *Mersenne-Twister* MT 19937 Zufallszahlengenerator [76] gearbeitet. Die Periodenlänge von rund $4,3 \cdot 10^{6001}$ ist für *Monte-Carlo* Simulationen mit bis zu 623 Dimensionen entwickelt wurden und somit in jedem Fall ausreichend.

Als konkretes Beispiel für eine Metaheuristik wurde ein PSO implementiert, wie er in Dockingprogrammen bislang noch nicht verwendet wurde. Die drei existierenden Varianten *SODOCK* [17], *ClustMPSO* in *AUTODOCK* [18] und *PSO@AUTODOCK* [77] sind alles Erweiterungen von *AUTODOCK*. Bei *SODOCK* und *PSO@AUTODOCK* handelt es sich aber eher um eine lokale Suche mit Schwarm-Assistenz als um einen PSO, da nach jeder Iteration eine lokale Suche durchgeführt wird. Bei *ClustMPSO* handelt es sich um eine multikriterielle Variante eines PSO. Dabei werden inter- und intramolekulare WW gleichzeitig aber getrennt optimiert. *ClustMPSO* ist nicht öffentlich verfügbar. Die geringe Größe des Testdatensatzes von 5 Komplexen lässt den Schluss zu, dass es sich hierbei eher um eine Machbarkeitsstudie als um eine einsatzfähige Software handelt.

Die prinzipielle Funktionsweise eines PSO wurde in Abschnitt 1.2 bereits be-

schrieben. Es kommt ein PSO mit kontinuierlich abnehmenden Trägheitsfaktor und mit Geschwindigkeitsbegrenzung zum Einsatz [32]. Der Konfigurationsabschnitt in der Inputdatei benötigt 10 Parameter in folgender Reihenfolge:

1. I Beendigung nach I Iterationen
2. N Anzahl der benutzten Partikel
3. N_N Größe der Nachbarschaft
4. w_{start} Trägheitsfaktor zu Beginn
5. w_{end} Trägheitsfaktor am Ende
6. c_c kognitiver Parameter
7. c_s sozialer Parameter
8. PV_{max} maximale Geschwindigkeit der Positionskomponenten
9. OV_{max} maximale Winkelgeschwindigkeit der Orientierungskomponenten
10. AV_{max} maximale Winkelgeschwindigkeit der Torsionswinkelkomponenten.

Ein Beispiel ist in Listing 3.1 enthalten. Normalerweise benötigt der Optimierer keine Informationen über die Art des Problem, wenn alle Dimensionen einfache lineare Größen sind. Da dies beim Docking nicht der Fall ist wurde der PSO und die Schnittstelle zur Fitnessfunktion an die Erfordernisse angepasst. Die zu optimierenden Freiheitsgrade sind die Position des Liganden, die Orientierung des Liganden und die Rotation um alle frei rotierbaren Bindungen des Liganden. Aufgrund der verschiedenen Eigenschaften der Freiheitsgrade werden diese während der Optimierung unterschiedlich behandelt.

Die Optimierung der Position des Liganden ist die direkte Umsetzung des Algorithmus 3 aus Abschnitt 1.2. Das gesamte Molekül wird an einen Ankerpunkt gekoppelt und kann so in alle drei Dimensionen bewegt werden. Zu Beginn der Optimierung wird eine Population von Partikeln zufällig erzeugt. Jeder Partikel benötigt eine Position und eine Geschwindigkeit, und wird wie in Algorithmus 4 gezeigt initialisiert. P_N sind die Koordinaten des Partikels und V_N sind die Koordinaten des Ortsvektors der Geschwindigkeit. $Zufall(-1;1)$ ist der Zufallszahlengenerator und liefert linear gleichverteilte

Algorithmus 4 *Initialisierung der Position und Geschwindigkeit eines Partikel***repeat**

$$P_X \leftarrow \text{Zufall}(-1;1)$$

$$P_Y \leftarrow \text{Zufall}(-1;1)$$

$$P_Z \leftarrow \text{Zufall}(-1;1)$$

until $\sqrt{P_X^2 + P_Y^2 + P_Z^2} \leq 1$

$$P_X \leftarrow P_X \cdot AS_{radius} + AS_X ; B_X \leftarrow P_X$$

$$P_Y \leftarrow P_Y \cdot AS_{radius} + AS_Y ; B_Y \leftarrow P_Y$$

$$P_Z \leftarrow P_Z \cdot AS_{radius} + AS_Z ; B_Z \leftarrow P_Z$$

repeat

$$V_X \leftarrow \text{Zufall}(-1;1)$$

$$V_Y \leftarrow \text{Zufall}(-1;1)$$

$$V_Z \leftarrow \text{Zufall}(-1;1)$$

until $\sqrt{V_X^2 + V_Y^2 + V_Z^2} \leq 1$

$$V_X \leftarrow V_X \cdot PV_{max}$$

$$V_Y \leftarrow V_Y \cdot PV_{max}$$

$$V_Z \leftarrow V_Z \cdot PV_{max}$$

Zufallszahlen zwischen -1 und 1 . AS_N sind die Koordinaten der Bindetasche und AS_{radius} ist ihr Radius. Die erzeugten Partikel sind innerhalb einer Kugel um das Zentrum der Bindetasche gleichverteilt, und die Geschwindigkeitsvektoren der Partikel sind in einer Kugel mit dem Radius PV_{max} gleichverteilt. B_N sind die Koordinaten der besten Position, die der Partikel bisher besucht hat, und zu Beginn gleich den Koordinaten des Partikels. Die Optimierung erfolgt durch iterative Evaluation und Bewegung entsprechend dem Algorithmus. In jeder Iteration wird jeder Partikel um seinen Geschwindigkeitsvektor (GV) durch die Fitnesslandschaft bewegt und der GV wird an die Fitnesswerte und Positionen der anderen Partikel des Schwarms angepasst. Abb. 3.4 zeigt beispielhaft die Konstruktion des GV zweidimensional. Die aktuelle Geschwindigkeit V_n (hellgrün) wird mit dem Trägheitsfaktor w multipliziert und ergibt den Anteil der alten Geschwindigkeit an der neuen Geschwindigkeit (dunkelgrün). In den neuen GV V_{n+1} (gelb) fließen ebenfalls noch Anteile von Vektoren in Richtung der besten Lösung des Partikels (rot unterlegt) und in Richtung der besten Lösung aller benachbarten Partikel (blau unterlegt) ein. Diese beiden Vektoren werden jeweils mit einer Zufallszahl und einem Parameter multipliziert. Die Addition des Anteils der alten

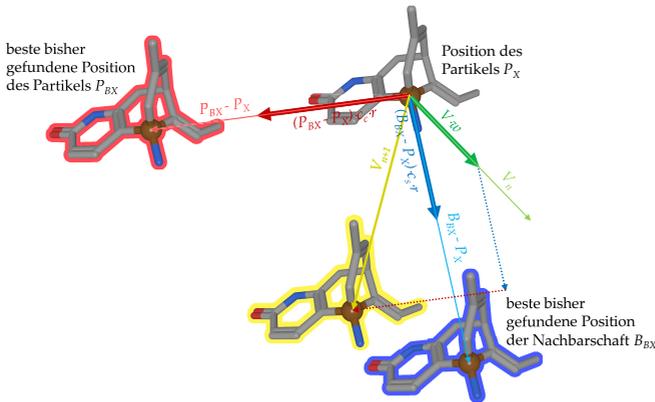


Abb. 3.4: Schematische Darstellung der Berechnung der Geschwindigkeitsvektoren. Das Ankeratom ist braun markiert.

Geschwindigkeit und der sozialen und kognitiven Vektoren ergibt den neuen GV des Partikels. Um diesen Vektor wird der Partikel in der nächsten Iteration bewegt (gelb unterlegt) und der Prozess beginnt erneut.

Die Optimierung der Orientierung versucht dieselbe Vorgehensweise zu verwenden, wie die Optimierung der Position. Da alle Orientierungen und Rotationsgeschwindigkeiten durch Quaternionen ausgedrückt werden, ist eine einfache Übertragung der Rechenregeln der Positionsoptimierung jedoch nicht möglich. Die Quaternionen werden vom Optimieralgorithmus als eine Einheit behandelt. Die Addition von Rotationen wird durch die Multiplikation der Quaternionen erreicht. Die Skalierung einer Rotation durch einen Faktor wird mittels einer *spherical linear interpolation* (SLERP) [78] erzeugt. SLERP berechnet ein Quaternion auf dem kürzesten Weg (Großkreisabschnitt) zwischen Ausgangsquaternion und Zielquaternion. Die Erzeugung der Zufallsquaternionen während der Initialisierung der Partikel erfolgt nach Shoemake [74]. Auch die zufälligen Rotationsgeschwindigkeiten werden mit diesem Algorithmus erzeugt und anschließend auf die maximale Rotationsgeschwindigkeit skaliert. Stellt man sich eine beliebige Orientierung als Rotation eines Punktes um das Zentrum einer Kugel vor, dann können alle Vorgänge während der Optimierung als Bewegung auf der Oberfläche einer

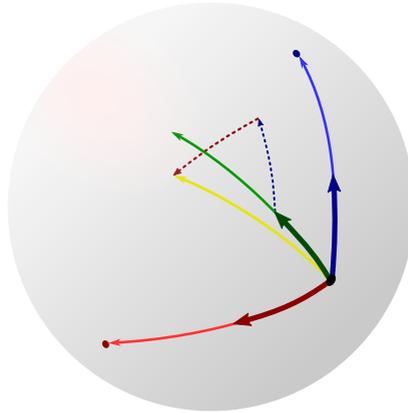


Abb. 3.5: Schematische Darstellung der Berechnung der Orientierung und Rotationsgeschwindigkeit. Der schwarze Punkt ist die aktuelle Orientierung des Partikels. Der rote und blaue Punkt entsprechen dem persönlich- und nachbarschaftsbesten Ergebnis. Der hellgrüne Pfeil ist die aktuelle Rotationsgeschwindigkeit und die resultierende Rotationsgeschwindigkeit ist in gelb gezeigt.

Kugel dargestellt werden. Eine beliebige Rotation entspricht einem Punkt auf der Oberfläche der Kugel und eine Rotationsgeschwindigkeit entspricht einem Vektor entlang eines Großkreises auf der Kugeloberfläche. In Analogie zu Abb. 3.4 wird die neue Rotationsgeschwindigkeit und Position des Partikels, wie in Abb. 3.5 gezeigt, konstruiert.

Im Verlauf der Optimierung der frei rotierbaren Bindungen muss der beschränkte Wertebereich von 2π für einen Vollkreis beachtet werden. Für die Position werden deshalb nur Werte zwischen 0 und 2π und für die Rotationsgeschwindigkeiten Werte zwischen $-\pi$ und π zugelassen. Falls der Wertebereich überschritten wird, wird der Wert um Vielfache von 2π korrigiert bis der Wertebereich wieder erreicht ist.

3.1.4 Fitnessfunktion/p-Score

Die Basisklasse `Fitness` definiert die Schnittstelle aller Metaheuristiken. Im Gegensatz zur Basisklasse `Optimizer` enthält `Fitness` einige wichtige Funk-

tionen. Alle Funktionen, die die Erzeugung der Konformationen des Liganden durchführen, wurden in der Basisklasse `Fitness` erstellt. Diese Funktionalität könnte auch durch jede Bewertungsfunktion separat bereitgestellt werden, da aber wahrscheinlich viele Bewertungsfunktionen diese Funktionen nutzen können, ist es sinnvoll, diese an einer zentralen Stelle zu definieren. Falls eine zukünftige Bewertungsfunktion eine andere Methode zur Erzeugung der Koordinaten verwenden soll, ist es jederzeit möglich, die Funktionen aus der Basisklasse `Fitness` nicht zu verwenden.

Wie in Abschnitt 3.1.3 beschrieben, erfolgt die Optimierung der Position eines Liganden, indem ein Ankeratom oder Ankerpunkt des Moleküls verschoben wird und alle anderen Atome relativ zu diesem Anker ihre Position beibehalten. Der Anker ist somit der Ursprung eines lokalen Koordinatensystems, welches im globalen System bewegt wird, und alle geometrischen Transformationen werden auf dieses Koordinatensystem angewendet. Die Bestimmung der Position des Ankers wird durch die Basisklasse `Fitness` im Verlauf der Vorprozessierung durchgeführt. Der Algorithmus sollte unabhängig von der Konformation des Input-Moleküls sein und keine negativen Auswirkungen

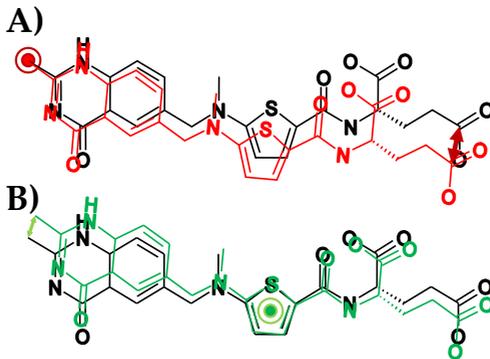


Abb. 3.6: Auswirkung der Wahl des Ankers auf eine Rotation. Die Anker sind als roter und grüner Kreis dargestellt. Bei **A**) ist der Anker ungünstig gewählt und führt zu ungleichmäßigen Änderungen der Koordinaten. Dieser Effekt tritt besonders bei länglichen Molekülen auf (Beispiel: Raltitrexed). Wird der Anker, wie in **B**), im Zentrum gewählt, dreht sich das Molekül gleichmäßig.

3 Ergebnisse und Diskussion

gen auf den Optimierungsprozess haben. Weder der Massenschwerpunkt noch der Mittelpunkt aller Atomkoordinaten eignen sich als Anker, da diese nicht unabhängig von der Konformation des Liganden sind. Es ist allerdings sinnvoll, den Anker in die Mitte des Liganden zu legen. Liegt der Anker an einem Ende des Moleküls oder außerhalb, führen Rotationen zu großen Änderungen der Koordinaten an einem Teil des Liganden, während sich der andere Teil des Liganden nicht oder nur wenig bewegt (siehe Abb. 3.6). Als Anker wird deshalb das Atom gewählt, das die niedrigste Anzahl an Ebenen in einer Breitensuche durch das gesamte Molekül aufweist. Gibt es mehrere Atome mit der gleichen Anzahl an Ebenen, wird das zuerst gefundene Atom benutzt. Die Anzahl der Ebenen ist für einige Beispielatome in Abb. 3.7 illustriert.

Die Analyse der rigiden Fragmente und frei rotierbaren Bindungen erfolgt ebenfalls während der Vorprozessierung des Liganden. Jede frei rotierbare Bindung ist ein Freiheitsgrad für die Optimierung. Als frei rotierbare Bindungen werden alle azyklischen Einfachbindungen angesehen. Da die Rotation um eine Bindung zu einer terminalen Gruppe keine neuen Konformationen erzeugt, werden diese Bindungen nicht als rotierbar markiert. Welche Atome oder Atomgruppen als terminale Gruppe eingestuft werden, hängt von der Behandlung der Wasserstoffatome ab. Falls diese explizit betrachtet werden, muss die Position der Wasserstoffatome ebenfalls optimiert werden. Somit sind nur Atome, die genau ein benachbartes Atom haben, eine terminale

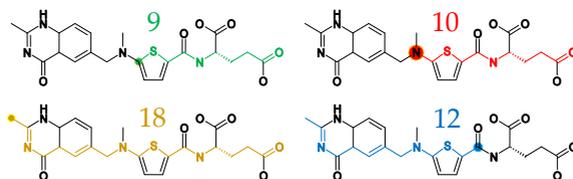


Abb. 3.7: Suche nach dem zentralen Atom eines Liganden. Das grün unterlegte Atom ist das zentrale Atom von Raltitrexed. Mit neun Schritten kann das am weitesten entfernte Atom in einer Breitensuche gefunden werden. Die anderen Farben zeigen Beispiele für andere Atome und deren Schrittzahl.

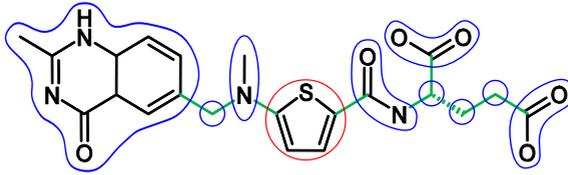


Abb. 3.8: Ergebnis der Vorprozessierung eines Liganden. Das zentrale rigide Fragment ist rot umrandet. Alle anderen rigiden Fragmente sind blau umrandet. Alle neun frei rotierbaren Bindungen des Raltitrexed sind grün markiert.

Gruppe. Falls die Wasserstoffe implizit betrachtet werden, sind alle Atome, die genau ein Schweratom als Nachbar haben, eine terminale Gruppe. Da p-Score die Wasserstoffatome implizit betrachtet, beziehen sich die folgenden Erklärungen auf implizite Wasserstoffe, obwohl der Algorithmus auch explizite Wasserstoffe oder nur polare Wasserstoffe verarbeiten kann. Alle nicht frei rotierbaren Bindungen werden nicht optimiert und bilden rigide Fragmente. Das rigide Fragment, das das Ankeratom enthält, ist das zentrale Fragment des Liganden. Der Mittelpunkt aller Atome des zentralen Fragments ist der Ursprung des lokalen Koordinatensystems des Liganden. Von den Ausgangskordinaten aller Atome muss der Ortsvektor des Ursprungs subtrahiert werden, um den Liganden im lokalen Koordinatensystem zu zentrieren. Ergebnis der Vorprozessierung ist die Anzahl an Freiheitsgraden, die vom Optimierer benötigt werden, und eine Datenstruktur, die eine effiziente Erzeugung der Ligandenkonformationen ermöglicht.

Im Verlauf der Optimierung wird durch den Optimierer ein Parametervektor als Eingabe für die Bewertungsfunktion übergeben. Aus diesem Parametervektor wird die Ligandenkonformation erzeugt und anschließend bewertet. Der Parametervektor besteht aus einem 3D-Vektor für die Translation, einem Quaternion für die Rotation und einem Winkel für jede frei rotierbare Bindung. Aus den Parametern wird für jedes Ligandenatom die entsprechende 3D-Transformation in Form einer 4x4-Matrix berechnet. Matrizen können Rotationen, Translationen, Skalierungen und Scheroperationen beschreiben, wobei hier nur Rotationen und Translationen von Interesse sind. Die Transfor-

mationen können durch Multiplizieren der Matrizen zusammengefasst werden, und durch Multiplizieren mit den Koordinaten eines Atoms angewendet werden. Somit kann eine einmal ausgerechnete, komplexe Transformation auf mehrere Atome effizient angewendet werden. Sowohl die Rotation als auch die Transformation müssen auf alle Atome angewendet werden. Die Rotationsmatrix M_R ergibt sich aus dem Quaternion gemäß der Formel aus Abschnitt 2.4 und die Translationsmatrix M_T ergibt sich aus dem 3D-Vektor $V = (x, y, z)$ und lautet

$$M_T = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Die globale Transformationsmatrix M_G ergibt sich aus M_R und M_T als

$$M_G = M_R \cdot M_T.$$

Geometrisch gesehen stellt M_G die Ausführung von Rotation und Translation nacheinander dar. Zusätzlich zu M_G besitzen alle Atome, die nicht zum zentralen Fragment gehören, noch weitere lokale Transformationen, die den Rotationen um die frei rotierbaren Bindungen entsprechen. Um die resultierende Transformation für jedes Atom zu erhalten, müssen alle Transformationen von Fragmenten, die näher am zentralen Fragment sind, mit einbezogen werden. Dies soll am Beispiel des Molekülfragments in Abb. 3.9 genau erläutert werden. Wenn man die Rotationsmatrizen für die Rotationen um die Bindungen B_1 , B_2 und B_3 als M_1 , M_2 und M_3 bezeichnet, so ist die Gesamttransformation für

$$F_1 \quad M = M_1 \cdot M_G,$$

$$F_2 \quad M = M_2 \cdot M_1 \cdot M_G \text{ und}$$

$$F_3 \quad M = M_3 \cdot M_2 \cdot M_1 \cdot M_G.$$

Der Quellcode für diese Funktionen befindet sich auf der beigelegten DVD in den Dateien `paradocks/fitness/fitness.[hc]pp`.

Die Funktion zur Bewertung der Komplexe wurde in Anlehnung an die Scoringfunktion X-SCORE [42] entwickelt. Da X-SCORE zur Vorhersage von Bindungsaffinitäten entwickelt wurde, gibt es eine Reihe von Unterschieden zur

hier verwendeten Funktion p-Score. Der Name p-Score wurde als Abkürzung für PARADOCKS-Score gewählt. Unverändert übernommen wurde das 8-4-Lennard-Jones-Potential zur Modellierung der vdW-Kontakte in der Form

$$E_{vdW} = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{L}} \left[\left(\frac{d_{0ij}}{d_{ij}} \right)^8 - 2 \left(\frac{d_{0ij}}{d_{ij}} \right)^4 \right],$$

wobei d_{0ij} die Summe der vdW-Radien der Atome i und j ist. Normalerweise werden in Kraftfeldern 12-6-Lennard-Jones-Potentiale verwendet. Durch den größeren Anstieg eines 12-6-Potentiales wäre die Fitnesslandschaft rauer und damit komplizierter zu optimieren. Zugunsten einer leichter optimierbaren Fitnesslandschaft wurden deshalb 8-4-Potentiale verwendet. Die Wasserstoffatome werden von der Funktion implizit behandelt. Es gibt keine Unterschiede in der Stärke der vdW-WW. Die Atomtypen wurden so genau wie möglich von X-SCORE übernommen und sind in Tabelle 3.1 aufgeführt. X-SCORE enthält keine Terme für die Bewertung der Ligandenkonformation. Da diese Funktionalität für ein Dockingprogramm benötigt wird, um ungünstige Ligandenkonformationen schlecht zu bewerten, wurde das Lennard-Jones-Potential auch zwischen allen Ligandenatomen mit mindestens vier Bindungen Abstand berechnet (E_{clash}). In anderen Kraftfeldern werden bereits Potentiale zum dritten Nachbar berechnet, die aber zum Teil anders parametrisiert sind. Da keine gesonderte Parametrisierung für 1,4-vdW-WW erfolgen sollte, wurden diese Potentiale nicht mit berechnet. Die Funktion zur Modellierung der Wasserstoffbrücken wurde mit Veränderungen übernommen. Die

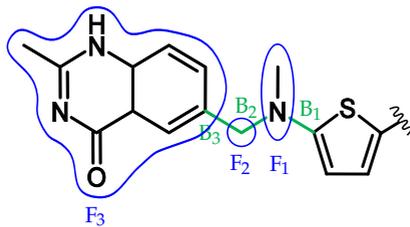


Abb. 3.9: Darstellung eines Teils von Raltitrexed mit nummerierten rigiden Fragmenten und rotierbaren Bindungen.

3 Ergebnisse und Diskussion

originale Funktion ist ausschließlich von den Winkeln θ_1 und θ_2 und dem Abstand d abhängig (s. Abschnitt 2.5). Dabei wird kein Unterschied zwischen frei rotierbarem Donor und Akzeptor und nicht frei rotierbarem Donor und Akzeptor gemacht. Diese Unterscheidung wurde eingeführt. Falls Donor oder Akzeptor mehr als ein Schweratom als Nachbar haben, ist eine freie Rotierbarkeit des Wasserstoffs am Donor oder des freien Elektronenpaares am Akzeptor nicht mehr gegeben. In diesem Fall werden die Ausrichtung von Elektronenpaar und Wasserstoff zueinander ausgewertet (s. Abb. 3.10). Kationen werden von diesem Algorithmus wie Wasserstoffe in einer Wasserstoffbrücke behandelt. Somit modelliert diese Funktion auch elektrostatische WW, jedoch nicht vollständig.

Die im X-SCORE enthaltenen Terme zur Bewertung des hydrophoben Effektes und der Term zur Abschätzung des Entropieverlustes wurden nicht mit

Tab. 3.1: vdW-Typen und Radien in X-SCORE.

Typ	Radius in Å
C tet	2,1
C tri ar	2,0
C tri re	1,9
C tri	1,9
C lin	1,8
N tet	1,8
N tri	1,75
N lin	1,75
O tet	1,65
O tri	1,55
H ₂ O	1,75
S tet	2,1
S tri	2,0
P	2,0
F	1,5
Cl	1,75
Br	1,9
I	2,05
Si	2,0
Metallionen	1,25

tet - tetraedrisches Atom, tri - trigonal-planares Atom, lin - lineares Atom

ar - Atom mit aromatischer Bindung, re - Atom mit delokalisierter Doppelbindung z. B.: Amid, Carboxylat

implementiert, da diese Terme wichtiger für die Vorhersage der Bindungsfähigkeit sind und weniger Einfluss auf die Erzeugung der Geometrie haben. Der Quellcode für die Bewertungsfunktion befindet sich auf der DVD in den Dateien `paradocks/fitness/pscore/pscore.[hc].pp`. Jedoch sind in diesen Dateien bereits die optimierten Parameter enthalten (s. Abschnitt 3.2.2).

3.2 Parametrisierung

Die Auswahl der optimalen Parameter für eine Metaheuristik ist nicht rational lösbar. Der Optimierungsprozess ist durch die Verwendung von Zufallszahlen nicht deterministisch und somit können die Resultate zweier Optimierungen nicht direkt miteinander verglichen werden. Auch mit ungünstigen Parametern kann „zufällig“ eine gute Lösung gefunden werden. Eine gute Lösung ist dabei aus Sicht des Optimierers eine Lösung mit guter Fitness. Ob die erzeugten Komplexe sinnvoll sind oder nicht kann durch den Optimierer nicht entschieden werden, da sich dieser ausschließlich nach der Fitness richtet. Maß für eine gute Optimierung ist somit eine gute Fitness.

Auch die Parametrisierung der Bewertungsfunktion ist nicht rational lösbar. Mit der Kristallstruktur liegt pro Komplex nur ein Datenpunkt vor, welcher für eine korrekte Parametrisierung nicht ausreichend ist. Weiterhin soll die Funktion nicht für einen bekannten Komplex exakte Resultate liefern, sondern unbekannte Komplexe gut vorhersagen. Ziel der Parametrisierung der Bewertungsfunktion ist eine Funktion, für die das globale Optimum möglichst dicht am experimentell bestimmten Komplex liegt. Korrekte Vorhersa-

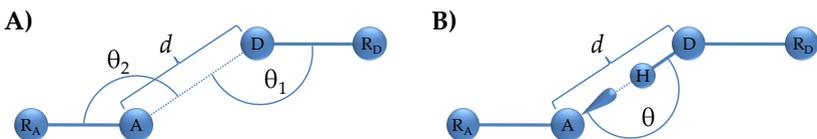


Abb. 3.10: Parameter zur Bewertung von Wasserstoffbrücken in A) X-SCORE und B) deren Erweiterung in p-Score. A - Akzeptor, D - Donor, H - Wasserstoffatom, R_A - Nachbar des Akzeptors, R_D - Nachbar des Donors

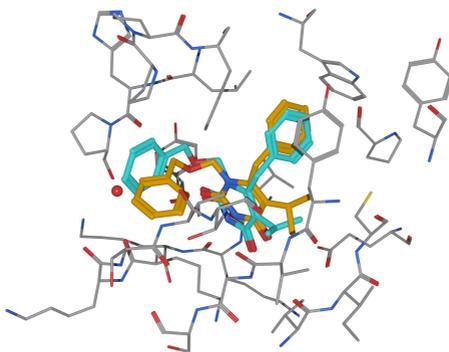


Abb. 3.11: Bindungstasche von 1JLA mit dem kokristallisierten Liganden (cyan) und der besten Vorhersage (orange). Der vorhergesagte Ligand hat einen RMSD von $1,3 \text{ \AA}$ und eine Fitness von $-585,97$. Das Dockingexperiment erfolgte mit 30 Wiederholungen und den Anfangsparametern.

gen können nur durch das Zusammenspiel von guter Bewertungsfunktion und zuverlässiger Optimierung erzeugt werden.

3.2.1 Wahl der Parameter des PSO

Um Parametersätze miteinander vergleichen zu können, muss die Anzahl an nötigen Wiederholungen ermittelt werden, die zuverlässige Vergleiche erlaubt. Verglichen werden können nur die Mittelwerte mehrerer Experimente, da die Einzelexperimente durch Zufallszahlen bestimmt werden. Es ist davon auszugehen, dass ein guter Parametersatz im Durchschnitt eine Lösung mit besserer Fitness liefert als ein schlechter Parametersatz. Für diesen Zweck wurde ein Testdatensatz aus dem Trainingsdatensatz *Astex Diverse Set* ausgewählt und mit vielen Wiederholungen gedockt. Ausgewählt wurde der Komplex mit der PDB-ID 1JLA, der mit 7 frei rotierbaren Bindungen nicht zu den einfachen Problemen gehört. Trotzdem erhält man auch mit den Anfangsparametern (s. Listing 3.1) gute Vorhersagen (s. Abb. 3.11).

Es wurden 8000 Wiederholungen durchgeführt und daraus acht zufällige Datenreihen mit je 1000 Wiederholungen gebildet. Diese Datenreihen wurden

statistisch ausgewertet. Mit zunehmender Anzahl an Wiederholungen kann eine Stabilisierung des Mittelwertes festgestellt werden (s. Abb. 3.12), die Mittelwerte werden vergleichbar. Es wurde festgestellt, dass bei 100 Wiederholungen der Mittelwert um etwas mehr als 11 variiert. Nach 200 Wiederholungen findet man eine Streuung von ca. 8, nach 600 Wiederholungen beträgt die Streuung 5 und nach 1000 Wiederholungen nur noch ca. 3. Ausgehend von diesem Vergleich wurde festgelegt, dass mindestens 200 Wiederholungen durchgeführt werden müssen, um Parametersätze miteinander zu vergleichen. Obwohl diese Festlegung willkürlich ist, kann anhand dieses Experimentes abgeschätzt werden, wie groß der mögliche Fehler ist und welche Unterschiede als signifikant einzustufen sind. Höchstwahrscheinlich werden sich die statistischen Parameter von Experimenten mit viel mehr Wiederholungen noch besser vergleichen lassen, jedoch muss die praktische Durchführbarkeit der Experimente gewahrt bleiben. Da die große Anzahl an Daten nicht tabellarisch aufgeführt werden kann, finden sich hier nur die Statistiken der Einzelexperimente. Die experimentellen Ergebnisse sind auf der DVD im Verzeichnis `DOCK_RUN_STATISTIK` zu finden.

Die ersten Parameter, die optimiert wurden, sind die Trägheitsfaktoren w_{start} und w_{end} . In Schritten von 0,1 wurden alle Möglichkeiten zwischen 0 und 1 mit $w_{start} \geq w_{end}$ getestet. Alle anderen Parameter wurden unverändert aus Listing 3.1 verwendet. Die Mittelwerte der Einzelexperimente sind in

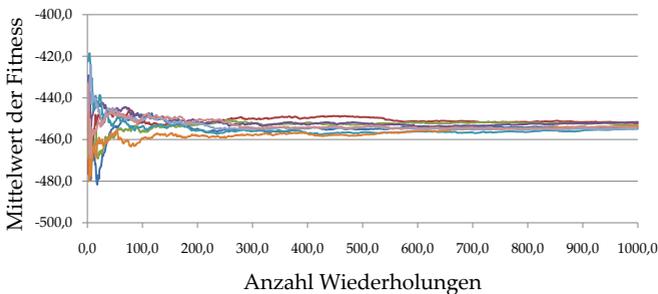


Abb. 3.12: Verlauf des Mittelwertes in Abhängigkeit von der Anzahl an Wiederholungen. Aufgetragen sind die Mittelwerte von acht Datenreihen mit bis zu 1000 Einzelexperimenten am Komplex 1JLA.

3 Ergebnisse und Diskussion

Abb. 3.13 dargestellt. Eine erste Analyse nach 200 Einzelexperimenten pro Parametersatz ergab, dass w_{start} zwischen 0,6 und 1 liegen muss und w_{end} zwischen 0 und 0,4. Die anderen Kombinationen lieferten viel schlechtere Ergebnisse und wurden in der Folge nicht weiter berechnet. Da mehr Wiederholungen zu weniger Ungenauigkeit der Ergebnisse führt, wurde die Statistik der guten Parametersätze auf 400 Wiederholungen erweitert. Die erzielten Verbesserungen der Optimierung ist signifikant. Die Ausgangsparameter von $w_{start} = 1$ und $w_{end} = 0,7$ lieferten eine durchschnittliche Fitness von -452. Die Ergebnisse für die getesteten Kombinationen reichen von -345 bis -579. Der Bestwert -579 wurde mit der Kombination $w_{start} = 1,0$ und $w_{end} = 0,2$ erreicht. Für die 12 besten Kombinationen wurde der Mittelwert mit 900 Wiederholungen abgesichert. Für alle folgenden Experimente wurden die Parameter $w_{start} = 1$ und $w_{end} = 0,2$ verwendet. Die experimentellen Ergebnisse sind auf der DVD im Verzeichnis `PS0_INERTIA_WEIGHT` zu finden.

Als nächstes wurde die Kombination aus kognitivem Parameter c_c und sozialem Parameter c_s untersucht. Da diese Parameter Einfluss auf den gleichen Teil des Optimieralgorithmus haben, wurden sie zusammen betrach-

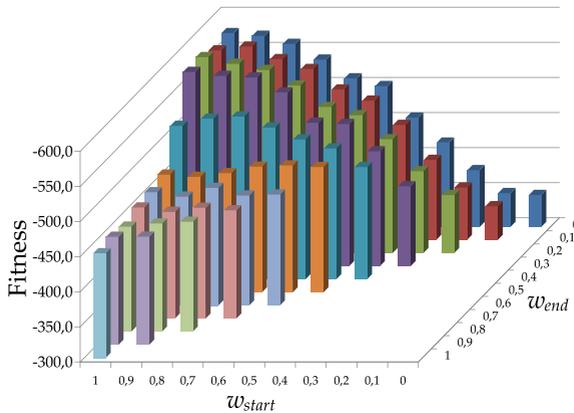


Abb. 3.13: Auswirkungen der Trägheitsfaktoren w_{start} und w_{end} auf die Effizienz der Optimierung. Aufgetragen ist der Mittelwert der Fitness von mindestens 200 Wiederholungen mit den angegebenen Trägheitsfaktoren.

tet. Es wurden alle Kombinationen der beiden Parameter zwischen 1 und 4 mit einer Schrittweite von 0,2 getestet. Auch bei diesen Parametern wurde nach 200 Wiederholungen der Wertebereich auf die besten Parameter eingeschränkt und die Statistik auf 1000 Wiederholungen erweitert. Die Ergebnisse sind in Abb. 3.14 dargestellt. Es wurde festgestellt, dass c_c kleiner als 2,2 sein sollte, da bei größeren Werten die Effizienz deutlich abnimmt. Der Einfluss von c_s ist weniger stark ausgeprägt. Das beste Ergebnis von -594 wurde mit $c_c = 1,0$ und $c_s = 3,4$ gefunden. Andere Parameterkombinationen lieferten ähnlich gute Werte und liegen noch innerhalb des Bereiches der Ungenauigkeit der Mittelwerte von ca. 5. Da es keine objektive Begründung für die Bevorzugung anderer äquivalenter Parameterkombinationen gibt, wurden alle folgenden Experimente mit $c_c = 1,0$ und $c_s = 3,4$ berechnet. Diese Parameter bedeuten, dass für eine gute Optimierung die globale Suche gegenüber der lokalen Optimierung stärker gewichtet sein muss. Die experimentellen Ergebnisse sind auf der DVD im Verzeichnis PS0_C0G_S0C_PAR zu finden.

Die verbleibenden Parameter PV_{max} , OV_{max} und AV_{max} wurden unabhängig voneinander getestet. Für jeden Parameter wurden schrittweise verschiedene Werte verwendet. PV_{max} , die maximale Geschwindigkeit des Zentrums des lokalen Koordinatensystems, wurde zwischen $1 \frac{\text{\AA}}{\text{Schritt}}$ und $20 \frac{\text{\AA}}{\text{Schritt}}$ in Abständen von $1 \frac{\text{\AA}}{\text{Schritt}}$ untersucht. OV_{max} , die maximale Rotationsgeschwindigkeit

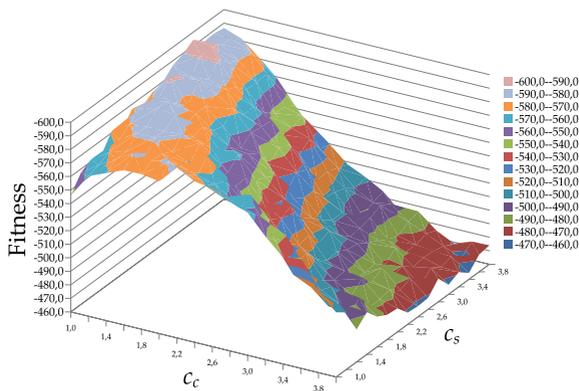


Abb. 3.14: Effizienz der Optimierung in Abhängigkeit von c_c und c_s .

3 Ergebnisse und Diskussion

keit des lokalen Koordinatensystems, und AV_{max} , die maximale Rotationsgeschwindigkeit der Rotation um die rotierbaren Bindungen, wurden in Schritten von $0,3 \frac{\text{rad}}{\text{Schritt}}$ zwischen $0,3 \frac{\text{rad}}{\text{Schritt}}$ und $3,0 \frac{\text{rad}}{\text{Schritt}}$ untersucht. Die Ergebnisse sind in Abb. 3.15 dargestellt. Es konnte kein positiver Effekt einer Begrenzung der Geschwindigkeit auf die Optimierung festgestellt werden. In allen folgenden Experimenten wurde daher ohne eine Begrenzung der Geschwindigkeit gearbeitet. Die experimentellen Ergebnisse befinden sich im Verzeichnis MAX_VEL auf der beigelegten DVD.

Abschließend blieb noch zu klären, ob wenige Partikel mit vielen Iterationen oder viele Partikel mit weniger Iterationen effizientere Optimierungen erlauben. Allerdings muss darauf geachtet werden, dass die Gesamtanzahl an Iterationen für alle Tests gleich ist, da mehr Iterationen automatisch zu besseren Ergebnissen führen. Bei allen vorhergehenden Experimenten wur-

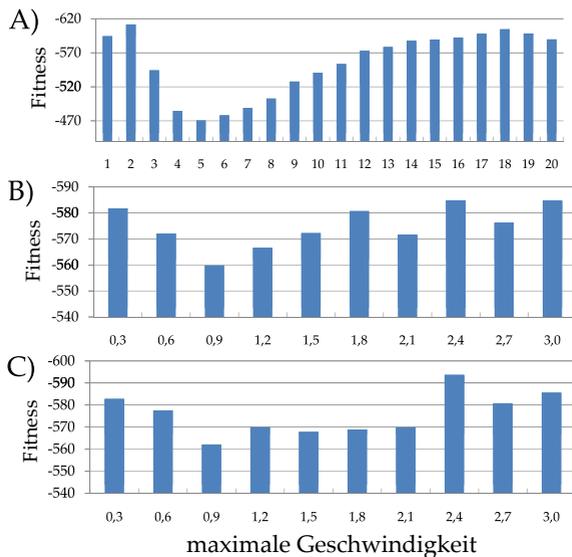


Abb. 3.15: Effizienz der Optimierung in Abhängigkeit von der maximalen Geschwindigkeit. A) PV_{max} in $\frac{\text{Å}}{\text{Schritt}}$; B) AV_{max} in $\frac{\text{rad}}{\text{Schritt}}$; C) OV_{max} in $\frac{\text{rad}}{\text{Schritt}}$

de eine Anzahl von 3 Millionen Iterationen pro Einzelexperiment verwendet und auch in dieser Untersuchung wurde für alle Einzelexperimente diese Begrenzung gewählt. Es wurden verschieden Kombinationen aus Partikelanzahl und Iterationen gebildet und getestet. Die Ergebnisse der Untersuchung sind in Abb. 3.16 dargestellt. Es ist klar erkennbar, dass mit abnehmender Iterationsanzahl und steigender Partikelanzahl die Effizienz der Optimierung abnimmt. Allerdings muss auch eine gewisse Anzahl an Partikeln vorhanden sein, damit die Effekte der Schwarmuche die Effizienz verbessern. In diesem Fall liegt das gefundene Optimum bei 20 Partikeln. Die experimentellen Daten dieses Versuches befinden sich im Verzeichnis `PSO_ITER_COUNT` auf der beigelegten DVD. Als letzte Fragestellung der Parametrisierung des PSO wurde untersucht, ob die Verwendung von Teilen des Schwarmes als Nachbarschaft die Effizienz der Optimierung verbessert. Die Untersuchung wurde an einer Schwarmgröße von 20 Partikeln durchgeführt. Es wurden Nachbarschaften von 20, 16, 12, 8 und 4 Partikeln untersucht. Das Ergebnis ist in Abb. 3.17 dargestellt. Für einen Schwarm mit 20 Partikeln konnte keine Verbesserung durch eine kleinere Nachbarschaft festgestellt werden. Deshalb wurde in allen weiteren Untersuchungen der gesamte Schwarm als Nachbar-

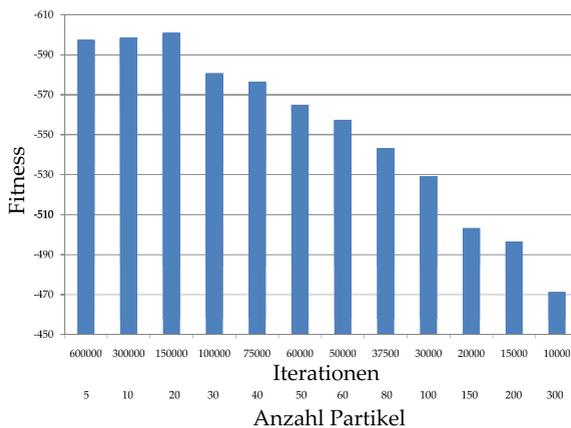


Abb. 3.16: Auswirkung der Partikelanzahl und Anzahl der Wiederholungen auf die Effizienz der Optimierung.

3 Ergebnisse und Diskussion

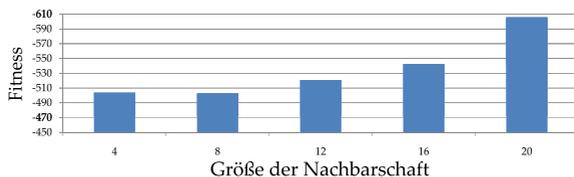


Abb. 3.17: Auswirkung der Größe der Nachbarschaft auf die Effizienz der Optimierung.

schaft gewählt. Die Ergebnisse der einzelnen Dockingexperimente befinden sich im Verzeichnis `MH_SIZE` auf der DVD.

Abschließend muss noch erwähnt werden, dass eine erneute Parametrisierung möglicherweise andere Parameter liefern würde, wenn die Analyse der Parameter in einer anderen Reihenfolge durchgeführt wird. Diese anderen Parameter könnten sogar besser sein als die hier gefundenen Parameter:

1. $I = 150000$
2. $N = 20$
4. $w_{start} = 1$
5. $w_{end} = 0,2$
6. $c_c = 1$
7. $c_s = 3,4$.

Außerdem gelten diese Parameter nur für 3 Millionen Iterationen pro Einzelexperiment. Da eine gute globale Optimierung auf dem Gleichgewicht zwischen Diversifizierung und Intensivierung beruht, ist die Wahl der Parameter ein sehr sensibler Teil des Dockingprogrammes. Schon die Wahl eines falschen Parameters kann die Effizienz der Optimierung stark negativ beeinflussen. Bisher wurde die Effizienz der Optimierung ausschließlich am numerischen Wert der Fitness gemessen. Um zu zeigen, dass verbesserte Fitnesswerte mit besseren Strukturen einhergehen, wurden die RMSD-Werte eines Dockings mit den Startparametern und die RMSD-Werte eines Dockings mit den optimierten Parametern in einem Histogramm miteinander verglichen (s. Abb. 3.18). Es ist klar erkennbar, dass deutlich mehr Ergebnisse mit niedrigen RMSD-Werten mit den optimierten Parametern gefunden werden, obwohl der RMSD-Wert bei der Untersuchung der Parameter nicht mit unter-

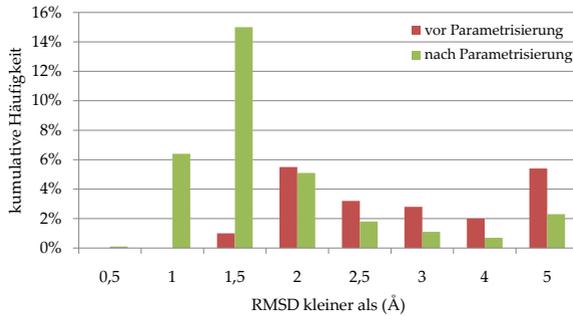


Abb. 3.18: Vergleich der Dockinggenauigkeit vor der Optimierung der PSO Parameter und danach.

sucht wurde.

3.2.2 Parametrisierung von p-Score

Als Trainingsdatensatz für den p-Score wird das *Astex Diverse Set* verwendet. Ein systematisches Vorgehen, ähnlich der Parametrisierung des PSO, ist für die Auffindung guter Parameter der Fitnessfunktion nicht möglich. Die Möglichkeiten sind zu vielfältig und alle Parameter hängen unmittelbar zusammen. Es musste eine andere Methode gewählt werden. In einer lokalen Optimierung wurden die Parameter der Fitnessfunktion, ausgehend von den Parametern des X-SCORE, solange zufällig um kleine Beträge geändert, bis keine weitere Verbesserungen erzielt werden konnten. Um eine Verbesserung feststellen zu können, wird ein Maß für die Güte der Fitnessfunktion benötigt. Optimalerweise würde der durchschnittliche RMSD-Wert der besten Lösungen des Trainingsdatensatzes aus einem umfangreichen Docking als Maß dienen. Der Trainingsdatensatz umfasst 85 Komplexe. Ein umfangreiches Docking sollte mindestens 200 Wiederholungen umfassen, um eine sehr gute Lösung zu finden. Pro Einzelexperiment sind ca. 3 Millionen Evaluationen der Fitnessfunktion nötig. Somit sind pro Parametersatz der Fitnessfunktion 51 Mrd. Evaluationen mit einer Dauer von je ca. 10^{-6} s nötig. Dies entspricht rund 6 CPU-Tagen pro Parametersatz und ist nicht mehr praktisch durchführbar, selbst wenn man mit mehreren CPUs arbeitet.

3 Ergebnisse und Diskussion

Das Ziel der Parametrisierung einer Bewertungsfunktion ist eine Funktion, die die Kristallstruktur besser bewertet als alle anderen Möglichkeiten. Ein Maß für die Genauigkeit der Funktion ist somit der Quotient aus Komplexen, die besser bewertet werden als die Kristallstruktur, und Komplexen, die schlechter bewertet werden als die Kristallstruktur. Je kleiner dieser Quotient ist, um so besser ist die Bewertungsfunktion. Die möglichen, aber falschen Komplexe zur Berechnung des Quotienten wurden aus falschen Dockinglösungen und aus leichten Veränderungen der Position des Liganden der Kristallstruktur gewonnen. Für jede Struktur im Trainingsdatensatz wurden 50 Komplexe aus der leicht veränderten Kristallstruktur mit RMSD-Werten kleiner als 2 \AA gebildet. Zusätzlich wurden die 30 besten Lösungen aus verschiedenen Dockingexperimenten mit den Ausgangsparametern, deren RMSD-Wert größer als 2 \AA ist verwendet. Somit fielen pro getestetem Parametersatz nur noch 80 Evaluationen für die falschen Komplexe und eine Evaluation für die Kristallstruktur pro Datensatz an.

Die Parameteroptimierung erfolgte durch folgende Vorgehensweise:

1. Die Ausgangsparameter werden gespeichert.
2. Die Ausgangsparameter werden durch Addition kleiner positiver oder negativer Zufallswerte verändert.
2. Die Fitness aller Komplexe wird berechnet.
4. Der Quotient $\frac{\text{Anzahl besser als Kristallstruktur bewerteter Komplexe}}{\text{Anzahl schlechter als Kristallstruktur bewerteter Komplexe}}$ wird für jeden Datensatz berechnet.
5. Das arithmetische Mittel aller Quotienten wird gebildet.
6. Wenn das arithmetische Mittel kleiner ist als das vorhergehende, werden die Ausgangsparameter durch die neuen Parameter ersetzt.
7. Wenn das arithmetische Mittel größer ist als das vorhergehende, werden die neuen Parameter verworfen.
8. Wenn das Zeitlimit nicht erreicht ist, wird wieder mit 1. begonnen.

Diese Berechnung wurde 5 mal mit je 1500 CPU-Stunden durchgeführt. Die ermittelten Quotienten sind 0,051; 0,055; 0,067; 0,080 und 0,086. Dies bedeutet, dass 4,9 % bis 8,2 % aller Strukturen besser als die Kristallstruktur bewertet werden. Der Parametersatz mit dem niedrigsten Quotienten wurde als Ergebnis der Parametrisierung ausgewählt. Falls ein Parameter im Trainingsdatensatz nicht benutzt wird, wurde dieser von der Optimierung ausgeschlossen.

Tab. 3.2: vdW-Typen und Radien in p-Score.

Typ	Radius in Å
C tet	2,36682
C tri ar	2,15347
C tri re	1,88582
C tri	2,23108
C lin	1,85249
N tet	1,77486
N tri ar	1,54659
N tri re	1,83274
N tri	1,75951
N lin	1,54278
O tet	1,79145
O tri ar	1,93169
O tri re	1,69397
O tri	1,74588
S tet	2,62296
S tri ar	1,91157
S tri	2,03912
P	2,00623
F	1,70554
Cl	1,71643
Br	1,76647
I	2,05 *
Si	2,0 *
K	2,0 *
Na	2,0 *
Ca	1,18868
Fe	0,793814
Mg	0,925381
Mn	1,06108
Zn	0,9429

*) im Trainingsdatensatz nicht enthalten

Die ermittelten vdW-Parameter der verschiedenen Atomtypen sind in Tabelle 3.2 aufgelistet. Der ermittelte Skalierungsfaktor für den Wasserstoffbrückenterm beträgt $-14,6$. Die Parameter der Abstandsfunktion des Wasserstoff-

brückentermes lauten

$$f(d_{ij}) = \begin{cases} 1 & d_{ij} \leq (d_{0ij} - 0,91\text{\AA}) \\ (1/0,91) \cdot (d_{0ij} - 0,91) & (d_{0ij} - 0,91\text{\AA}) < d_{ij} \leq d_0 \\ 0 & d_{ij} > d_0 \end{cases}$$

Die Parameter der Winkelfunktion des Wasserstoffbrückentermes lauten

$$f(\theta) = (1/180) \cdot (180 - \theta) \quad ,$$

wobei θ die Abweichung von der optimalen Wasserstoffbrückengeometrie ist. Auch hier gilt, dass die gefundenen Parameter nicht die besten sein müssen, da die Bestimmung der Parameter durch eine Heuristik erfolgt. Aufgrund der aufwendigen Berechnungen konnten jedoch nicht mehr Wiederholungen durchgeführt werden.

Alle verwendeten Komplexe befinden sich in vorprozessierter Form im Verzeichnis `PSCORE_OPT` auf der beigelegten DVD. Der Quellcode des Programmes zur Parameteroptimierung befindet sich in der Datei `paradocks/test/optimize_parameter.cpp`. Eine Wiederholung der Experimente mit identischem Ergebnis ist jedoch aufgrund des Zufallsuche-Algorithmus nicht möglich. Eine Wiederholung der Experimente kann sowohl bessere als auch schlechtere Parametersätze erzeugen. Komplett lösbar ist diese Aufgabenstellung jedoch nicht. Eine Vergrößerung des Trainingsdatensatzes und eine Verwendung von mehr Komplexen pro Datensatz könnte jedoch die Genauigkeit verbessern. Dies gilt besonders für Atomtypen, die nur selten im Trainingsdatensatz vorkommen.

3.3 Genauigkeit des Dockingprogrammes

In dieser Untersuchung wurde überprüft, wie genau bekannte Kristallstrukturen reproduziert werden können. Diese Aufgabe spiegelt das Einsatzgebiet eines Dockingprogrammes im Gebiet des Wirkstoffdesigns wider, auch wenn normalerweise Moleküle gedockt werden, für die keine bekannte Komplexstruktur existiert. Durch den Vergleich mit der experimentell ermittelten Struktur kann die Genauigkeit der Vorhersagen überprüft werden.

Als Testdatensatz diente die *PDBbind*-Datenbank. Verglichen mit dem Trainingsdatensatz *Astex Diverse Set* gibt es nur vier Komplexe, die in beiden Datensätzen enthalten sind. Diese vier Datensätze wurden aus dem Testdatensatz *PDBbind* entfernt. Ebenfalls entfernt wurden Komplexe, die Peptidliganden mit mehr als 4 Aminosäuren enthielten, und Komplexe mit besonders kleinen Liganden (Molmasse < 150 Da). Diese Liganden unterscheiden sich stark von denen des Trainingsdatensatzes und sind keine Arzneistoffe. Die verbliebenen 173 Komplexe befinden sich fertig vorbereitet im Verzeichnis `TESTSET/pdbbind_core` auf der DVD.

Die Komplexe wurden manuell vorbereitet. Nachdem die Komplexe aus der PDB-Datenbank heruntergeladen wurden, wurden sie mit dem Programm MOE bearbeitet. Es wurden alle Moleküle, die mehr als 4,5 Å vom Liganden entfernt waren, aus der Struktur gelöscht. Der verbleibende Komplex wurde mit der `Protonate3D`-Funktion protoniert. Falls eine funktionale Gruppe des Liganden oder Proteins neutral oder geladen vorliegen kann, optimiert `Protonate3D` das Wasserstoffbrückennetz und versucht sinnvolle Protonierungszustände zu erzeugen.

Obwohl oftmals als Vorbereitung eines Docking-Experiments alle Wassermoleküle aus der Bindetasche gelöscht werden, können Wassermoleküle für die Ausbildung eines Komplexes von großer Wichtigkeit sein. Die entropisch getriebene Verdrängung von Wassermolekülen durch den Liganden kann einen wichtigen Teil der Bindungsenergie des Komplexes liefern. In einem solchen Fall ist es notwendig die Wassermoleküle aus der Bindetasche des Proteins zu entfernen, da eine Verdrängung von Wassermolekülen durch heutige Dockingprogramme nicht simuliert wird. Wassermoleküle können aber auch direkt an der Bindung des Liganden beteiligt sein, indem sie Wasserstoffbrücken zum Protein und zum Liganden bilden. Mehrere Studien [79, 80, 81] haben gezeigt, dass die Verwendung benötigter expliziter Wassermoleküle während des Docking zu besseren Ergebnissen führt. Eine automatische Bearbeitung expliziter Wassermoleküle ist bisher nur wenig untersucht [79] und gehört zu den kommenden Herausforderungen auf dem Gebiet des automatischen Protein-Ligand-Docking. Ein Grund wichtige explizite Wassermoleküle während eines Dockings in der Bindetasche zu belassen, ist die Form des vdW-Potentials. Leere Räume, die durch das Entfernen des Wasser

entstehen, werden im Laufe der Optimierung immer durch andere Atome belegt werden, da das Optimum eines vdW-Potentials ein dicht gepacktes System ist.

Deshalb wurden in den Komplexen alle Kristallwasser belassen, die unmittelbar an der Bindung des Liganden beteiligt sind. Als unmittelbar an der Bindung beteiligt wurden alle Kristallwasser eingestuft, die tief in der Binde-tasche direkt zwischen Ligand und Protein liegen, und Wasserstoffbrücken zu Protein und Ligand bilden. Die Bearbeitung und Entscheidung erfolgte manuell während der visuellen Inspektion jeden Komplexes. Dabei wurden auch alle Atomtypen auf ihre Plausibilität überprüft und korrigiert. Alle übrigen Wassermoleküle wurden entfernt, der Ligand wurde extrahiert, zentriert und einzeln abgespeichert. Da weder PARADOCKS noch GOLD in der Lage sind, dreidimensionale Strukturen aus Konnektivitäten oder zweidimensionalen Daten zu erzeugen, wurden die Liganden in ihrer ursprünglichen Konformation belassen. Falls ein unbekanntes Molekül gedockt wird, ist die bioaktive Konformation nicht bekannt, und die dreidimensionalen Koordinaten müssen mit einem externen Programm erzeugt werden. Falls die erzeugte Konformation in einem rigiden Fragment falsche Geometrien aufweist, ist im Ergebnis des Dockings diese Konformation ebenfalls falsch. Es muss davon ausgegangen werden, dass unbekannte Liganden schlechter gedockt werden, als aus der Kristallstruktur extrahierte Liganden.

Als Vergleich wurde das Dockingprogramm GOLD in der Version 4 [82] benutzt. GOLD ist ein ausgereiftes und zuverlässiges Dockingprogramm. Andere Dockingprogramme wurden nicht mit in den Vergleich einbezogen, da schon eine Reihe von Vergleichsdaten zu GOLD in der Literatur beschrieben sind [83, 84]. Die Eingabedaten waren für PARADOCKS und das Vergleichsprogramm identisch. Für beide Programme wurden 50 Einzelexperimente pro Komplex durchgeführt. Für GOLD fanden die Standard GA-Parameter mit 100.000 Generationen Verwendung. Die Berechnungen mit PARADOCKS wurden mit den in den beiden vorhergehenden Kapiteln ausgearbeiteten Parametern durchgeführt.

Die Ergebnisse des Dockings sind in Abb. 3.19 dargestellt. Zur Auswertung wurden die Ergebnisse in Cluster mit dem Kriterium $\text{RMSD} < 2 \text{ \AA}$ eingeteilt. Der Kandidat mit der besten Fitness wurde als Vertreter dieses Clusters ge-

3.3 Genauigkeit des Dockingprogrammes

nutzt. Aufgetragen wurden die Häufigkeiten der RMSD-Werte aus dem Cluster mit der besten Fitness (1. Cluster). Üblicherweise wird als Grenze für ein erfolgreiches Docking ein RMSD-Wert von 2 Å angesehen. Es ergibt sich eine Rate von 54 % erfolgreicher Ergebnisse mit PARADOCKS und 62 % mit GOLD. Ebenfalls aufgetragen wurden die Ergebnisse mit dem niedrigsten RMSD-Wert des besten und zweitbesten Clusters zusammen (2. Cluster). Betrachtet man diese Ergebnisse gemeinsam, so findet PARADOCKS in 62 % der Fälle und GOLD in 69 % der Fälle eine erfolgreiche Lösung. Diese Statistik wurde auf die gemeinsame Betrachtung der besten fünf Cluster erweitert (5. Cluster). Bei Betrachtung der fünf besten Cluster jedes Komplexes wird mit beiden Programmen in 72 % der Fälle eine Lösung innerhalb von 2 Å gefunden. PARADOCKS ist fast so genau wie das Vergleichsprogramm GOLD, wobei p-Score noch Potential für Verbesserungen aufweist, da die Sortierung der Ergebnisse bei GoldScore etwas besser ist.

Eine Aufteilung des Testdatensatzes in Teildatensätze offenbarte die Ursache der etwas schlechteren Ergebnisse von PARADOCKS. Alle Komplexe wurden anhand der frei rotierbaren Bindungen des Liganden in drei verschiedene Gruppen eingeteilt. Im Testdatensatz gibt es 89 Datensätze mit bis zu fünf frei rotierbaren Bindungen, 46 Datensätze mit mehr als fünf und bis zu zehn frei rotierbaren Bindungen und 38 Datensätze mit mehr als zehn frei rotierbaren Bindungen

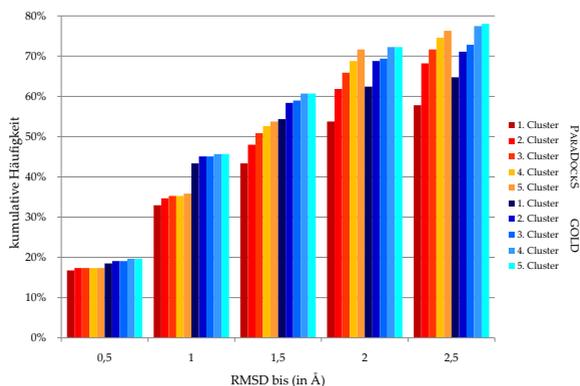


Abb. 3.19: Histogramm der Dockingergebnisse von PARADOCKS und GOLD.

3 Ergebnisse und Diskussion

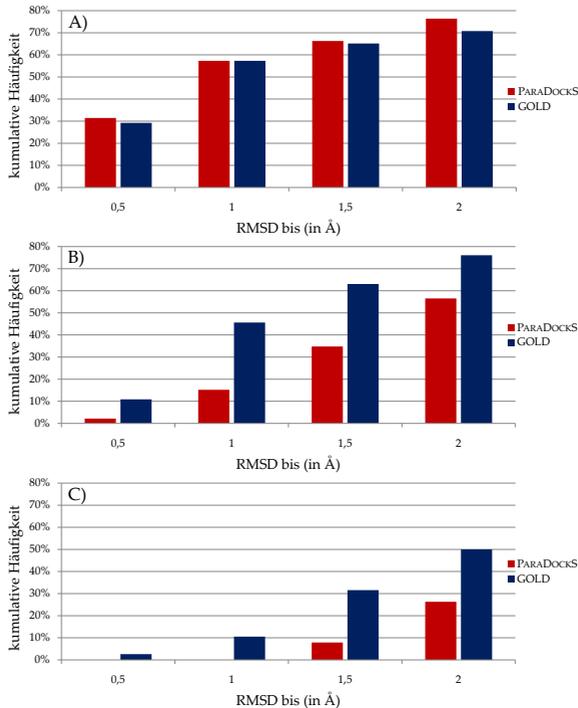


Abb. 3.20: Histogramm der Dockingergebnisse von PARADOCKS und GOLD. Die Komplexe wurden anhand der Anzahl der frei rotierbaren Bindungen des Liganden in Untergruppen unterteilt und getrennt analysiert. A) Liganden mit bis zu fünf frei rotierbaren Bindungen, B) Liganden mit mehr als fünf und bis zu zehn frei rotierbaren Bindungen, C) Liganden mit mehr als zehn frei rotierbaren Bindungen.

Bindungen. Die Ergebnisse der Analyse sind in Abb. 3.20 dargestellt. Wie zu erwarten war, sinkt mit zunehmender Anzahl an frei rotierbaren Bindungen die Genauigkeit des Dockings, da die Komplexität des Problemes zunimmt. Dieser Effekt ist jedoch bei PARADOCKS weitaus stärker ausgeprägt als bei GOLD. Während Liganden mit bis zu fünf frei rotierbaren Bindungen von PARADOCKS geringfügig genauer gedockt werden als von GOLD, lässt die Genauigkeit bei mehr als fünf frei rotierbaren Bindungen deutlich nach.

Die Schwäche beim Docking flexibler Liganden ist durch die Bewertungsfunktion erklärbar. Mit zunehmender Anzahl an rotierbaren Bindungen, wird die Energiehyperfläche aller Konformationen eines Liganden immer vielfältiger. Somit nimmt der Einfluss der ligandinternen Wechselwirkungen auf die Gesamtenergie des Komplexes zu. Der einfache Strafterm für *vdW-clashes* in p-Score kann diese Effekte nicht hinreichend genau beschreiben. GOLD hingegen benutzt statistische Verteilungen von Torsionswinkeln aus bekannten Kristallstrukturen um die Anzahl der Ligandenkonformationen einzugrenzen und besitzt ein ligandinternes vdW-Potential und ein ligandinternes Torsionswinkelpotential. Um auch Liganden mit vielen frei rotierbaren Bindungen zuverlässig docken zu können, muss die Bewertung der Ligandenkonformation in p-Score erweitert werden. Eine mögliche Erweiterung wäre die Verwendung eines Diederpotentials aus einem Kraftfeld für die frei rotierbaren Bindungen. Somit würde die Konformationsenergie des Liganden ebenfalls mit berücksichtigt werden. Eine weitere Verbesserung könnte die Einschränkung der Diederwinkel auf Bereiche, die experimentell häufig gefunden werden, bewirken. Dieser Ansatz lässt sich allerdings nur schwer mit der kontinuierlichen Suche eines PSO kombinieren. Ebenfalls nicht in p-Score enthalten sind die Modellierung ligandinterner Wasserstoffbrücken und elektrostatischer WW, die bei größeren Liganden mit vielen Freiheitsgraden immer mehr Bedeutung gewinnen. Eine Einführung dieser Terme könnte das Ergebnis ebenfalls verbessern.

3.4 Eignung für VS-Experimente

Im Verlauf eines strukturbasierten VS wird Docking zur Erzeugung von möglichen Protein-Ligand-Komplexen verwendet. Für die erzeugten Komplexe wird durch eine Bewertungsfunktion die Aktivität vorhergesagt. Eine Sortierung der vorhergesagten Aktivitäten sollte eine Anreicherung aktiver Substanzen im Bereich hoher Aktivität ergeben. Darüberhinaus kann die Fitness der erzeugten Komplexe direkt zur Sortierung herangezogen werden. Obwohl die Fitnessfunktion nicht für die Vorhersage von Aktivitäten entwickelt wurde, zeigen diese Ergebnisse oft eine Anreicherung aktiver Substanzen. Deshalb wurde die Eignung von PARADOCKS für VS-Experimente in diesem

Abschnitt überprüft.

Die Untersuchung erfolgte an drei verschiedenen Zielstrukturen mit aktiven und inaktiven Verbindungen wie in den Kapiteln 2.1.3, 2.1.4 und 2.1.5 beschrieben. Das Docking erfolgte mit den in den Kapiteln 3.2.1 und 3.2.2 vorgestellten Parametern. Die Auswertung erfolgte durch ROC-Kurven. Eine ROC-Kurve (*receiver operating characteristic*) ist die graphische Auswertung von Sensitivität und Spezifität einer binären Klassifizierung. Die Kurve gibt Aufschluss über die Trennschärfe des Klassifizierungsverfahren. Der Vorteil einer ROC-Kurve ist, dass die graphische Darstellung unabhängig von der Anzahl der aktiven und inaktiven Verbindungen ist. Dies ist für eine einfache Anreicherungskurve nicht der Fall. In einer ROC-Kurve wird die

$$\text{Sensitivität} = \frac{\text{Anzahl richtig positiv}}{\text{Anzahl richtig positiv} + \text{Anzahl falsch negativ}}$$

gegen

$$1 - \text{Spezifität}$$

mit

$$\text{Spezifität} = \frac{\text{Anzahl richtig negativ}}{\text{Anzahl richtig negativ} + \text{Anzahl falsch positiv}}$$

aufgetragen. Die Klassifizierung in positiv und negativ erfolgt anhand eines bestimmten Fitnesswertes. Die Kurve ergibt sich aus den Wertepaaren von Sensitivität und Spezifität aller sinnvollen Klassifizierungen.

Eine Kurve über der Ursprungsgeraden mit dem Anstieg eins zeigt eine Trennschärfe besser als eine zufällige Auswahl an. Die Fläche unter der Kurve entspricht der Trennschärfe des Klassifizierungsverfahrens. In Abb 3.21 sind die Ergebnisse des VS der drei ausgewählten Datensätze dargestellt. Die Fläche unter der Kurve (*AUC-area under the ROC curve*) wurde mit der gaußschen Trapezformel bestimmt. Das VS ergab die Resultate $AUC_{ER} = 0,841$, $AUC_{AChE} = 0,867$ und $AUC_{PRMT1} = 0,781$. Eine AUC von 0,5 würde einer zufälligen Klassifizierung entsprechen und ein AUC von 1,0 wäre eine optimale Klassifizierung. Die Ergebnisse spiegeln somit eine Anreicherung aktiver Substanzen wider.

Um diese Ergebnisse einordnen zu können, wurde wiederum das Programm GOLD wie in Kapitel 3.3 angegeben als Vergleich genutzt. Abb 3.22 stellt

die Ergebnisse als ROC-Kurve dar. Die Ergebnisse sind $AUC_{ER} = 0,449$, $AUC_{AChE} = 0,801$ und $AUC_{PRMT1} = 0,837$. Besonders hervorzuheben sind die schlechten Ergebnisse mit GoldScore für das VS am Estrogenrezeptor. Das erzielte Ergebnis entspricht ungefähr einer Zufallsverteilung. Die GoldScore-Funktion beruht auf polaren Wechselwirkungen und funktioniert erfahrungsgemäß schlechter an hydrophoben Bindetaschen. Die Vorhersage der Aktivität scheitert im Fall der stark hydrophoben Bindetasche des Estrogenrezeptors. Die ebenfalls in GOLD enthaltene Fitnessfunktion ChemScore liefert andererseits für den Estrogenrezeptor die beste Klassifizierung aller untersuchten Datensätze mit $AUC_{AChE} = 0,94$. Die Ergebnisse am AChE-Datensatz und am PRMT1-Datensatz sind ungefähr vergleichbar, mit leichten Vorteilen für GOLD im Falle der PRMT1 und leichten Vorteilen für PARADOCKS im Falle der AChE.

Das Verhalten von GOLD am ER-Datensatz unterstreicht deutlich, dass es sinnvoll ist, mehr als ein Bewertungsmodell zur Verfügung zu haben. Da Bewertungsfunktionen verschiedene Aspekte der Bindung eines Liganden unterschiedlich stark betonen, funktionieren sie, je nach Schlüsselinteraktionen in der Bindetasche, unterschiedlich gut. Das offene Design und die leichte Erweiterbarkeit von PARADOCKS trägt dieser Notwendigkeit Rechnung. Am Beispiel des PMF04-Potentials [47] konnte die leichte Erweiterbarkeit von PARADOCKS bereits in einer Diplomarbeit [85] gezeigt werden.

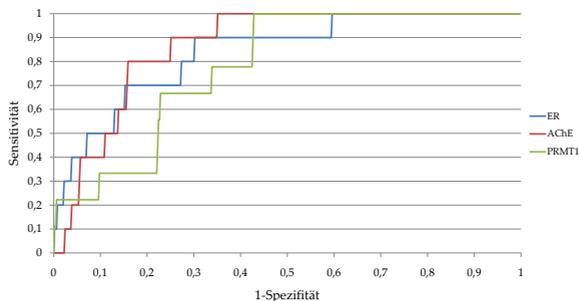


Abb. 3.21: Anreicherung der aktiven Verbindungen mit PARADOCKS dargestellt als ROC-Kurve.

3 Ergebnisse und Diskussion

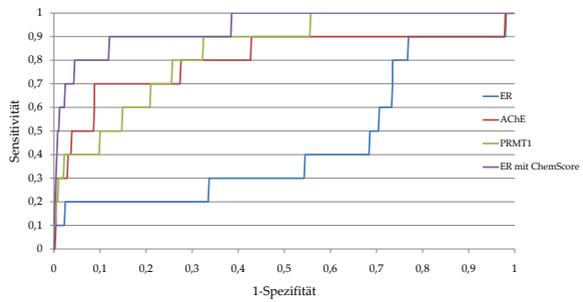


Abb. 3.22: Anreicherung der aktiven Verbindungen mit GOLD dargestellt als ROC-Kurve.

4 Zusammenfassung und Ausblick

In dieser Arbeit wurde die Entwicklung des neuen Dockingprogrammes PARADOCKS vorgestellt. PARADOCKS ist ein quelloffenes, portables und leicht zu erweiterndes Dockingprogramm. In der hier vorgestellten Form beruht das Programm auf der Optimierung einer empirischen Bewertungsfunktion durch die populationsbasierte Optimierungsmethode Partikelschwarm-Optimierung. Das Design des Programmes ermöglicht ein problemloses Austauschen von Bewertungsfunktion und Optimierungsmethode. Anhand der Reproduktion der Komplexe des Testdatensatzes *PDBbind* mit 173 Strukturen konnte belegt werden, dass die verwendeten Komponenten in der Lage sind, Liganden mit einer Anzahl von bis zu zehn frei rotierbaren Bindungen genau so gut wie das Vergleichsprogramm GOLD vorherzusagen. Für 70 % aller Komplexe, deren Liganden bis zu zehn frei rotierbare Bindungen haben, konnte die Lösung mit weniger als 2 Å RMSD vorhergesagt werden. An drei verschiedenen Datensätzen wurde eine Anreicherung aktiver Verbindungen in einem virtuellen Screening nachgewiesen. Die erhaltene Anreicherung ist gleich oder nur wenig schlechter als die des Vergleichsprogrammes. Da die Verarbeitung der Eingabedaten voll automatisch geschieht, eignet sich PARADOCKS als Plattform für virtuelles Screening.

Die Entwicklung des Dockingprogrammes kann noch nicht als abgeschlossen angesehen werden. Die Erweiterung von p-Score um einen Term zur Bewertung der Ligandenkonformation könnte mit geringem Aufwand die Genauigkeit der Strukturvorhersage von großen Liganden verbessern. Dafür wäre z. B. die Integration der Diederpotentiale eines bestehenden Kraftfeldes, wie MMFF94 [86] oder AMBER [38], geeignet. Eine weitere Möglichkeit zur Verbesserung könnte die Einführung expliziter elektrostatischer Potentiale darstellen, die ebenfalls aus bestehenden Kraftfeldern übernommen werden könnten.

Durch das variable Design von PARADOCKS bieten sich eine Reihe von Er-

4 Zusammenfassung und Ausblick

weiterungen und Weiterentwicklungen an. Als noch zu lösende Herausforderungen auf dem Gebiet der Vorhersage von Protein-Ligand-Komplexen verbleiben die Modellierung von Wassermolekülen in der Bindetasche und die Einbeziehung der Flexibilität des Proteins. Die Modellierung von Wassermolekülen in der Bindetasche könnte über die Optimierung der Anzahl und Position expliziter Wassermoleküle oder durch eine Erweiterung der Bewertungsfunktion geschehen. Eine solche Erweiterung müsste Kavitäten in der richtigen Größe und an der korrekten Position mit einem Bonus bewerten und so Wassermoleküle implizit modellieren können. Proteinflexibilität auf Basis weniger flexibler Aminosäureseitenketten könnte relativ einfach in das bestehende Programm integriert werden, indem die frei rotierbaren Bindungen der Aminosäureseitenketten in derselben Weise optimiert werden wie die frei rotierbaren Bindungen des Liganden. Voraussetzung für diese Vorgehensweise ist eine Bewertungsfunktion, die auch die Konformation der Aminosäureseitenketten bewerten kann. Eine Betrachtung eines voll flexiblen Protein wird auf absehbare Zeit mit diesen Methoden nicht möglich sein. Die kombinatorische Explosion der Möglichkeiten ist mit aktueller Rechentechnik nicht zu bewältigen.

Eine Weiterentwicklung sollte aber nicht nur neue Funktionen beinhalten. Wichtig ist auch eine Verbesserung der Genauigkeit der Strukturvorhersage und der Aktivitätsvorhersage. Die Genauigkeit der Strukturvorhersage einer empirischen Bewertungsfunktion lässt sich möglicherweise durch die Einführung von Straftermen für negative WW verbessern. Im Moment beruht die Strukturvorhersage auf der Maximierung positiver WW. Wenn dabei auch eine Vielzahl negativer WW gebildet wird, hat das keine Auswirkungen auf das Ergebnis. Eine ausgewogene Mischung aus positiven und negativen WW sollte die Genauigkeit verbessern und vor allem falsche Lösungen schlechter bewerten können. Natürlich kommen auch andere Bewertungsfunktionen in Frage. Eine Funktion auf Basis eines Kraftfeldes stellt eine vielversprechende Alternative dar. Vor allem in Verbindung mit der Berechnung von Solvationseffekten sollte mit einer Kraftfeldfunktion die Aktivitätsvorhersage zu verbessern sein.

Ein weiteres interessantes Gebiet sind gemischte Bewertungsfunktionen, die z. B. Pharmakophorhypothesen oder Abstandsbegrenzungen von Atomen

oder Gruppen schon während des Dockings als Nebenbedingungen einführen und somit die Verwendung einer Bindungshypothese ermöglichen. Da es relativ kompliziert ist, eine Wichtung von Bewertungsfunktion und Nebenbedingungen festzulegen, würde sich dafür als elegante Methode eine multikriterielle Optimierung zur Bestimmung der nicht-dominierten Lösungen anbieten. Mit einem Analyseprogramm könnte der Anwender dann einen Kompromiss aus der Menge der berechneten Lösungen auswählen.

Dieses Projekt kann nur erfolgreich fortgesetzt und verbessert werden, wenn die Anzahl der Entwickler und Nutzer vergrößert wird. Der Quellcode ist umfangreich dokumentiert und das Programm ist klar strukturiert und modular aufgebaut. Ein Einstieg ist somit einfach. Es bleibt zu hoffen, dass sich viele Wissenschaftler inspiriert fühlen einen Teil beizutragen. Das Programm und die Dokumentation werden unter <http://www.paradocks.org> veröffentlicht.

4 Zusammenfassung und Ausblick

A Datensätze

A.1 Astex Diverse Set

Die Strukturen sind verfügbar unter: http://www.ccdc.cam.ac.uk/products/life_sciences/gold/validation/downloads/download.php4 oder auf der beigelegten DVD unter: ASTEX_DIVERSE_SET.

Tab. A.1: Proteine und Liganden im *Astex Diverse Set*

PDB-ID	Protein	Ligand
1g9v	deoxy hemoglobin	RSR-13
1gkc	matrix metalloprotease 9	reverse hydroxamate inhibitor
1gm8	penicillin G acylase	PGSO
1gpk	acetylcholinesterase	huperzine A
1hnn	phenylethanolamine N-methyltransferase	SK&F 29661
1hp0	purine specific nucleoside hydrolase	3-deaza-adenosine
1hq2	6-hydroxymethyl-7,8-dihydropterin pyrophosphokinase	HP
1hvy	thymidylate synthase	tomudex
1hwi	HMG-CoA reductase	fluvastatin
1hww	R-mannosidase II	swainsonine
1ia1	dihydrofolate reductase	compound 3
1ig3	thiamin pyrophosphokinase	thiamin/vitamin B1
1j3j	dihydrofolate reductase	pyrimethamine
1jd0	carbonic anhydrase XII	acetazolamide
1lje	metallo β -lactamase	compound 11
1jla	HIV-1 reverse transcriptase	TNK-651
1k3u	tryptophan synthase	N-[1H-indol-3-yl-acetyl]aspartic acid
1ke5	cyclin-dependent kinase 2	compound 98
1kzk	HIV-1 protease	JE-2147/AG1776/KNI-764
1l2s	β -lactamase	compound 1
1l7f	neuraminidase A	BCX-1812
1lpz	factor Xa	compound 41
1lrh	auxin-binding protein 1	1-naphthalene acetic acid
1m2z	glucocorticoid receptor	dexamethasone
1meh	inosine monophosphate dehydrogenase	mycophenolic acid
1mmv	neuronal nitric-oxide synthase	NG-propyl-L-arginine
1mzc	protein farnesyltransferase	compound 33a

A Datensätze

PDB-ID	Protein	Ligand
1n1m	dipeptidyl peptidase IV	valine-pyrrolidine
1n2j	pantothenate synthetase	pantoate
1n2v	tRNA-guanine transglycosylase	compound 6
1n46	thyroid hormone receptor 1	compound 3
1nav	thyroid hormone receptor R1	compound 15
1of1	thymidine kinase	(S)-MCT
1of6	DAHPh synthase	tyrosine
1opk	c-Abl tyrosine kinase	PD166326
1oq5	carbonic anhydrase II	celecoxib
1owe	urokinase	compound 6
1oyt	thrombin	compound 4
1p2y	cytochrome P450cam	nicotine
1p62	deoxycytidine kinase	gemcitabine
1pmn	c-Jun terminal kinase 3	compound 1
1q1g	purine nucleoside phosphorylase	MT-ImmH
1q41	glycogen synthase kinase 3	indirubin-3'-monoxime
1q4g	prostaglandin H2 synthase 1	R-methyl-4-biphenylacetic acid
1r1h	neprilysin	compound 1
1r55	ADAM33	marimastat
1r58	methionine aminopeptidase 2	A357300
1r9o	cytochrome P450 2C9	flurbiprofen
1s19	vitamin D nuclear receptor	calcipotriol
1s3v	dihydrofolate reductase	compound 2
1sg0	quinone reductase 2	resveratrol
1sj0	estrogen receptor R	compound 4-D
1sq5	pantothenate kinase	pantothenate
1sqn	progesterone receptor	norethindrone
1t40	aldose reductase	IDD552
1t46	c-kit tyrosine kinase	gleevec
1t9b	acetohydroxyacid synthase	chlorsulfuron
1tow	adipocyte fatty acid-binding protein	compound 1
1tt1	glutamate receptor 6	kainate
1tz8	transthyretin	diethylstilbestrol
1u1c	uridine phosphorylase	5-benzyl-acyclouridine
1u4d	activated Cdc42 kinase 1	debromohymenialdisine
1uml	adenosine deaminase	compound 4c
1unl	cyclin-dependent kinase 5	(R)-roscovitine
1uou	thymidine phosphorylase	TPI
1v0p	protein kinase 5	purvalanol B
1v48	purine nucleoside phosphorylase	DFPP-G
1v4s	glucokinase	compound A
1vcj	neuraminidase B	BANA207
1w1p	chitinase B	cyclo-(gly-L-pro)
1w2g	thymidylate kinase	deoxythymidine
1x8x	tyrosyl-tRNA synthetase	L-tyrosine
1xm6	phosphodiesterase 4B	(R)-mesopram
1xoq	phosphodiesterase 4D	roflumilast
1xoz	phosphodiesterase 5A	tadalafil

A.2 PDBbind core set

Die Strukturen sind verfügbar unter:

<http://www.pdbbind.org>

oder auf der beigelegten DVD unter:

PDBbind/v2008.

Tab. A.2: PDB Code, Affinität und Name der Proteine in der *PDBbind* Datenbank

PDB code	Affinität	Protein
1a30	$K_i = 50\mu M$	HIV-1 protease
10gs	$K_i = 0.4\mu M$	glutathione s-transferase
1a08	$K_d = 2.4\mu M$	tyrosine kinase C-src
1a1b	$K_d = 0.4\mu M$	tyrosine kinase C-src
1a69	$K_i = 5\mu M$	purine nucleoside phosphorylase
1ai5	$K_i = 0.189mM$	penicillin amidohydrolase
1ajp	$K_i = 5.85mM$	penicillin amidohydrolase
1ajq	$K_i = 0.049mM$	penicillin amidohydrolase
1amw	$K_d = 29\mu M$	heat shock protein 90
1avn	$K_d = 0.125mM$	carbonic anhydrase ii
1b38	$K_d = 0.254\mu M$	cyclin dependent kinase 2
1b7h	$K_d = 0.0095\mu M$	oligo-peptide binding protein
1b8o	$K_i = 23pM$	purine nucleoside phosphorylase
1b9j	$K_d = 1.100\mu M$	oligo-peptide binding protein
1bcu	$K_d = 0.53mM$	thrombin alpha
1bgq	$K_d = 2.7nM$	heat shock protein 90
1bma	$K_i = 26\mu M$	elastase
1bra	$K_i = 15mM$	trypsin beta D189G G226D
1bxo	$K_i = 0.10nM$	penicillopepsin
1bxq	$K_i = 42nM$	penicillopepsin
1c1v	$K_i = 0.023\mu M$	thrombin alpha
1cps	$K_i = 0.22\mu M$	carboxypeptidase a
1d09	$K_i = 27nM$	aspartate carbamoyltransferase
1d7j	$K_d = 500\mu M$	FK506 binding protein
1df8	$K_d = 0.20nM$	streptavidin
1dhi	$K_d = 55nM$	dihydrofolate reductase
1e66	$K_i = 0.13nM$	acetylcholinesterase
1ela	$K_i = 0.44\mu M$	elastase
1elb	$K_i = 70nM$	elastase
1f4e	$K_i = 1.1mM$	thymidylate synthase
1f4f	$K_i = 24\mu M$	thymidylate synthase
1f4g	$K_i = 330nM$	thymidylate synthase

PDB code	Affinität	Protein
1f5k	$K_i = 180\mu M$	urokinase-type plasminogen activator
1fcx	$K_d = 64nM$	retinoic acid receptor gamma-1
1fcz	$K_d = 0.6nM$	retinoic acid receptor gamma-1
1fd0	$K_d = 4nM$	retinoic acid receptor gamma-1
1fh7	$K_i = 5.8\mu M$	xylanase beta-1,4
1fh8	$K_i = 0.13\mu M$	xylanase beta-1,4
1fh9	$K_i = 0.37\mu M$	xylanase beta-1,4
1fkb	$K_d = 0.2nM$	FK506 binding protein
1fki	$K_i = 100nM$	FK506 binding protein
1fkn	$K_i = 1.6nM$	secretase beta
1flr	$K_d = 0.1nM$	antibody fab
1ftm	$K_d = 24.8nM$	glutamate receptor 2
1fzj	$K_d = 8nM$	h-2 class i histocompatibility antigen
1fzk	$K_d = 4nM$	h-2 class i histocompatibility antigen
1g7f	$K_i = 3.4\mu M$	tyrosine phosphatase 1b
1g7q	$K_d = 877nM$	h-2 class i histocompatibility antigen
1gni	$K_d = 8.5nM$	serum albumin
1gpk	$K_i = 4.3\mu M$	acetylcholinesterase
1h23	$K_i = 4.5nM$	acetylcholinesterase
1ha2	$K_d = 2.9\mu M$	serum albumin
1hfs	$K_i = 2nM$	stromelysin-1
1hk4	$K_d = 4.9\mu M$	serum albumin
1hnn	$K_i = 0.58\mu M$	phenylethanolamine n-methyltransferase
1if7	$K_d = 30pM$	carbonic anhydrase ii
1is0	$K_d = 0.1\mu M$	tyrosine kinase C-src
1j16	$K_i = 143\mu M$	trypsin beta
1j17	$K_i = 6.05\mu M$	trypsin beta
1jaq	$K_i = 33\mu M$	matrix metalloproteinase-8
1jq8	$K_i = 1.01\mu M$	phospholipase a2
1jys	$K_i = 300\mu M$	mta/sah nucleosidase
1k4g	$K_i = 1.4\mu M$	tRNA-guanine transglycosylase
1k9s	$K_i = 0.30\mu M$	purine nucleoside phosphorylase
1kv1	$K_d = 1.16\mu M$	mitogen-activated protein kinase p38
1sgu	$K_i = 4.235\mu M$	pol polyprotein
1v2o	$K_i = 18.45\mu M$	trypsin beta
2arm	$K_i = 7.4nM$	phospholipase a2 vrv-pl-viii
1l2s	$K_i = 26\mu M$	beta-lactamase
1lag	$K_d = 500nM$	diga16
1lah	$K_d = 30nM$	diga16
1lol	$K_i = 0.41\mu M$	orotidine 5'-monophosphate decarboxylase
1loq	$K_i = 0.2mM$	orotidine 5'-monophosphate decarboxylase
1lst	$K_d = 14nM$	lysine, arginine, ornithine-binding protein

A Datensätze

PDB code	Affinität	Protein
1m0n	$K_i = 6.0mM$	2,2-dialkylglycine decarboxylase
1m0q	$K_i = 0.13mM$	2,2-dialkylglycine decarboxylase
1m2q	$K_i = 0.80\mu M$	casein kinase ii
1mq6	$K_i = 7pM$	coagulation factor xa
1n2v	$K_i = 83\mu M$	queuine tRNA-ribosyltransferase
1n5r	$K_i = 2.2\mu M$	acetylcholinesterase
1nc1	$K_i = 0.75\mu M$	mta/sah nucleosidase
1ndw	$K_i = 5900nM$	adenosine deaminase
1ndy	$K_i = 680nM$	adenosine deaminase
1ndz	$K_i = 7.7nM$	adenosine deaminase
1nfy	$K_i = 1.3nM$	coagulation factor xa
1nhu	$K_i = 2.2\mu M$	hepatitis c virus ns5b RNA polymerase
1nja	$K_d = 0.49\mu M$	thymidylate synthase
1nje	$K_d = 160\mu M$	thymidylate synthase
1nny	$K_i = 22nM$	tyrosine phosphatase 1b
1nvq	$K_i = 5.6nM$	serine/threonine-protein kinase chk1
1o0h	$K_i = 1.2\mu M$	ribonuclease a
1o3f	$K_i = 0.011\mu M$	trypsin beta
1o3p	$K_i = 0.22\mu M$	urokinase-type plasminogen activator
1ols	$K_d = 1.52\mu M$	2-oxoisovalerate dehydrogenase
1olu	$K_d = 39.3\mu M$	2-oxoisovalerate dehydrogenase
1om1	$K_i = 0.17\mu M$	casein kinase ii
1p1q	$K_d = 12.8\mu M$	glutamate receptor 2
1pb9	$K_i = 241\mu M$	glutamate [nmda] receptor
1pbq	$K_i = 0.54\mu M$	glutamate [nmda] receptor
1pr5	$K_i = 120\mu M$	purine nucleoside phosphorylase
1pxo	$K_i = 2.0nM$	cyclin dependent kinase 2
1pz5	$K_d = 4\mu M$	antibody fab
1q84	$K_i = 8.9pM$	acetylcholinesterase
1q8t	$K_d = 17.5\mu M$	cyclin dependent kinase 2
1re8	$K_i = 0.3nM$	cyclin dependent kinase 2
1s39	$K_i = 20nM$	tRNA-guanine transglycosylase
1sl3	$K_i = 1.4pM$	thrombin alpha
1slg	$K_d = 125\mu M$	streptavidin
1sqa	$K_i = 0.62nM$	urokinase-type plasminogen activator
1sv3	$K_i = 18\mu M$	phospholipase a2
1swr	$K_d = 0.12\mu M$	streptavidin core
1syh	$K_i = 487nM$	glutamate receptor 2
1tmn	$K_i = 50nM$	thermolysin
1toi	$K_d = 0.090mM$	aspartate aminotransferase
1toj	$K_d = 0.41mM$	aspartate aminotransferase
1tok	$K_d = 3.4mM$	aspartate aminotransferase

PDB code	Affinität	Protein
1trd	$K_i = 4\mu M$	triosephosphate isomerase
1tsy	$K_d = 11.0\mu M$	thymidylate synthase
1u1b	$K_d = 16nM$	ribonuclease a
1u2y	$K_i = 18mM$	alpha-amylase
1u33	$K_i = 25\mu M$	alpha-amylase
1utp	$K_d = 36mM$	trypsinogen
1uwt	$K_i = 1.08\mu M$	beta-galactosidase
1v16	$K_d = 134.8\mu M$	2-oxoisovalerate dehydrogenase
1v48	$K_i = 16nM$	purine nucleoside phosphorylase
1vfn	$K_i = 2.5\mu M$	purine nucleoside phosphorylase
1w0y	$K_i = 0.38\mu M$	coagulation factor viia
1x1z	$K_i = 8.8pM$	orotidine 5'-monophosphate decarboxylase
1x8r	$K_i = 750nM$	3-phosphoshikimate 1-carboxyvinyltransferase
1x8t	$K_i = 16nM$	3-phosphoshikimate 1-carboxyvinyltransferase
1xd1	$K_i = 0.012\mu M$	alpha-amylase
1xgj	$K_i = 1.0\mu M$	beta-lactamase
1y1m	$K_i = 15.3mM$	glutamate [nmda] receptor
1y6q	$K_i = 2pM$	mta/sah nucleosidase
1ydt	$K_i = 48nM$	cyclin dependent kinase 2
1zc9	$K_d = 0.6mM$	2,2-dialkylglycine decarboxylase
1zoe	$K_i = 40nM$	protein kinase ck2, alpha subunit
1zs0	$K_i = 700nM$	neutrophil collagenase
1zvx	$K_i = 0.6nM$	neutrophil collagenase
2azr	$K_i = 230\mu M$	tyrosine phosphatase, non-receptor type
2b1v	$K_i = 1.8\mu M$	estrogen receptor
2b7d	$K_i = 2nM$	coagulation factor vii
2baj	$K_d = 4.0nM$	mitogen-activated protein kinase 14
2bak	$K_d = 37nM$	mitogen-activated protein kinase 14
2bok	$K_i = 0.28\mu M$	coagulation factor xa
2brb	$K_i = 13.7\mu M$	serine/threonine-protein kinase chk1
2bz6	$K_i = 0.081\mu M$	coagulation factor viia
2c3j	$K_i = 0.659\mu M$	serine/threonine-protein kinase chk1
2ceq	$K_i = 53nM$	beta-galactosidase
2cer	$K_i \leq 0.6nM$	beta-glucosidase a
2cet	$K_d = 9.6nM$	beta-glucosidase a
2cgr	$K_d = 53nM$	antibody fab
2ctc	$K_i = 0.13mM$	carboxypeptidase a
2d0k	$K_d = 9.51\mu M$	dihydrofolate reductase
2d1o	$K_i = 0.02\mu M$	stromelysin-1
2d3u	$K_i = 0.12\mu M$	polyprotein
2d3z	$K_i = 0.23\mu M$	polyprotein
2drc	$K_d = 0.13nM$	dihydrofolate reductase

A Datensätze

PDB code	Affinität	Protein
2exm	$K_i = 78\mu M$	Cell division protein kinase 2
2f80	$K_i = 6.6nM$	pol polyprotein
2fdp	$K_i = 26nM$	beta-secretase 1
2fzc	$K_i = 1990\mu M$	aspartate carbamoyltransferase
2g71	$K_i = 35nM$	Phenylethanolamine N-methyltransferase
2g8r	$K_i = 103\mu M$	ribonuclease pancreatic
2g94	$K_i = 0.3nM$	beta-secretase 1
2gss	$K_i = 11.5\mu M$	glutathione s-transferase
2h3e	$K_d = 2\mu M$	aspartate carbamoyltransferase
2ha3	$K_d = 930\mu M$	Acetylcholinesterase
2hb3	$K_i = 4.5pM$	HIV-1 protease
2hdq	$K_i = 40mM$	beta-lactamase
2hs1	$K_i = 3.3nM$	HIV-1 protease v32i mutant
2hs2	$K_i = 4.9nM$	HIV-1 protease m46l mutant
2i0d	$K_i = 0.8pM$	HIV-1 protease
2i4x	$K_i = 1.9pM$	Protease(I84V, L90M)
2it4	$K_i = 24.1mM$	Carbonic anhydrase 1
2iwx	$K_d = 0.21\mu M$	ATP-dependent molecular chaperone HSP82
2j27	$K_i = 0.3mM$	triosephosphate isomerase glycosomal P168A
2j77	$K_d = 12.9\mu M$	beta-glucosidase a
2j78	$K_d = 384nM$	beta-glucosidase a
2jdn	$K_d = 2.78\mu M$	FUCOSE-BINDING LECTIN PA-III(S22A)
2jdu	$K_d = 0.19\mu M$	FUCOSE-BINDING LECTIN PA-III(G24N)
2jdy	$K_d = 42.9\mu M$	FUCOSE-BINDING LECTIN PA-III(G24N)
2nmx	$K_d = 2.7\mu M$	Carbonic Anhydrase I
2nn7	$K_d = 0.3\mu M$	Carbonic anhydrase 1
2obf	$K_i = 1.4nM$	Phenylethanolamine N-methyltransferase
2p3a	$K_i = 11nM$	Pol protein
2pog	$K_i = 0.29nM$	Estrogen receptor alpha
2pow	$K_i = 63nM$	Carbonic anhydrase 2
2qe4	$K_i = 11.0nM$	Estrogen receptor
2qfu	$K_i = 66\mu M$	3-phosphoshikimate 1-carboxyvinyltransferase
2qwb	$K_i = 1820\mu M$	neuraminidase
2qwd	$K_i = 14\mu M$	neuraminidase
2qwe	$K_i = 0.033\mu M$	neuraminidase
2rkm	$K_i = 125\mu M$	oligo-peptide binding protein
2std	$K_i = 0.14nM$	scytalone dehydratase
2usn	$K_i = 0.31\mu M$	stromelysin-1
2v00	$K_d = 0.22mM$	ENDOTHIAPEPSIN
2wec	$K_i = 110\mu M$	PENICILLOPEPSIN
3gss	$K_i = 1.5\mu M$	glutathione s-transferase
3pce	$K_i = 10mM$	protocatechuate 3,4-dioxygenase

PDB code	Affinität	Protein
3pcj	$K_d = 0.06\mu M$	protocatechuate 3,4-dioxygenase
3pcn	$K_d = 220\mu M$	protocatechuate 3,4-dioxygenase
3std	$K_i = 7.7pM$	scytalone dehydratase
4er2	$K_i = 0.5nM$	endothiapsin
4tim	$K_i = 6.9mM$	triosephosphate isomerase
4tln	$K_i = 190\mu M$	thermolysin
4tmn	$K_i = 0.068nM$	thermolysin
5er2	$K_i = 0.27\mu M$	endothiapsin
6cpa	$K_i = 3pM$	carboxypeptidase a
6std	$K_i = 2.3nM$	scytalone dehydratase

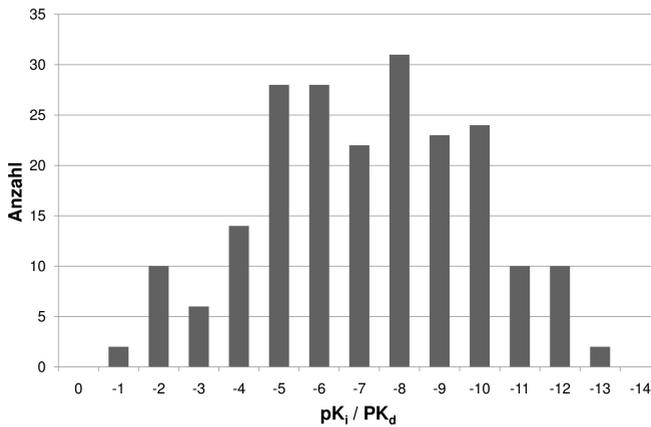


Abb. A.2: Verteilung der Affinitäten aller Komplexe des PDBbind core set

Literaturverzeichnis

- [1] DiMASI, JA ; HANSEN, RW ; GRABOWSKI, HG: The price of innovation: new estimates of drug development costs. In: *J. Health Econ.* 22 (2003), Nr. 2, S. 151–185
- [2] ADAMS, CP ; BRANTNER, VV: Estimating the cost of new drug development: Is it really \$802 million? In: *Health Aff.* 25 (2006), Nr. 2, S. 420–428
- [3] BAJORATH, F: Integration of virtual and high-throughput screening. In: *Nat. Rev. Drug Discov.* 1 (2002), Nr. 11, S. 882–894
- [4] KLEBE, Gerhard: Virtual ligand screening: strategies, perspectives and limitations. In: *Drug Discov. Today* 11 (2006), Nr. 13-14, S. 580–594
- [5] CRAMER, RD ; PATTERSON, DE ; BUNCE, JD: Recent advances in comparative molecular field analysis (CoMFA). In: *Prog. Clin. Biol. Res.* 291 (1989), S. 161–165
- [6] SHOICHET, BK ; KUNTZ, ID: Matching chemistry and shape in molecular docking. In: *Protein Eng.* 6 (1993), Nr. 7, S. 723–732
- [7] KUNTZ, ID ; BLANEY, JM ; OATLEY, SJ ; LANGRIDGE, R ; FERRIN, TE: A geometric approach to macromolecule-ligand interactions. In: *J. Mol. Biol.* 161 (1982), Nr. 2, S. 269–288
- [8] RAREY, M ; KRAMER, B ; LENGAUER, T ; KLEBE, G: A fast flexible docking method using an incremental construction algorithm. In: *J. Mol. Biol.* 261 (1996), Nr. 3, S. 470–489
- [9] JAIN, AN: Surfex: fully automatic flexible molecular docking using a molecular similarity-based search engine. In: *J. Med. Chem.* 46 (2003), Nr. 4, S. 499–511
- [10] ZSOLDOS, Z ; REID, D ; SIMON, A ; SADJAD, BS ; JOHNSON, AP: eHiTS: an innovative approach to the docking and scoring function problems. In: *Curr. Protein Pept. Sci.* 7 (2006), Oct, S. 421–435
- [11] KENNEDY, J ; EBERHART, R: Particle swarm optimization. In: *Proc. IEEE Int. Conf. on Neural Networks* 4 (1995), S. 1942–1948

- [12] FRIESNER, RA ; BANKS, JL ; MURPHY, RB ; HALGREN, TA ; KLICIC, JJ ; MAINZ, DT ; REPASKY, MP ; KNOLL, EH ; SHELLEY, M ; PERRY, JK ; SHAW, DE ; FRANCIS, P ; SHENKIN, PS: Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. In: *J. Med. Chem.* 47 (2004), Nr. 7, S. 1739–1749
- [13] HALGREN, TA ; MURPHY, RB ; FRIESNER, RA ; BEARD, HS ; FRYE, LL ; POLLARD, WT ; BANKS, JL: Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. In: *J. Med. Chem.* 47 (2004)
- [14] JONES, G ; WILLETT, P ; GLEN, RC ; LEACH, AR ; TAYLOR, R: Development and validation of a genetic algorithm for flexible docking. In: *J. Mol. Biol.* 267 (1997), Nr. 3, S. 727–748
- [15] MORRIS, GM ; GOODSSELL, DS ; HALLIDAY, RS ; HUEY, R ; HART, WE ; BELEW, RK ; OLSON, AJ: Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. In: *J. Comput. Chem.* 19 (1998), Nr. 14, S. 1639–1662
- [16] TAYLOR, JS ; BURNETT, RM: DARWIN: a program for docking flexible molecules. In: *Proteins* 41 (2000), Nov, S. 173–191
- [17] CHEN, HM ; LIU, BF ; HUANG, HL ; HWANG, SF ; HO, SY: SODOCK: Swarm optimization for highly flexible protein-ligand docking. In: *J. Comput. Chem.* 28 (2007), Nr. 2, S. 612–623
- [18] JANSON, S ; MERKLE, D ; MIDDENDORF, M: Molecular docking with multi-objective particle swarm optimization. In: *Appl. Soft. Comput.* 8 (2008), Nr. 1, S. 666–675
- [19] KORB, O ; STÜTZLE, T ; EXNER, TE: An ant colony optimization approach to flexible protein-ligand docking. In: *Swarm Intelligence* 1 (2007), Nr. 2, S. 115–134
- [20] MCGANN, MR ; ALMOND, HR ; NICHOLLS, A ; GRANT, JA ; BROWN, FK: Gaussian docking functions. In: *Biopolymers* 68 (2003), Nr. 1, S. 76–90
- [21] NELDER, JA ; MEAD, R: A Simplex Method for Function Minimization. In: *Comp. J.* 7 (1965), S. 308–313
- [22] KIRKPATRICK, S ; GELATT JR., CD ; VECCHI, MP: Optimization by Simulated Annealing. In: *Science* 220 (1983), S. 671–680
- [23] GLOVER, F: Tabu Search - Part I. In: *ORSA J. Comput.* 1 (1989), S. 190–206

- [24] GLOVER, F: Tabu Search - Part II. In: *ORSA J. Comput.* 2 (1990), S. 4–32
- [25] APOLLONI, B ; CARVALHO, C ; FALCO, D de: Quantum stochastic optimization. In: *Stoch. Proc. Appl.* 33 (1989), Dec, S. 233–244
- [26] DUECK, G ; SCHEUER, T: Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. In: *J. Comput. Phys.* 90 (1990), S. 161–175
- [27] FRASER, AS: Simulation of genetic systems by automatic digital computers. I. Introduction. In: *Australian J. Biol. Sci.* 10 (1957), S. 484–491
- [28] GOLDBERG, DE: Real-coded genetic algorithms, virtual alphabets, and blocking. In: *Complex Systems* 5 (1991), S. 139–167
- [29] GREFENSTETTE, JJ: Lamarckian learning in multi-agent environments. (1991), S. 303–310
- [30] DORIGO, M. ; MANIEZZO, V. ; COLORNI, A.: The Ant System: Optimization by a Colony of Cooperating Agents. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part B* 26 (1996), S. 29–41
- [31] CLERC, M ; KENNEDY, J: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. In: *IEEE Trans. Evolutionary Computation* 6 (2002), S. 58–73
- [32] EBERHART, RC ; SHI, Y: Comparing inertia weights and constriction factors in particle swarm optimization. In: *Proceedings of the Congress on Evolutionary Computing* (2000), S. 84–89
- [33] VENTER, G ; SOBIESZCZANSKI-SOBIESKI, J: A parallel particle swarm optimization algorithm accelerated by asynchronous evaluations. In: *Journal of Aerospace Computing, Information, and Communication* 3 (2006), S. 123–137
- [34] MUEGGE, I. ; RAREY, M.: Small Molecule Docking and Scoring. In: *Reviews in Computational Chemistry* 17 (2001), S. 1–60
- [35] BÖHM, HJ ; STAHL, M: The Use of Scoring Functions in Drug Discovery Applications. In: *Reviews in Computational Chemistry* 18 (2002), S. 41–87
- [36] MEHLER, EL ; SOLMAJER, T: Electrostatic effects in proteins: comparison of dielectric and charge models. In: *Protein Eng.* 4 (1991), Nr. 8, S. 903–910
- [37] NICHOLLS, A ; HONIG, B: A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation. In: *J. Comput. Chem.* 12 (1991), Nr. 4, S. 435–445

- [38] WEINER, SJ ; KOLLMAN, PA ; CASE, DA ; SINGH, UC ; GHIO, C ; ALAGONA, G ; PROFETA, S ; WEINER, P: A new force field for molecular mechanical simulation of nucleic acids and proteins. In: *JACS* 106 (1984), Nr. 3, S. 765–784
- [39] MENG, EC ; SHOICHET, BK ; KUNTZ, ID: Automated docking with grid-based energy evaluation. In: *J. Comput. Chem.* 13 (1992), Nr. 4, S. 505–524
- [40] BÖHM, HJ: The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure. In: *J. Comput. Aided Mol. Des.* 8 (1994), S. 243–256
- [41] BÖHM, HJ: The computer program LUDI: A new method for the de novo design of enzyme inhibitors. In: *J. Comput. Aided Mol. Des.* 6 (1992), S. 61–78
- [42] WANG, RX ; LAI, LH ; WANG, SM: Further development and validation of empirical scoring functions for structure-based binding affinity prediction. In: *J. Comput. Aided Mol. Des.* 16 (2002), Nr. 1, S. 11–26
- [43] GEHLHAAR, DK ; VERKHIVKER, GM ; REJTO, PA ; SHERMAN, CJ ; FOGEL, DB ; FOGEL, LJ ; FREER, ST: Molecular recognition of the inhibitor AG-1343 by HIV-1 protease: conformationally flexible docking by evolutionary programming. In: *Chem. Biol.* 2 (1995), S. 317–324
- [44] BERMAN, HM ; WESTBROOK, J ; FENG, Z ; GILLILAND, G ; BHAT, TN ; WEISSIG, H ; SHINDYALOV, IN ; BOURNE, PE: The Protein Data Bank. In: *Nucleic Acids Res.* 28 (2000), S. 235–242
- [45] MITCHELL, JBO ; LASKOWSKI, RA ; ALEX, A ; THORNTON, JM: BLEEP - potential of mean force describing protein-ligand interactions: I. Generating potential. In: *J. Comput. Chem.* 20 (1999), S. 1165–1176
- [46] MITCHELL, JBO ; LASKOWSKI, RA ; ALEX, A ; FORSTER, MJ ; THORNTON, JM: BLEEP - potential of mean force describing protein-ligand interactions: II. Calculation of binding energies and comparison with experimental data. In: *J. Comput. Chem.* 20 (1999), S. 1177–1185
- [47] MUEGGE, I: PMF scoring revisited. In: *J. Med. Chem.* 49 (2006), Nr. 20, S. 5895–5902
- [48] GOHLKE, H ; HENDLICH, M ; KLEBE, G: Knowledge-based scoring function to predict protein-ligand interactions. In: *J. Mol. Biol.* 295 (2000), Nr. 2, S. 337–356

- [49] HARTSHORN, MJ ; VERDONK, ML ; CHESSARI, G ; BREWERTON, SC ; MOOIJ, WT ; MORTENSON, PN ; MURRAY, CW: Diverse, high-quality test set for the validation of protein-ligand docking performance. In: *J. Med. Chem.* 50 (2007), Nr. 4, S. 726–741
- [50] SADOWSKI, J ; KUBINYI, H: A scoring scheme for discriminating between drugs and nondrugs. In: *J. Med. Chem.* 41 (1998), Nr. 18, S. 3325–3329
- [51] WANG, R ; FANG, X ; LU, Y ; WANG, S: The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. In: *J. Med. Chem.* 47 (2004), Nr. 12, S. 2977–2980
- [52] WANG, R ; FANG, X ; LU, Y ; YANG, CY ; WANG, S:
- [53] EVANS, RM: The steroid and thyroid hormone receptor superfamily. In: *Science* 240 (1983), Nr. 4854, S. 889–895
- [54] THOMSON SCIENTIFIC, Philadelphia, PA, USA: *World Drug Index*
- [55] *NCI 2D and 3D Structural Information*. http://dtp.nci.nih.gov/docs/3d_database/Structural_information/structural_data.html, . - zugegriffen am 30. Oktober 2008
- [56] IRWIN, JJ ; SHOICHET, BK: ZINC - A Free Database of Commercially Available Compounds for Virtual Screening. In: *J. Chem. Inf. Model.* 45 (2005), Nr. 1, S. 177–182
- [57] SIPPL, W ; CONTRERAS, JM ; PARROT, I ; RIVAL, YM ; WERMUTH, CG: Structure-based 3D QSAR and design of novel acetylcholinesterase inhibitors. In: *J. Comput.-Aided Mol. Des.* 15 (2001), Nr. 5, S. 395–410
- [58] CONTRERAS, JM ; PARROT, I ; SIPPL, W ; RIVAL, YM ; WERMUTH, CG: Design, Synthesis, and Structure-Activity Relationships of a Series of 3-[2-(1-Benzylpiperidin-4-yl)ethylamino]pyridazine Derivatives as Acetylcholinesterase Inhibitors. In: *J. Med. Chem.* 44 (2001), Nr. 17, S. 2707–2718
- [59] GEULA, C ; MESULAM, MM: Cholinesterases and the pathology of Alzheimer disease. In: *Alzheimer Dis. Assoc. Disord.* 9 (1995), S. 23–28
- [60] GIACOBINI, E: From molecular structure to Alzheimer therapy. In: *Jpn. J. Pharmacol.* 74 (1997), Nr. 3, S. 225–241
- [61] HUANG, S ; LITT, M ; FELSENFELD, G: Methylation of histone H4 by arginine methyltransferase PRMT1 is essential in vivo for many subsequent histone modifications. In: *Genes Dev.* 19 (2005), Nr. 16, S. 1885–1893

- [62] SPANNHOFF, A ; HEINKE, R ; BAUER, I ; TROJER, P ; METZGER, E ; GUST, R ; SCHÜLE, R ; BROSCHE, G ; SIPPL, W ; JUNG, M: Target-based approach to inhibitors of histone arginine methyltransferases. In: *J. Med. Chem.* 50 (2007), Nr. 10, S. 2319–2325
- [63] HEINKE, R ; SPANNHOFF, A ; MEIER, R ; TROJER, P ; BAUER, I ; JUNG, M ; SIPPL, W: Virtual Screening and Biological Characterization of Novel Histone Arginine Methyltransferase PRMT1 Inhibitors. In: *ChemMedChem* 4 (2009), Nr. 1, S. 69–77
- [64] *Xerces-C++*. <http://xerces.apache.org/xerces-c/>, . – zugegriffen am 30. Oktober 2008
- [65] *OpenSceneGraph*. <http://www.openscenegraph.org/projects/osg/>, . – zugegriffen am 30. Oktober 2008
- [66] *OpenMPI*. <http://www.open-mpi.org/>, . – zugegriffen am 30. Oktober 2008
- [67] *MPI-2: Extensions to the Message-Passing Interface*. <http://www.mpi-forum.org/docs/mpi-20-html/mpi2-report.html>, . – zugegriffen am 30. Oktober 2008
- [68] *Boost C++ Libraries*. <http://www.boost.org/>, . – zugegriffen am 30. Oktober 2008
- [69] *ISO/IEC 14977 Information technology - Syntactic metalanguage - Extended BNF*. [http://standards.iso.org/ittf/PubliclyAvailableStandards/s026153_ISO_IEC_14977_1996\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/s026153_ISO_IEC_14977_1996(E).zip), . – zugegriffen am 30. Oktober 2008
- [70] TRIPOS L.P., St. Louis, MO, USA: *Triplos Mol2 File Format*. http://www.tripos.com/tripos_resources/fileroot/mol2_format_Dec07.pdf, . – zugegriffen am 30. Oktober 2008
- [71] CHEMICAL COMPUTING GROUP INC., Montreal, Quebec, Canada: *MOE 2007.09*
- [72] THE CAMBRIDGE CRYSTALLOGRAPHIC DATA CENTRE, Cambridge, UK: *GOLD 3.2*
- [73] SHOEMAKE, Ken: *Quaternions*. <http://www.sfu.ca/~jwa3/cmpt461/files/quatut.pdf>, . – zugegriffen am 30. Oktober 2008
- [74] *Kapitel SHOEMAKE, Ken : UNIFORM RANDOM ROTATIONS*. In: KIRK, D. (Hrsg.): *Graphics Gems III*. New York : Academic Press, 1992

- [75] HESS, B ; KUTZNER, C ; SPOEL, D van d. ; LINDAHL, E: GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. In: *J. Chem. Theory Comput.* 4 (2008), Nr. 3, S. 435–447
- [76] MATSUMOTO, Makoto ; NISHIMURA, Takuji: Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. In: *ACM Trans. Model. Comput. Simul.* 8 (1998), Nr. 1, S. 3–30
- [77] NAMASIVAYAM, Vigneshwaran ; GUENTHER, Robert: PSO@AUTODOCK: A fast flexible molecular docking program based on swarm intelligence. In: *Chem. Biol. Drug Des.* 70 (2007), Nr. 6, S. 475–484
- [78] SHOEMAKE, Ken: Animating rotation with quaternion curves. In: *Proceedings of the 12th annual conference on Computer graphics and interactive techniques* (1985), S. 245 – 254
- [79] VERDONK, ML ; CHESSARI, G ; COLE, JC ; HARTSHORN, MJ ; MURRAY, CW ; NISSINK, JWM ; TAYLOR, RD ; TAYLOR, R: Modeling Water Molecules in Protein-Ligand Docking Using GOLD. In: *J. Med. Chem.* 48 (2005), Nr. 20, S. 6504–6515
- [80] ROBERTS, BC ; MANCERA, RL: Ligand-Protein Docking with Water Molecules. In: *J. Chem. Inf. Model.* 48 (2008), Nr. 2, S. 397–408
- [81] GRAAF, C de ; POSPISIL, P ; POS, W ; FOLKERS, G ; VERMEULEN, NPE: Binding Mode Prediction of Cytochrome P450 and Thymidine Kinase Protein-Ligand Complexes by Consideration of Water and Rescoring in Automated Docking. In: *J. Med. Chem.* 48 (2005), Nr. 7, S. 2308–2318
- [82] THE CAMBRIDGE CRYSTALLOGRAPHIC DATA CENTRE, Cambridge, UK: *GOLD 4.0*
- [83] MOITESSIER, N ; ENGLEBIENNE, P ; LEE, D ; LAWANDI, J ; CORBEIL, CR: Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. In: *Br. J. Pharmacol.* 153 (2007), S. 7–26
- [84] KELLENBERGER, E ; RODRIGO, J ; MULLER, P ; ROGNAN, D: Comparative evaluation of eight docking tools for docking and virtual screening accuracy. In: *Proteins* 57 (2004), Nr. 2, S. 225–242
- [85] PIPPEL, M: *Development and Validation of a Statistical Scoring Function for a Protein-Ligand Docking Program*, Martin-Luther-Universität Halle-Wittenberg, Diplomarbeit, 2008

- [86] HALGREN, TA: Merck Molecular Force Field. I. Basis, Form, Scope, Parameterization, and Performance of MMFF94. In: *J. Comput. Chem.* 17 (1996), Nr. 5, S. 490–519

Publikationen

HEINKE, R ; SPANNHOFF, A ; MEIER, R ; TROJER, P ; BAUER, I ; JUNG, M ; SIPPL, W: Virtual Screening and Biological Characterization of Novel Histone Arginine Methyltransferase PRMT1 Inhibitors. In: *ChemMedChem* 4 (2009), Nr. 1, S. 69 - 77

MEIER, R ; BALDAUF, C ; MERKLE, D: A Modular Framework for the Evaluation of Population-Based Algorithms for Molecular Docking. In: *Proceedings of META 2008 International Conference on Metaheuristics and Nature Inspired Computing* 29.-31.10.2008, Hammamet, Tunisia; Special session on 'Metaheuristics and Structural Biology'

UCIECHOWSKA, U ; SCHEMIES, J ; NEUGEBAUER, RC ; HUDA, EM ; SCHMITT, ML ; MEIER, R ; VERDIN, E ; JUNG, M ; SIPPL, W: Thiobarbiturates as sirtuin inhibitors: virtual screening, free-energy calculations, and biological testing. In: *ChemMedChem* 3 (2008), Nr. 12, S. 1965 - 1976

NEUGEBAUER, RC ; UCIECHOWSKA, U ; MEIER, R ; HRUBY, H ; VALKOV, V ; VERDIN, E ; SIPPL, W ; JUNG, M: Structure-activity studies on splitomicin derivatives as sirtuin inhibitors and computational prediction of binding mode. In: *J. Med. Chem.* 51 (2008), Nr. 5, S. 1203 - 1213

SCHLEGEL, B ; LAGGNER, C ; MEIER, R ; LANGER, T ; SCHNELL, D ; SEIFERT, R ; STARK, H ; HÖLTJE, HD ; SIPPL, W: Generation of a homology model of the human histamine H(3) receptor for ligand docking and pharmacophore-based screening. In: *J. Comput. Aided Mol. Des.* 21 (2007), Nr. 8, S. 437 - 453

TRAPP, J ; MEIER, R ; HONGWISSET, D ; KASSACK, MU ; SIPPL, W ; JUNG, M: Structure-activity studies on suramin analogues as inhibitors of NAD⁺-dependent histone deacetylases (sirtuins). In: *ChemMedChem* 2 (2007), Nr. 10, S. 1419 - 1431

TRAPP, J ; JOCHUM, A ; MEIER, R ; SAUNDERS, L ; MARSHALL, B ; KUNICK, C ; VERDIN, E ; GOEKJIAN, P ; SIPPL, W ; JUNG, M: Adenosine mimetics as inhibitors of NAD⁺-dependent histone deacetylases, from kinase to sirtuin inhibition. In: *J. Med. Chem.* 49 (2006), Nr. 25, S. 7307 - 7316

FROHBERG, P ; WAGNER, C ; MEIER, R ; SIPPL, W: Derivatives of arylhydrazonic acids. Part 3: Stereochemical rearrangement of Z-oxanilo-N1-dialkyl-N2-arylamidra-zones. In: *Tetrahedron* 62 (2006), S. 6050 - 6060

Posterbeiträge

MEIER, R ; SIPPL, W: A new approach for flexible protein-ligand docking based on particle swarm optimisation. *3. German Conference on Chemoinformatics*, Goslar, 11.-13.11.2007

HEINKE, R ; MEIER, R ; SPANNHOFF, A ; BAUER, I ; GUST, R ; BROSCHE, G ; JUNG, M ; SIPPL, W: Virtual Screening of Novel Histone Arginine Methyltransferase PRMT1 Inhibitors. *Jahrestagung der Deutschen Pharmazeutischen Gesellschaft*, Erlangen, 10.-13.10.2007

MEIER, R ; SIPPL, W: A new approach for flexible protein-ligand docking based on particle swarm optimisation. *21. Darmstädter Modelling Workshop*, Erlangen, 16.-17.05.2007

SIPPL, W ; HEINKE, R ; MEIER, R ; SPANNHOFF, A ; BAUER, I ; GUST, R ; BROSCHE, G ; JUNG, M: Virtual and Biological Screening of Novel Histone Arginine Methyltransferase PRMT1 Inhibitors. *Annual Meeting "Frontiers in Medicinal Chemistry" of the GDCh division of Medicinal Chemistry and the DPhG Division of Pharmaceutical/Medicinal Chemistry*, Berlin, 22.-24.03.2007

TRAPP, J ; JOCHUM, A ; MEIER, R ; SAUNDERS, L ; MARSHALL, B ; KUNICK, C ; VERDIN, E ; GOEKJIAN, P ; SIPPL, W ; JUNG, M: Adenosine mimetics as inhibitors of NAD⁺-dependent histone deacetylases - from kinase to sirtuin inhibition. *ChemBioNet Dechema*, Frankfurt, 12.12.2006

MEIER, R ; SIPPL, W: Hierarchical Virtual Screening of Regulators of G-Protein Signalling(RGS)-Inhibitors. *20. Darmstädter Modelling Workshop*, Erlangen, 23. - 24.05.2006

GUPTA, MK ; PRABHAKAR, YS ; MEIER, R ; SIPPL, W: *Aminopyridazine Derivatives as Acetylcholinesterase Inhibitors - An Assessment with Multi-Model QSAR studies ACS-CSIR Conference*, Pune, India, 07.-09.01.2006

MEIER, R ; SIPPL, W: Hierarchical Virtual Screening of Regulators of G-Protein Signalling(RGS)-Inhibitors. *CCG User Group Meeting 2005*, Köln, 19.-20.09.2005

MEIER, R ; SIPPL, W: The impact of scoring functions on the results of molecular docking studies. *19th Molecular Modelling Workshop*, Erlangen, 03.-04.05.2005

Vorträge

MEIER, R ; SIPPL, W: PARADOCKS - An Extensible Framework for Parallel Molecular Docking. *22nd Molecular Modeling Workshop*, Erlangen, 30.4.2007

MEIER, R ; SIPPL, W: Using a Modified XScore as Fitness Function for PARADOCKS. *Molecular Docking, Complexity, and Optimization, An ECCS 2007 Satellite Conference*, Dresden, 4.10.2007

Lebenslauf

Persönliche Angaben:

Name: René Meier
Anschrift: Bernhardstr. 24
06110 Halle/Saale
E-Mail: meier.rene@googlemail.com
Geburtsdatum: 15.11.1977
Geburtsort: Meißen
Familienstand: ledig
Staatsangehörigkeit: deutsch

Bildungsgang:

09/1984 – 06/1992 „Thomas Müntzer“ POS Miltitz (Sachsen)
09/1992 – 06/1996 Franziskanerum Meißen (Sachsen)
1996 allgemeine Hochschulreife
10/1997 – 02/2003 Studium der Biochemie an der Universität Leipzig
2003 Abschluss als Diplom-Biochemiker
09/2003 – 02/2009 Erarbeitung der vorliegenden Dissertation unter
der Leitung von Prof. Wolfgang Sippl an der
Naturwissenschaftlichen Fakultät I der Martin-
Luther-Universität Halle-Wittenberg

Wehrdienst:

07/1996 – 03/1997 Grundwehrdienst in Leipzig

Nebentätigkeiten:

04/1997 – 09/1997 Vermessungshelfer bei Rohrnetzbau GmbH, Cos-
wig
02/1999 – 05/2001 wiss. Hilfskraft bei Cellular Products, Leipzig
07/2001 – 09/2001 wiss. Hilfskraft am MPI für evolutionäre Anthro-
pologie, Leipzig

Halle(Saale), 18.05.2009

René Meier

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst, keine anderen als die angegebenen Hilfsmittel verwendet und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Halle(Saale), 18.05.2009

Rene Meier