MODELING PROTEIN INTERACTIONS IN PROTEIN BINDING SITES AND OLIGOMERIC PROTEIN COMPLEXES

Dissertation zur Erlangung des Grades
"Doktor der Naturwissenschaften"
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes

vorgelegt von

Matthias Michael Dietzen, M.Sc.

Saarbrücken
Juli, 2014

Tag des Kolloquiums	20. November 2014
Dekan der Naturwissenschaftlich-Technischen Fakultät I	Univ.- Prof. Dr. Markus Bläser
Vorsitzender des Prüfungsausschusses	Prof. Dr. Hans-Peter Lenhof
Erstgutachter / Doktorvater	Prof. Dr. Dr. Thomas Lengauer
Zweitgutachter	Prof. Dr. Andreas Hildebrandt
Akademischer Beisitzer	Dr. Olga Kalinina

ABSTRACT

Three-dimensional structures of protein-ligand and protein-protein complexes can provide key insights into biochemical processes within living cells, yet, their experimental determination is often expensive, time-consuming, or can fail due to the heterogeneity in the complex composition and thus the binding affinities of different components. A computational prediction of these structures can overcome these problems in certain cases and is thus highly demanded in many areas of research.

In this work, we address two questions: first, can one predict conformational changes of the protein backbone upon ligand binding, using the energetically most favorable motions obtained from normal mode analysis of elastic network models, and second, can one computationally assemble large protein complexes, using the structures and stoichiometries of their monomers and the approximate interaction geometries.

For the first problem, using a diverse set of 433 pairs of bound and unbound protein conformations, we could show that the benefit from such motions is small: modeling ligand-induced conformational changes using normal modes is rather ineffective. To solve the second problem, we have developed a novel scoring function and an efficient algorithm for iterative complex assembly based on pairwise dockings, 3D-MOSAIC, that, on a diverse benchmark set of 308 complexes, can accurately and efficiently assemble protein complexes of up to 60 monomers and 15 protein types.

ZUSAMMENFASSUNG

Dreidimensionale Strukturen von Protein-Ligand- und Protein-Protein-Komplexen können wichtige Einblicke in biochemische Prozesse in Zellen bieten, doch ihre Bestimmung ist oft teuer, aufwändig, oder misslingt wegen der Heterogenität der Komplexzusammensetzung und der Bindungsstärken der verschiedenen Komponenten. Die computergestützte Vorhersage solcher Komplexe kann diese Probleme in bestimmten Fällen überwinden und ist daher in vielen Forschungsbereichen hoch gefragt.

In dieser Arbeit gehen wir folgende Fragestellungen an: Erstens, kann man mittels der durch Normalmoden-Analyse eines Elastic-Network-Models vorhergesagten energetisch günstigsten Bewegungen des Proteinrückgrats die Konformationsänderungen bei Ligandaufnahme modellieren, und zweitens, kann man große Proteinkomplexe computergestützt assemblieren, wenn nur Struktur und Stöchiometrie der Monomere, sowie die groben Interaktionsgeometrien bekannt sind?

Zur ersten Frage konnten wir auf einer diversen Menge von 433 Paaren gebundener und ungebundener Proteinstrukturen zeigen, dass der Nutzen solcher Normalmoden klein ist: die Modellierung ligand-induzierter Konformationsänderungen beim Docking mittels solcher Bewegungen erweist sich als ineffektiv. Zur Lösung des zweiten Problems entwickelten wir eine neue Bewertungsfunktion und einen effizienten Algorithmus, 3D-MOSAIC, der auf einer diversen Testmenge von 308 Komplexen akkurat und effizient Proteinkomplexe von bis zu 60 Monomeren und 15 Proteintypen assemblieren kann.

PUBLICATIONS

Identification of CYP106A2 as a Regioselective Allylic Bacterial Diterpene Hydroxylase. Bleif, S., Hannemann, F., Lisurek, M., von Kries, J. P., Zapp, J., Dietzen, M., Antes, I., and Bernhardt, R. *ChemBioChem*, 12(4):576–582, 2011.

On the Applicability of Elastic Network Normal Modes in Small-Molecule Docking. Dietzen, M., Zotenko, E., Hildebrandt, A., and Lengauer, T. *Journal of Chemical Information and Modeling*, 52(3):844–856, 2012.

Efficient Computation of Root Mean Square Deviations Under Rigid Transformations. Hildebrandt, A. K., Dietzen, M., Lengauer, T., Lenhof, H.-P., Althaus, E., and Hildebrandt, A. *Journal of Computational Chemistry*, 35(10):765–771, 2014.

Oligomeric Complex Structures can be Computationally Assembled Using Efficient Combination of Docked Interfaces. Dietzen, M., Kalinina, O., Taškova, K., Kneissl, B., Hildebrandt, A. K., Jaenicke, E., Decker, H., Lengauer, T., and Hildebrandt, A. In preparation, 2014.

AWARDS

RCSB PDB Poster Prize. Three-Dimensional Modeling of Macromolecular Assemblies by Efficient Combination of Pairwise Dockings. Dietzen, M., Kalinina, O., Taškova, K., Kneissl, B., Jaenicke, E., Decker, H., Lengauer, T., and Hildebrandt, A. *ISMB/ECCB 2013*.

DANKSAGUNG

Ich möchte mich and dieser Stelle von ganzem Herzen bei all jenen Menschen bedanken, die mich auf vielfältige Art und Weise in der Durchführung und Fertigstellung dieser Arbeit unterstützt haben, insbesondere bei allen, die durch ihre Betreuung zum Gelingen dieser Arbeit wesentlich beigetragen haben, aber auch bei meinen Kollegen und Freunden, und nicht zuletzt meiner gesamten Familie, deren Rückhalt und Unterstützung einen unschätzbaren Beitrag zu dieser Arbeit geleistet haben.

Mein Dank gilt an dieser Stelle zuallerst meinem Doktorvater, Herrn Prof. Dr. Dr. Thomas Lengauer, der mir die Chance gegeben hat, mir unter seiner hervorragenden Führung neue Bereiche der Bioinformatik zu erschließen und mich darin zu etablieren. Vergessen möchte ich hierbei auch insbesondere nicht die Zeit, in der gute Ergebnisse rar waren, seine Unterstützung aber weiterhin Bestand hatte. Ohne diese Unterstützung wäre diese Arbeit nicht möglich gewesen.

Danken möchte ich insbesondere auch Prof. Dr. Andreas Hildebrandt, der sich meiner Betreuung in der entscheidenden Phase meiner Promotion angenommen hat und mir jederzeit mit Rat und Tat zur Seite stand. Seine Ideen und seine Anregungen halfen mir stets, meine Forschung aus neuen Perspektiven zu betrachten und dabei nicht das Wesentliche aus dem Auge zu verlieren.

Mein Dank geht auch an Dr. Olga Kalinina, deren Fachkenntnis und Enthusiasmus mir mehr als einmal eine große Hilfe und Motivation waren, und deren nüchterne Betrachtungsweise mich mindestens einmal davor bewahrt hat, die Nerven zu verlieren.

Des Weiteren möchte ich Dr. Elena Zotenko für die die fruchtbaren Diskussionen und die Unterstützung im Bereich der Normalmoden-Analyse, sowie Prof. Dr. Iris Antes für die Betreuung und Einführung während des ersten Jahrs meiner Promotion danken.

Selbstverständlich gilt mein Dank auch all meinen Kollegen der Arbeitsgruppe Computational Biology and Applied Algorithmics, mit denen ich viele fruchtbare Diskussion geführt, denen ich so manche Erkenntnis zu verdanken, und mit denen ich auch neben der Promotion Zeit verbracht habe: allen voran Dr. Bastian Beggel, Peter Ebert, Dr. Christoph Hartmann, Fabian Müller, Dr. Nico Pfeifer und Alejandro Pironti, Dr. Marcel Schulz, außerdem Dr. Joachim Büch, Georg Friedrich und Ruth Schneppen-Christmann, die durch ihre Arbeit im Hintergrund eine hervorragende Umgebung für effizientes wissenschaftliches Arbeiten geschaffen haben.

Weiterhin möchte ich mich bei Prof. Dr. Elmar Jaenicke und Prof. Dr. Heinz Decker vom Institut für Molekulare Biophysik der Johannes-Gutenberg-Universität Mainz für die Projektidee zur Assemblierung von Proteinkomplexen und ihre großartigen Beiträge zum Verständis der zu Grunde liegenden Mechanismen bedanken. Die Chance, in diesem Forschungsgebiet arbeiten zu können, betrachte ich als außergewöhnlichen Glücksfall.

Auch Prof. Dr. Hans-Peter Lenhof sowie den aktuellen und ehemaligen Mitgliedern seiner Arbeitsgruppe, der ich seit meinem Studium eng verbunden bin, möchte ich meinen Dank aussprechen, sowohl für die fruchtbaren Diskussionen als auch für die gelegentlichen gemeinsamen freizeitlichen Aktivitäten: Dr. Anna-Katharina Hilde-

brandt, Daniel Stöckel, Lara Schneider, Dr. Christina Backes, Dr. Marc Hellmuth, Dr. Oliver Müller und Prof. Dr. Andreas Keller.

Ebenso möchte ich mich bei der Arbeitsgruppe “Software Engineering and Bioinformatics” von Prof. Dr. Andreas Hildebrandt bedanken, insbesondere Dr. Katerina Taškova und Dr. Benny Kneissl, mit dem mich seit den Anfängen unseres Studiums eine enge Freundschaft verbindet. Auch Sabine Müller und Dr. Lars Steinbrück möchte ich für ihre Freundschaft und Unterstützung danken.

Außergewöhnlicher Dank gilt auch meiner gesamten Familie: meinen Eltern Marianne und Günter Dietzen, meinen Geschwistern Stephan, Philipp, Thomas und Corinna, ihren Partnern, und meinen Großeltern Irene und Ferdinand Karos, auf deren mentale Unterstützung ich jederzeit bauen konnte. Ebenso möchte ich mich bei allen jenen Menschen bedanken, die mir auf diesem Weg eine besonders große Unterstützung waren: Désirée Schäfer, Michael Ewig, Claudia Kalka, meiner Theatergruppe Bohemian Company, sowie dem Team von Vocallessons. Sie haben immensen Anteil daran, dass ich nicht den Boden unter den Füßen verloren habe und mir meine Kreativität bewahren konnte.

Auch allen Menschen, die vor oder während der Entstehung dieser Arbeit Teil meines Lebens waren, mir mit Rat und Tat oder wohlmeinenden Worten zur Seite standen, aber hier namentlich unerwähnt bleiben, möchte ich ganz herzlich danken: Ihr habt im Großen und Kleinen Anteil an dieser Arbeit und das werde ich immer zu schätzen wissen. Danke.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Part I

FOUNDATIONS

1 INTRODUCTION

Computational Biology and hence the work presented in this thesis stands in the long tradition of the evolution of medicine as a science over the course of the last millennia as well as modern, more specialized areas of science. The latter comprise the natural sciences biology, physics, and chemistry, together with the comparatively new field of computer science, and provide the methodological background on which this thesis is based.

In this introductory chapter, we will first present selected discoveries and research from the above mentioned scientific areas, many of them related to protein structure determination, prediction, modeling and drug design. The presented experimental methods, theoretical concepts, algorithms and experimental data form the breeding ground on which the research presented in this work could dwell.

In the subsequent section, we will introduce the reader to the two main problems tackled in this thesis: the application of elastic network normal modes in protein-small molecule docking and the assembly of oligomeric protein complexes.

Finally, we will conclude the chapter with a structural overview of the main part of this thesis.

1.1 DISCOVERIES AND RESEARCH RELEVANT FOR PROTEIN STRUCTURE MODELING

The field of modern molecular structure determination is based on an observation Wilhelm Konrad Röntgen made in 1895 when he began to systematically study a certain kind of electromagnetic radiation which he became aware of during his experiments. This radiation which he named X-rays [1] would later form the basis for the field of X-ray crystallography, to date the state-of-the-art method to determine the structures of proteins and other molecules from X-ray diffraction patterns of crystals of these molecules. Such patterns were first observed by Max von Laue in 1913 [2] and are explained by Bragg's Law, proposed in 1912 by William Henry and William Lawrence Bragg [3]. In 1929, Linus Pauling established five rules to describe the principles that govern the structure of complex ionic crystals [4] and should facilitate the process of crystal structure determination.

A key event of macromolecular structure determination took place in 1951 when Linus Pauling first determined a protein secondary structure using X-ray crystallography [5]. Five years earlier, another method relevant for structure determination, called nuclear magnetic resonance (NMR), was developed by Edward Mills Purcel and Felix Bloch [6, 7]. In 1953, James Watson and Francis Crick developed a model suggesting a double-helix structure for DNA [8], supported by experimental data obtained by Rosalind Franklin. The first complete monomeric protein structure, sperm whale myoglobin, was determined in 1958 by John Kendrew [9] using X-ray crystallography. Its tetrameric pendant, hemoglobin, was resolved in 1968 by Max Perutz [10] and represents the first structurally determined protein complex. 1968 was also the year that BRAD (Brookhaven RAster Display) [11], the first program to visualize

protein structures in 3D, became available. Three years later, the Protein Data Bank (PDB), the major source for protein structural data, was born [12].

Parallel to the progress in protein structure determination, advances were also made in the fields of structure prediction, modeling and drug design.

First experimental observations related to that area date back to 1827, when Robert Brown observed particles moving through water in a seemingly random fashion [13]. The underlying mechanism, which many to-date drug absorption simulations try to model, would later be explained by Albert Einstein and become known as Brownian Motion (1905) [14]. In 1840, another fundamental theory relevant for such simulations and docking methods was proposed by Germain Hess, stating that the energy difference between two states of a chemical reaction is solely dependent on the the reaction partners and not on the reaction pathway [15].

In 1894, Emil Fischer was the first to compare the mechanism by which a substrate is bound by a biochemical macromolecule to a key fitting its lock [16]. In 1958, David Koshland extended the lock-and-key principle by Emil Fisher to the so-called induced-fit theory [17]. It postulates that any catalytic activity upon substrate binding requires an exact orientation of the catalytic groups in the active site, and that this orientation can be reached through conformational changes induced by the binding process. One of the first approaches able to model these conformational changes was developed in 1959 by Alder and Wainwright: the method of molecular dynamics (MD) simulations [18], modeling the interactions and movements between a set of atoms over time.

In the same year that Crick and Watson presented their DNA model (1953), Nicolas Metropolis introduced the Monte Carlo Method [19], a stochastic sampling algorithm that employs Boltzmann's constant and was developed in the late 1940's at the Los Alamos National Laboratory. The Boltzmann constant (and the field of statistical mechanics which greatly affected the field of thermodynamics) has its origins in 1877 when Ludwig Boltzmann derived statistical descriptions of entropy, a measure for the disorder of a thermodynamic system [20], and was introduced by Max Planck in 1901 [21].

An extension of the Monte Carlo Method, the so-called simulated annealing (first developed by Kirkpatrick, Gelatt and Vecchi in 1983) [22], represents one of the state-of-the-art methods for medium-scale sampling of energy landscapes of biological macromolecules and global optimization of protein-ligand interactions. The genetic algorithm, a related method emulating evolutionary processes by selecting only the fittest members of a population in each generation, was introduced in the mid 1970's by John Holland [23, 24].

Similarly, continuous research in the field of potential energy functions to simulate small molecules led to the development of CHARMM [25], a program for energy and dynamics calculations as well as the minimization of macromolecular systems. CHARMM was developed by Martin Karplus and featured significant contributions from Karplus' fellow 2013's Nobel Prize winners Arieh Warshal and Michael Levitt, a pioneer on the field of protein energy calculations [26].

Many of the first steps in the field of bioinformatics primarily involved the development of methods that investigate the evolutionary relationships between genetic sequences as well as their similarity, such as Fitch's phylogeny algorithm in 1967 [27], the Needleman-Wunsch algorithm for pairwise global alignment of molecular sequences in 1970 [28], and the Smith-Waterman algorithm for local sequence alignment developed in 1981 [29].

The 1990's, following the pioneering DOCK algorithm (1982) [30], became a decade of structural bioinformatics and witnessed the emergence of a wealth of docking algorithms trying to computationally predict the interactions between proteins and their binding partners. While the first methods considered proteins as rigid bodies, the rapidly increasing computational power soon led to the first algorithms incorporating ligand flexibility. A ground-breaking contribution to the field was made in 1996, when FlexX, the first program that was able to quickly screen through libraries of thousands of flexible ligands, became available [31]. Nowadays, there exists a wealth of docking algorithms, both for modeling protein-small molecule as well as protein-protein complexes. Because the problem of molecular docking is an integral part of this thesis, the underlying principles and the algorithms used throughout this work are described in detail in Section 3.3; Chapter 4 shortly discusses further prominent protein-small molecule docking algorithms and related techniques.

1.2 SPECIFIC OBJECTIVES OF THIS THESIS

Nowadays, a rapidly increasing amount of data on structures of proteins and other macromolecules becomes available. In addition, the technological progress of methods to determine, manipulate and measure the biochemical properties of proteins has led to a manifold of algorithms and databases from which information on these structures and their interactions can be obtained. This vast amount of structural and interaction data allows for a thorough investigation of the complex interplay of these structures in all its facets from atomic to cellular levels. Consequently, bioinformatics methods have become increasingly indispensable not only when it comes to processing, filtering, and analyzing, but also to integrating and interpreting this information.

Insights into cellular process can be gained, for example, through determination of the structure and biochemical properties of proteins and oligomeric assemblies. These properties are the main determinants of a protein's function (or that of an assembly), hence, investigating such structures to understand the resulting function (or dysfunction) and the interplay with other biochemical compounds or macromolecules represents one of the key approaches to detect, treat and possibly cure diseases. The data used for such a study can for example comprise high- and low-resolution structural data, e.g., X-ray structures and electron density maps, protein dynamics data from MD simulations or conformational ensembles from NMR studies, as well as interaction data obtained from experiments or predicted computationally, e.g. by docking methods, correlated mutations or homology modeling.

In this thesis, we investigate two different areas related to the prediction of protein structure and interaction research.

As stated in the previous section, computational power now allows for the incorporation of structural flexibility into docking methods. However, while the treatment of flexible ligands can be considered feasible with state-of-the-art computers, modeling protein flexibility still represents a major challenge. In particular, a fast prediction of protein backbone flexibility upon ligand binding, as required for high-throughput drug screening, is hardly doable. In this context, we investigate the applicability of normal modes, a technique to efficiently model the most dominant movements of a protein based on a coupled harmonic potential, a so-called elastic network, using a coarse-grained protein representation.

The second field of research addresses the modeling of macromolecular protein complexes consisting of a large number of protein monomers. Their structural determination is often difficult or even impossible. The aim of this work is thus to assemble these complexes from sub-complexes of pairs of monomers obtained from docking. While the prediction of interactions and binding modes from pairwise dockings is nowadays a standard technique, the extension of this methodology to multi-component systems still presents a major problem.

For example, the formation of a homomeric complex with m monomers requires that at least $m-1$ connections between the participating monomers must be established. If we can choose between d docking poses to establish each of these $m-1$ connections, an exhaustive search leads to a theoretically possible number of d^{m-1} unique combinations of docking poses from which a complex can be obtained. We thus observe an exponential increase in the number of solutions for increasing m. Such problems are known as combinatorial problems and rapidly lead to a large space of theoretically possible solutions, a phenomenon that is commonly known as combinatorial explosion and, if treated naïvely or exhaustively, becomes intractable even for small values of m (and d to a much smaller extent).

In this thesis, we present an iterative tree-based greedy algorithm, called 3D-MOSAIC, which only requires a high-resolution representative structure for each kind of monomer present in the complex as well as pairwise dockings sampling the assumed native interactions. To this end, it uses a novel scoring function that we developed and which can rapidly discriminate between reasonable and unreasonable solutions, and can thus be used to effectively prune the combinatorial space that has to be considered.

1.3 OVERVIEW

This thesis is divided into five parts. The first part subsumes the state-of-the-art knowledge providing the basis for this thesis and comprises two additional chapters besides the above introduction. Chapter 2 presents the structural hierarchy of proteins, their synthesis as well as their functions and the molecular machineries of which they can be part. Chapter 3 introduces the methods which are used throughout or are related to this work.

The second part is based on our publication on the applicability of elastic network model normal modes in small-molecule docking and is divided into chapters with an introduction into the field, a description of normal-mode related methods, a discussion of the results and conclusions drawn from this study.

The third part addresses the three-dimensional modeling of macromolecular oligomeric assemblies and is organized as follows: after a short introduction follows a chapter presenting all the notations and prerequisites required for the development of an algorithm to assemble such complexes from pairwise dockings. The subsequent chapter treats the development of the transformation match score. This score represents the core component or our algorithm 3D-MOSAIC which is presented in the next chapter. The experimental design and the evaluation of our algorithm are presented in the two subsequent chapters. Finally, a concluding chapter summarizes the work presented in this part.

The fourth part contains an outlook on future projects and improvements of 3D-MOSAIC, followed by the fifth part (the appendix) which contains supplementary information on parts two and three.

2

THE STRUCTURE OF PROTEINS AND THEIR ROLE IN BIOCHEMICAL PROCESSES

Together with DNA and RNA, proteins comprise the three kinds of macromolecules that mainly determine the biochemical behavior and functionality of every living cell (other macromolecules play important roles as well, e.g., lipids which are an integral part of cell membranes). Proteins perform a large variety of different tasks, e.g. energy acquisition, structural functions, message passing, catalytic functions, and replication (some of these functions are addressed in more detail in different contexts throughout this chapter) [32].

The reason why proteins are able to perform such diverse tasks lies in the structural flexibility of the protein main chain and the chemical diversity of amino acids, the basic building blocks proteins are composed of. The general structural properties of amino acids and proteins are explained in detail in Section 2.1. Section 2.2 describes the interactions within and between proteins and other compounds, and the mechanism by which proteins are synthesized in cells is shortly sketched in Section 2.3.

We will conclude this Chapter by presenting some of the tasks performed by proteins in more detail, first from a more functional (Section 2.4) and then from a structural perspective. In the latter part, we explicitly present some examples where these functions are carried out by macromolecular assemblies (Section 2.5), whose prediction is the subject of the second project presented in this thesis.

2.1 THE STRUCTURAL HIERARCHY OF PROTEIN BUILDING BLOCKS

The molecular function a protein is able to perform is to a large extent the result of its three-dimensional structure. In analogy to the large variety of different roles proteins can assume in an organism, their structural diversity is very high. To systematically investigate and classify proteins and their structures, one commonly distinguishes four different levels of structural hierarchy: the protein's primary, secondary, tertiary and quaternary structure.

In the following subsections, we will explain this hierarchy with its elements and principles in more detail before we present some examples of protein functions resulting from particular combinations of elements, especially the so-called tertiary and quaternary structure (Sect. 2.4 and 2.5). For more detailed information, the interested reader is referred to [32, 33] which, unless stated otherwise, provide the main sources for the following subsections.

2.1.1 *Primary Structure*

The bottom level of structural hierarchy is the primary structure, the amino-acid sequence. In general, a set of 20 different natural amino acids is considered to comprise the standard repertoire of building blocks for peptides and proteins. Each of these 20

amino acids is composed of two structural segments, one that is common to all amino acids and one that determines the individual chemical property of the amino acid, the so-called side chain.

The structural segment all 20 amino acids have in common consists of an amino group (NH_2), a carboxyl group (CO(OH)) and a hydrogen (Fig. 2.1) which are all covalently bound to a central carbon atom, the C_α atom. Through this element, amino acids are capable of interconnecting with other amino acids and thus forming chains thereof. The sequence of amino acids in such a chain defines the primary structure.

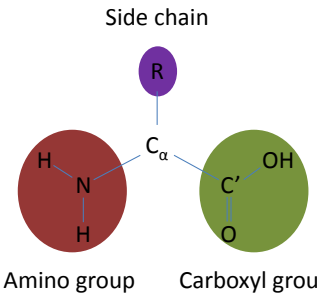


Figure 2.1: The general structure of amino acids, consisting of an amino and a carboxyl group, as well as a side chain, all attached to a central C_α atom. Inspired by [32].

These chains are formed by establishing a bond between the carboxyl carbon (C') of one amino acid and the nitrogen of the succeeding amino acid under release of a water molecule. This water molecule is the result of a process called hydrolysis which entails a chemical reaction of the hydroxyl group from the carboxyl unit and a hydrogen from the amino group. The bond connecting the two amino acids is called a peptide bond and hence, a sequence of connected amino acids is called a polypeptide chain (Fig. 2.2). All atoms involved in the peptide bonds of such a chain as well as the C_α atoms and attached hydrogens comprise the chain's backbone. The process of forming polypeptide chains is called protein synthesis and is sketched in Section 2.3. The vast majority of proteins is linear, containing a start and an end residue (N- and C-terminus), however circular proteins have also been observed [34].

The side chains are unique for all 20 amino acids and are bound via the remaining fourth valence of the C_α atom. Each type exhibits a specific combination of biochemical properties such as hydrophobicity, charge, polarity, and aromaticity. The number of theoretically possible amino acid sequences with a length corresponding to the average number of amino acids proteins in the human body consist of (around 300 amino acids) is 20^{300}, yet only about 100,000 different types of functional proteins are assumed to be present in human cells [35].

Amino acid sequences corresponding to natural proteins exhibit properties which are not characteristic for random amino acid sequences, rather, functional sequences have been selected through evolutionary pressure to efficiently serve specific purposes in the cell [35]. The respective combination of amino acids in such functional se-

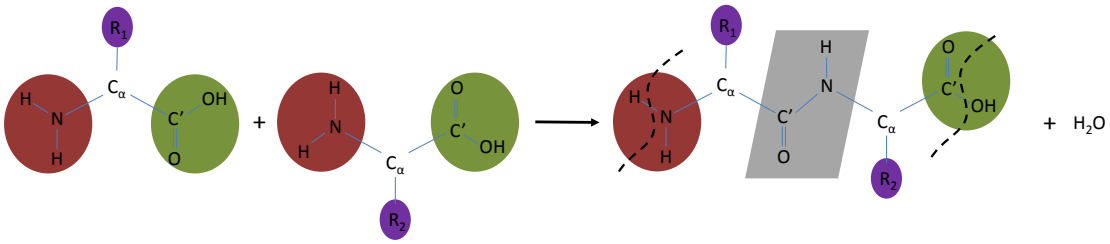


Figure 2.2: The formation of a rigid peptide bond (gray plane) between two amino acids under cleavage of a water molecule (hydrolysis). The N- and C-terminus (red and green) provide points of attachment for further amino acids, indicated by the dashed, curved lines. Inspired by [32].

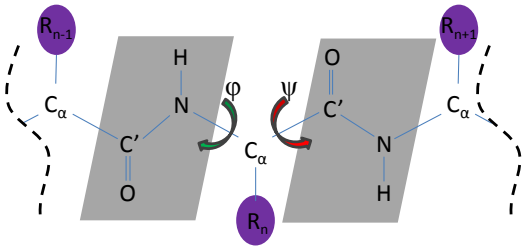
quences provides the biochemical environment for interactions with other molecules, and in many cases enables the protein to assume a three-dimensional structure to perform its function. However, there are also many functional proteins that have no stable three-dimensional structure, contain at least disordered regions [36, 37, 38] or function through transition between ordered and disordered states of the polypeptide chain or parts thereof [39]. The process of adopting a three-dimensional structure is called protein folding and is addressed in Section 2.3.1.1.

It is worth noting that, due to the fact that all amino acids except glycine have four different groups attached to the C_α atom, they can exist in two different chiralities which are called D- and L-form. In the L-form, the carboxyl, side chain, and amino groups are arranged in a clockwise manner around the hydrogen–Cα axis and in a counter-clockwise fashion in the D-form. However, evolutionary selection has led to a strong preference of L-amino acids and the D-form is almost irrelevant in natural proteins.

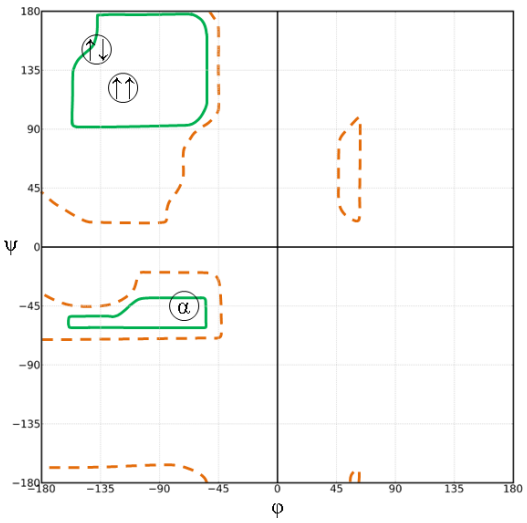
2.1.2 *Secondary Structure*

The protein secondary structure is the result of regular three-dimensional arrangements of segments of the primary structure. As explained above, each peptide bond involves the carbonyl group and the NH group between two consecutive C_α atoms; the formation of such a bond results in a mesomeric system where the electrons of the carbonyl group, the C′–N bond, and the nitrogen's free electron pair are delocalized over the corresponding atoms and covalent bonds (rather than being associated with a single one). This mesomeric system stabilizes the involved atoms in an almost rigid peptide plane, hence, the only remaining degrees of freedom in the protein backbone are the rotation angles φ and ψ around the respective nitrogen–C_α bond and the C_α–C′ bond of each amino acid (Fig. 2.3a).

These degrees of freedom are the determinants of the space of possible backbone conformations. Often, subsequent backbone segments show regular patterns in their backbone conformations, the most prominent ones being the α-helix, the β-sheet, and the rather irregular loop regions (Fig. 2.4). Fig. 2.3b shows a so-called Ramachandran plot



(a) φ and ψ torsional angles of the protein backbone. Inspired by [32].



(b) Exemplary Ramachandran plot showing the core and extended regions for different secondary stucture elements, most importantly the α-helix (α), the parallel (↑↑) and anti-parallel (↑↓) beta-sheets (image adapted from [40]).

Figure 2.3: Backbone torsional angles and Ramachandran plot.

which contains the distributions of φ and ψ for the different regular secondary structure elements.

The main characteristics of the α helix are its repetitive φ and ψ angles of -60° and -50° of a segment of consecutive amino acids, resulting in 3.6 amino acids on average per helix turn. This structure is stabilized by hydrogen bonds between the n-th C′=O and the $n+4$-th NH group in the amino acid sequence and shows a rise of about 1.5Å per residue. Depending on the sequence, the length of the helices can vary greatly between different structures. Other types of helices exist, but are much rarer.

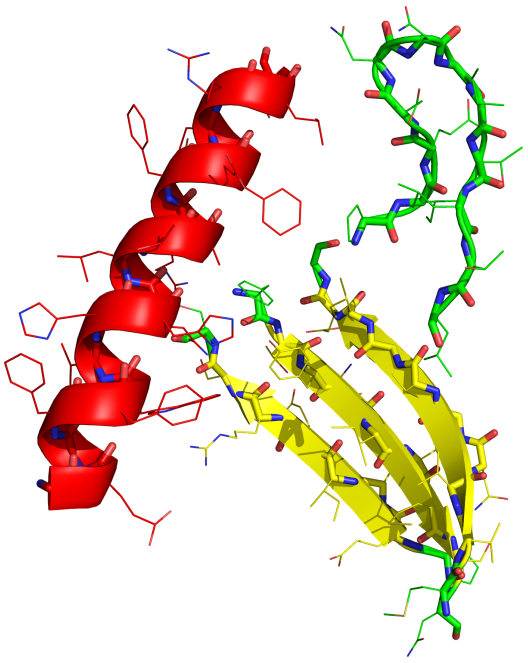


Figure 2.4: The most common secondary structure elements: α-helix (red), (anti-parallel) beta-sheet (yellow), and irregular loops (green) with backbone trace (sticks), side chains (lines), and illustration of the secondary structure (cartoon). Isolated from 1HCY.

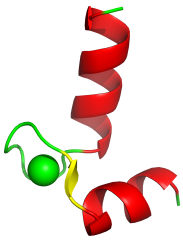
In contrast, β-sheets are not formed by one single segment of amino acid residues but rather require several stretches which are arranged in a parallel (same directionality) or anti-parallel (alternating directionality) way. Typically, β-sheets prefer to be exclusively one or the other although mixtures of both have been rarely observed.

The allowed φ- and ψ-angles cover a very wide range of -60° to -150° for φ and -90° to -175° for ψ. In both forms, all possible backbone hydrogen bonds are formed for the internal strands, but those in parallel sheets are evenly spaced while the ones in anti-parallel strands are alternatingly wide- and narrow-spaced.

To connect the aforementioned secondary structure elements and to allow for changes in spatial orientation of the polypeptide chain, a third, rather irregular element is required: the loop region. Loops are very flexible and exhibit no particular shape. In the case of soluble proteins, loops can often be found at the protein surface forming hydrogen bonds to the surrounding solvent while the more regular secondary structure elements form the hydrophobic core and often the center of activity of the protein.

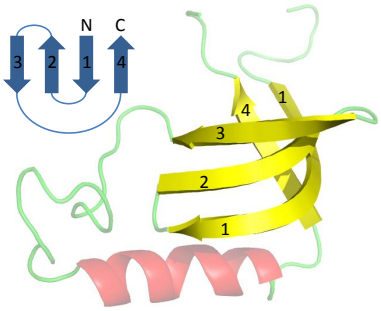
2.1.3 *Tertiary Structure*

The three-dimensional arrangement of combinations of secondary structure elements of a single polypeptide chain comprises the tertiary structure. Here, helices and sheets are tightly packed to form functional units, so-called domains. Commonly, domains are defined as contiguous segments of a protein or polypeptide chain that “fold into compact, local, semi-independent units” [41]. However, there also exist domains that do not consist of contiguous segments [42]. Depending on the purpose, slightly alternative definitions of domains are used, for example a more evolution-oriented definition as “an evolutionary unit observed in nature either in isolation or in more than one context in multidomain proteins” [43].

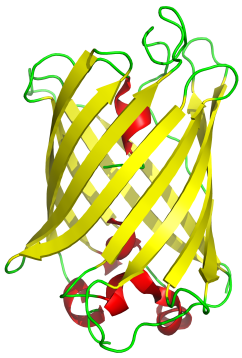


(a) EF-hand, isolated from parvalbumin (pdb code 1B8R).

(b) β-hairpin, isolated from bovine trypsin inhibitor (pdb code 4J2K).



(c) Greek key motif (non-transparent, numbered sheets), from staphylococcal nuclease (pdb code 2SNS) including a schematic drawing of the motif (upper left).



(d) β-barrel, isolated from green fluorescent protein (pdb code 1RRX).

Figure 2.5: Examples of motifs of secondary structure elements.

Domains are comprised of secondary structure elements [44] and simpler supersecondary structure motifs combining several few α-helices and/or β-sheets [32]. Such motifs can be distinguished by the number and frequency of the different secondary structure elements they contain as well as the special geometric topology they exhibit. The same motif is often present in a large variety of different proteins, and the combination of motifs determines the protein's specific functional properties.

One of the simplest and most prominent motifs observed in many proteins is the calcium-binding EF hand consisting of a helix, a loop turn and a second helix with the two helices arranged in an almost orthogonal fashion (similar to the outstretched thumb and forefinger of a hand, Fig. 2.5a) and the loop in between containing the calcium ion. Another common and very elementary motif is the β-hairpin (Fig. 2.5b) which directly connects two antiparallel strands of β-sheets and can be found in very many different proteins. Generally, the number of possibilities to connect the individual strands of β-sheets increases rapidly with the number of strands. For example, one frequently occurring way of connecting four anti-parallel β-strands is the Greek key motif (Fig. 2.5c), borrowing its name from the meandering patterns often found in Greek art, which is reminiscent of the way the sheets are connected by loops in the motif.

From these and many further simple motifs, more complex motifs such as β-barrels (Fig. 2.5d), domains and ultimately fully functional proteins are formed. The sec-

ondary and tertiary structure elements belonging to the same motif or domain are often found to be close in the primary structure, the protein's sequence. However, backbone segments forming parallel β-sheets are necessarily more distant in sequence, because here, the chain segment connecting two strands in the sheet must be longer to arrange the strands in an parallel fashion and can even contain other secondary structure elements. The process of forming a protein's secondary and tertiary structure is called protein folding and is described in Section 2.3.1.1.

While all individual secondary structure elements (including more exotic ones not described here) except loops are relatively rigid due to their strong hydrogen bonding patterns, the connections between the individual components of a motif are often not so strong and thus allow for a certain amount of flexibility. According to Koshland's induced fit theory [17], this flexibility represents the key mechanism of the molecular recognition of binding partners and hence the functions of the protein. Some examples of essential protein functions and pathways are described in Section 2.4.

2.1.4 *Quaternary Structure*

In addition to forming specific tertiary structures, several protein molecules often assemble to form a quaternary structure, or a protein complex. The mechanisms that govern the complex assembly process are the same as those resulting in the formation of the tertiary structure (see Section 2.2). But in the case of the quaternary structure, the interactions are established between different polypeptide chains rather than between secondary structure elements or domains within a single chain, as in the case of the tertiary structure. Yet, proteins present as individual components and assembling into a quaternary structure in one species may be covalently linked, often through further secondary structure elements such as unstructured loops, to form a multi-domain protein (tertiary structure) in other species.

A popular example of a protein complex is hemoglobin (e.g. pdb code: 2HHB, Fig. 2.6), the first structurally resolved complex. Hemoglobin is responsible for the oxygen transport mainly in vertebrates and consists of two α-globin and β-globin chains (sequence identity 40.268%).

Another example of a protein complex is the molecular chaperone GroEL (example pdb code: 1XCK) which, in presence of ATP and the GroES complex, can assist in protein folding (see Section 2.3.1.1) and recovery of denaturated proteins.

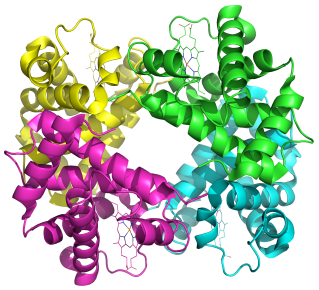


Figure 2.6: The heterotetrameric deoxyhemoglobin, consisting of two α-globin (green, purple) and β-globin chains (cyan, yellow) each.

The proteins involved in such an assembly often perform their function cooperatively, for example through allosteric modulation as in the case of the aforementioned hemoglobin [45], where the binding of an oxygen to one chain induces conformational changes in the complex that facilitate the uptake of oxygen in the other chains. Complexes can also be scaffolds for the ordered execution of subsequent chemical reactions, and are capable of feedback regulations which can, for example, adjust the rates of the individual chemical reactions to the rate-limiting step of the whole reaction cascade [46].

Protein complexes can be obligate, i.e., permanently assembled as in the case of hemoglobin, or transient, i.e., they assemble to perform their function and dissociate after completion as for example in the case of the GroEL-GroES complex. Of particular interest in the context of this thesis are the so-called oligomeric macromolecular assemblies which are discussed in Sect. 2.5.

2.2 PROTEIN INTERACTIONS AND INTERFACES

In the previous section, we have outlined the structural hierarchy of proteins. This structural hierarchy allows for a large variety of different proteins and thus functions. This functional diversity is to a large extent the result of the various three-dimensional structures proteins can adopt as well as their interactions with biochemical compounds and other macromolecules. Both, the assumption of a three-dimensional fold (see Section 2.3.1.1) as well as the establishment of interactions with other molecules is only possible if the resulting situation is energetically favorable w.r.t. to the environment (e.g. the cytosol or membrane).

The three-dimensional structure of folded proteins or macromolecular complexes results from internal interactions between the comprising amino acids, sometimes prosthetic groups (permanently bound biochemical compounds that contribute to the protein's function), and interactions with the environment. In many instances, interactions with other molecules in the environment take place at special sites, mostly at the surface of the protein, called binding sites, binding pockets, active sites or interfaces and are typically very specific w.r.t. the molecule to be bound.

The establishment of such interactions is due to two major factors: enthalpy and entropy. The enthalpy is an energetic term that reflects the internal energy of the system that largely depends on the non-covalent interactions between its components as well as solvent effects. These interactions are the result of forces that act on the interaction partners and are caused by their biochemical properties. Non-covalent interactions comprise van der Waals interactions, i.e. dispersion effects arising from induced dipoles in the binding partners as well as repulsion effects between the interaction partners for atoms whose distance is smaller than the sum of their van der Waals radii. Non-covalent interactions also include electrostatic interactions, i.e., interactions between charged and/or dipolic atoms or functional groups and lead to the formation of ionic, dipole/dipole and hydrogen bonds.

Solvent effects include the preference of hydrophobic or hydrophilic surface patches to be in contact with the same kind of surface patch, the latter often water-mediated. In the case of hydrophobic surfaces, this effect is enhanced by the fact that in solution – the native environment of most proteins – water molecules are removed from that area upon binding. This phenomenon is known as desolvation and describes the effect that water molecules located at hydrophobic surfaces, where they are less capable of forming hydrogen bonds, restore that ability when being surrounded by water molecules again. The water molecules released from the surface can form hydrogen bonds and thus increase enthalpy.

The entropic contribution arises from an effect described by the second law of thermodynamics. It determines that the creation of order (i.e., a reduction in the number of degrees of freedom) in a thermodynamic system requires a certain amount of energy. Such order is not only found when hydrophobic patches are in contact with water, rather, folded proteins in general represent highly ordered systems. In addi-

tion, the interaction with other molecules reduces the number of degrees of freedom in all binding partners, and thus also the entropy.

Consequently, a system can only be stable if the enthalpy arising from energetically favorable interactions outweighs the loss of energy due to entropic effects arising from the loss of disorder upon realizing the interaction. As stated by David Koshland's induced-fit theory [17], the binding process, and thus the establishment of energetically favorable interactions, can be supported by conformational changes of the interaction partners, to bring the complementary groups into close contact. The flexibility in the protein can hereby be restricted to a change of the rotameric state of the side chains, local backbone rearrangements or even large domain motions.

The strength of the binding is measured by the binding free energy ΔG_{bind}, the difference between the free energies of the molecules in the bound and unbound state:

$$\Delta G_{bind} = G_{bound} - G_{unbound} \tag{2.1}$$

Accurately estimating ΔG_{bind} for a given complex structure is the aim of most scoring functions employed in docking methods (see Section 3.3). Experimentally, ΔG_{bind} can be determined from the association and dissociation rates k_{on} and k_{off} of the involved molecules. The ratio of these rates is known as the inhibition constant k_i, whose logarithm is proportional to ΔG_{bind} [47]:

$$\Delta G_{bind} = -RT \ln \left(\frac{k_{on}}{k_{off}} \right) = -RT \ln k_i \tag{2.2}$$

Here, the scaling factor $-RT$ is a product of the temperature T at which the experiment is performed and the gas constant R. Obviously, for an interaction to take place, the bound state should be preferred over the unbound one, i.e., its free energy should be lower. Consequently, if $\Delta G_{bind} < 0$, an interaction can be considered favorable.

2.3 THE PROCESS OF PROTEIN BIO-SYNTHESIS

As already stated in Section 2.1.1, proteins are polypeptide chains, i.e., polymers composed of amino acids. In most organisms, the blueprint containing the information on the amino acid sequence of a protein is encoded in the DNA (though many viruses for example encode this information in form of RNA), in particular a gene. A gene is loosely defined as "a locatable region of genomic sequence, corresponding to a unit of inheritance, which is associated with regulatory regions, transcribed regions and/or other functional sequence regions" [48]. Genes coding for proteins are called (protein-)coding genes as opposed to non-coding (RNA) genes which are translated into functional RNA [49] rather than proteins. Because this thesis focuses on research related to proteins and their structure, we here sketch the process of protein biosynthesis from DNA.

Before the actual protein synthesis can take place, the encoding gene must first be transcribed to messenger RNA (mRNA). Gene transcription is heavily regulated to allow for an adaption of the cell to environmental conditions and to prevent the cell from damage and uncontrolled cell growth (see Subsection 2.4.3). During transcription, RNA nucleotides complementary to the DNA template strand are attached to the 3'-end of the growing RNA molecule by a protein called RNA polymerase [50, 51].

In a post-processing step, the non-coding parts of the mRNA molecule, called introns, are spliced and the exons encoding the protein structure are joined. The final mRNA molecule can then leave the nucleus towards the ribosome where the protein synthesis takes place.

In the context of this process [52, 53, 54], three different roles of RNA are of superior importance: the aforementioned mRNA as well as the transfer and the ribosomal RNA (tRNA [55] and rRNA respectively [56]). Whereas the mRNA is the messenger carrying the information from which the protein is to be built, the rRNA and tRNA are directly involved in the synthesis itself: in complex with a minimum of 50 other proteins, the rRNA forms the ribosome, the biological factory where new proteins are synthesized. rRNA contributes two subunits to the complex, the large subunit (LSU) and the small one (SSU), which, in total, make up the majority of the molecular weight of the whole ribosome. The protein synthesis can be divided into three steps, performed at the ribosome's three different binding sites (A,P,E): the (aminoacyl-)tRNA which links a sequence of three nucleotides (a so-called codon) to one of the 20 different amino acids, is bound to the ribosome at site A according to a complementary three-nucleotide sequence provided by the mRNA. Here, the aminoacyl-tRNA forms a peptide bond with the amino acid held by another (peptidyl-)tRNA located at site P and elongates the protein under synthesis by another elementary module. After formation, the tRNA at site A is moved to P (becoming the new peptidyl-tRNA), while that at P is moved to E, where it is dissected from the peptide chain and released. This process is iterated until the whole mRNA is processed and the polypeptide chain has been built.

2.3.1 *Protein Folding and Co- and Post-Translational Modifications*

Even before the translation has been completed, the already synthesized part of the polypeptide chain may undergo several post-translational modifications that enable the protein's functionality. For example, the polypeptide chain may correspond to several proteins instead of a single one. In that case, a cleavage at the corresponding peptide bonds in the chain is performed. Furthermore, the methionine at the N-terminus of the protein, corresponding to the start codon of the gene encoding for the polypeptide chain, is often removed. In addition, the insertion of membrane proteins into the membrane is performed, either co- or post-translationally [57, 58].

2.3.1.1 *Protein Folding*

However, the most vital change to which the polypeptide chain is subject is the step of protein folding [59]: here, it assumes its three-dimensional structure which ultimately determines the protein function. This can happen domain-wise in a more or less spontaneous fashion during the translation process (co-translationally) or after synthesis of the complete polypeptide chain. Though many proteins fold within microseconds, some of them require hours to achieve their native fold [60, 61] or even do not fold without further assistance by a special class of proteins, so-called molecular chaperones.

The folding process does not only depend on the amino acid sequence but to some extent also on the environmental conditions, such as pH and salt concentration, and is mainly governed by the amino-acid sequence and the resulting repulsive and at-

tractive forces, especially the burial of hydrophobic patches as well as interactions via hydrogen bonds between the residues (compare Section 2.2).

Molecular chaperones [62] which we briefly introduced in Subsection 2.1.4 can assist in the folding process and are essential for folding of some proteins. These chaperones provide favorable environments for protein folding as well as complex formation and can either actively assist the folding process under energy consumption (present in form of ATP) or by binding intermediate folding stages to prevent them from aggregation (which is likely to occur in an crowded environment such as cells) until the folding is completed.

Furthermore, disulfide bridges, i.e., a covalent linkage of cystein residues, may be introduced in secreted proteins (predominantly) by an enzyme called protein disulfide isomerase [63]. Some proteins also require the presence of additional cofactors that enhance the protein's functionality or activity, for example the hemoglobin presented in Subsection 2.1.4. These so-called prosthetic groups are typically also introduced during protein folding.

2.3.1.2 *Acetylation and Phosphorylation*

Not only the activity of genes but also that of proteins can be regulated via modifications. Two significant processes that can enable or disable protein activity are acetylation [64] and phosphorylation [65].

The acetylation usually takes place on lysine residues, replacing one of the hydrogens by an acetyl group. This mechanism requires the presence of an acetylase protein as well as an acetyl-CoenzymeA complex which provides the corresponding acetyl group. The attached acetyl groups can then modify the protein's activity or be recognized by other proteins and regulate their respective activities, as for example in the case of histones, where the (de-)methylation (acetylation by means of a methyl group) influences DNA transcription.

Phosphorylation works analogously to acetylation, however, in this case, a phosphate group instead of an acetyl group is attached by kinases (or removed by phosphatases). Target residues susceptible to phosphorylation are histidine, serine, threonine and tyrosine. A common property of these amino acids is their hydrophobicity; an addition of a three-fold negatively charged phosphate group turns their hydrophobic character into a highly polar one. As a result, a repulsion from surrounding hydrophobic residues and an attraction to residues with complementary charge or polarity takes place. These forces lead to changes in the biochemical properties of the functional site of the protein and sometimes also in its conformation, and eventually activate or deactivate the protein.

Phosphorylation is for example found in signalling pathways, e.g., as an mechanism of inhibition on the insulin pathway or degradation via the ubiquitination/proteasome pathway [66] (comp. Subsection 2.5.2) whereas acetylation is often associated with gene regulation and metabolism.

2.4 PROTEIN FUNCTIONS IN THE MOLECULAR MACHINERY OF A CELL

In Section 2.3 we have outlined some of the most essential life processes, and while this description is very superficial it nevertheless gives a good overview of the complexity of macromolecular mechanisms leading to the biochemical entity called cell.

Proteins are ubiquitous: so far, we have seen that they are relevant for DNA structure organization, replication, modification, transcription, and translation, but the tasks they have to perform are much more distinguished.

They perform structural tasks such as establishing the cell skeleton and preserving its structure, and are relevant for signal transduction, energy recovery, as well as metabolism. Via forming complex machineries they can assist in the synthesis of new proteins, can help with folding each other, or protect proteins from denaturation, digest proteins, RNA, and DNA. They are ultimately even responsible for the apoptosis – the cell's death (see Subsection 2.5.2).

Because proteins can be considered to comprise the machinery of the cell, understanding their properties, behavior, and interplay among each other, as well as with nucleic acids and other biochemical compounds provides a key path to the detection, alleviation and cure of diseases. In the following, we want to present some basic concepts of processes in which proteins play a major role and which are indispensable for a cell to function properly. While the following concepts might not be as well-separable in Nature as presented here, distinguishing them from each other might help to gain a deeper understanding of the basic principles.

2.4.1 *Enzymatic Catalysis*

Proteins catalyzing chemical reactions form one of the most essential classes of proteins: enzymes. The main function of enzymes is the accelerated and energy-efficient metabolization of biochemical compounds. They are often highly substrate-specific: they can only process a certain compound or class of compounds.

Typically, an enzymatic catalysis is performed in the following way: one or more compounds whose reaction is to be catalyzed, so-called substrates, serve as the educts of the reaction. These compounds bind non-covalently to the active site of the enzymes performing the specific reaction. Here, they are converted via a cascade of chemical events to the reaction products and released again [67].

By providing a well-defined environment for the catalytic reaction and assisting in the reaction itself, the energy barrier for the reaction to take place is decreased and the reaction equilibrium is reached much faster. In fact, without these catalytic reactions most of the biological processes relevant to sustain life would not be possible at a sufficient rate.

The reason for the tremendous increase in the reaction rate (up to several million times faster than in the unaided case) [68] is the lowering of the activation energy required for the reaction to take place. Several mechanisms to lower the activation energy have been observed: when a chemical reaction takes place, the reactants often enter an intermediate transition state before being transformed into the reaction products. This transition state can be stabilized for example via electrostatic effects or providing charges complementary to the functional groups of the reactants. In addition, enzymes bring the substrates in the correct orientation to each other or to the catalytic center; they thus increase the rate of correct encounters between the reaction sites which in turn increases the rate of metabolization.

This process is often aided by cofactors which provide/store additional energy or improve the chemical environment of the catalytic reaction. These cofactors can either associate with the protein upon reaction or can be bound permanently. In the former case, they are called coenzymes, in the latter prosthetic groups. There exist many

different kinds of cofactors, the most prominent of them being the anorganic heme group in complex with an iron (Fe^{2+}) ion required for example for oxygen binding in hemoglobin [69] (cmp. Fig. 2.6) and in cytochromes performing redox reactions such as the hydroxylation of steroid hormones [70], e.g., progesterone, or the electron transport in photosynthesis and the respiratory chain.

2.4.2 *Signal Transduction*

Signal transduction, also known as cell signaling, is a very important cellular process that enables the communication between cells and their environment. Typically, this communication occurs between cells in the same organism, but signal transduction between cells of different organisms has also been observed [71].

Proteins responsible for the mediation of signals from the cell's environment to its interior typically obey the following structural organization, consisting of three parts: one or more receptor domains at the cell surface, a transmembrane part that integrates the protein into the cell membrane, and one or several domains on the inside of the cell, responsible for the signal mediation into the cell interior. Proteins that are anchored in and cross the cell membrane are called transmembrane proteins, e.g. G-protein coupled receptors (GPCR's, only in eukaryotes) [72], ligand-gated ion channels [73], or receptor tyrosine kinases (RTK's) [74].

In general, the process of signal transduction can be summarized as follows: a messenger compound sent from the environment of the cell reaches the cell surface, where it is recognized by the specific receptor domain(s). Once that substrate has been received, i.e., bound to the receptor domain(s), the receptor is activated. Typically, the activation entails a change of the protein conformation. In the case of ligand-gated ion channels this results in the opening of the gate, enabling the corresponding ion(s) to pass. A ligand-induced conformational change in GPRC's in contrast activates the heterotrimeric G-protein (subunits: Gα, Gβ, Gγ), causing Gα to dissociate and expose the other two subunits (new results indicate the possibility that the G-protein might bind only after activation [75]). Both the Gα and the G$\beta\gamma$ units can then fulfill signaling functions. The activation of RTK's causes the monomeric RTK domains to dimerize through phosphorylation (see Subsection 2.3.1.2) of a tyrosine in each monomer on the intra-cellular side. This results in the creation of a binding site to which other proteins can bind and propagate the message. During any stage, the corresponding message can be amplified or received by multiple receptors in the cytoplasm. These receptors can then again trigger further actions, a phenomenon which is called signal cascade.

One important function relying on cell signaling is the development of immune responses against pathogens [76], i.e., foreign and possibly malicious substances, viruses, and microorganisms such as bacteria: here, for example, so-called T-lymphocytes (or T-cells) recognize specific antigens, i.e., small fragments of the pathogen, presented to them by surface proteins (so-called MHCs) of antigen-presenting cells. The recognition takes place via the T-lymphocytes' transmembrane T-cell receptor (TCR). The TCR then triggers a signal cascade leading to an activation of the T-cell and ultimately a proper immune response, in the case of T-killer cells the search and destruction of cells with the same antigen.

Further examples of signal transduction pathways are the MAPK/ERK [77] pathway and the insulin signaling pathway [78]. The MAPK/ERK pathway involves

GPCR's and is associated with cell division. Defects on that pathway often lead to cancer, the result of unregulated cell growth and division. The insulin pathway in turn relies on RTK's and is involved in the regulation of metabolic pathways, e.g., the carbohydrate metabolism. Aberrations on that pathway can for example lead to diabetes mellitus type II.

2.4.3 *Gene Regulation*

The final target of signal transduction pathways are often gene regulatory processes. Especially the above-mentioned immune response and MAPK/ERK pathways strongly affect gene regulation. The purpose of this regulation is the alteration of the concentrations of the respective gene products, proteins in many cases, in the cell. During a cell's life cycle, not all proteins are needed at any time or can have adversary effects when present at the wrong time or in the wrong place. In addition, a change in the environmental conditions can require an adaption of the cell's behavior. This behavior strongly depends on the entirety of proteins and their concentrations, the proteome. Hence, it is vital to the cell to be able to regulate the activity of specific genes and thus influence the proteome [79].

Several different mechanisms of gene regulation are known. For example, the access to these genes and the frequency by which the gene products are synthesized can be regulated by epigenetic modifications [80, 81]. Through these modifications, the gene transcription rate can be increased, decreased or the gene expression can be silenced completely. Epigenetic modifications can either be acquired during lifetime as a response to the environmental conditions but are also heritable and represent an inherently essential prerequisite for tissue-specific gene expression and cell differentiation.

Another example of a regulatory mechanism, is the direct modulation of the gene expression level by affecting the activity of RNA polymerase: repression, activation, or enhancement [50]. The corresponding proteins comprise the class of transcription factors and bind to specific regions of DNA, that typically preceed the actual gene to be transcribed, either blocking or changing the capability of RNA polymerase to bind to the promoter region of the corresponding gene.

2.5 MACROMOLECULAR PROTEIN ASSEMBLIES

Many of the processes presented in the previous subsections require the formation of (at least transient) macromolecular assemblies. There exist a manifold of different reasons for such a behavior [82, 83]. First and foremost, one has to keep in mind that cells form a very crowded environment where no space is wasted. For any biological process to be performed at a sufficient rate, all involved macromolecular participants have to be located close to each other. Not only can these assemblies perform processes of which individual proteins are not capable, but often they also show a cooperative behavior. For example, hemoglobin [69] which is responsible for the oxygen transport in many higher-order organisms, especially vertebrates, consists of four subunits, two α- and two β-chains each of which is able to bind one oxygen molecule. The uptake of oxygen in one of the subunits induces a conformational change that also causes the other three subunits to assume a conformation as if they also had bound an oxygen.

Hence, binding of additional oxygens is facilitated by the uptake of the previous ones, an effect called cooperativity.

Another advantage is that, similar to the active site of an enzyme, a protein assembly can provide a well-defined environment where biochemical reactions might be facilitated and accelerated. This may also involve keeping toxic or highly-reactive intermediate products such as free radicals or electrons from entering the cytosol and cause damage to the cell, as for example in the electron transport system in the respiratory chain. Here, the energy of activated electrons is gradually transformed to chemical energy via a system of different redox reactions [84, 85, 86]. Releasing these highly reactive electrons into the cell might harm the cell in an unpredictable manner.

A further benefit is especially imminent for proteins involved in the same metabolic pathway. They all perform a cascade of enzymatic reactions (compare Subsection 2.4.1) and an assembly of these proteins into an oligomeric complex bears the advantage that the molecules to be processed can be directly passed from one protein to the next in line, similar to a production line in a factory. Furthermore, such complexes are capable of a very efficient mechanism – the feedback regulation [87, 88]. This mechanism allows for the adaption of reaction rates of the individual proteins to the environmental conditions present at the beginning and the end of the pathway. If, for example, the final substrates produced by the assembly pile up, the last protein in the pipeline may reduce its activity, which again causes its predecessor to reduce its activity and hence propagate the signal to the starting protein in the assembly.

Signal transduction and cellular responses (see Subsection 2.4.2) can also involve complexes. For example, the already mentioned cell-membrane-bound G-protein coupled receptors (GPCRs) which are able to detect and bind messenger molecules arriving outside the cell represent an assembly of several proteins, namely the transmembrane receptor and the G-protein. The G-protein itself is again a heterotrimeric complex consisting of an α-, β-, and γ-subunit, which dissociates upon signal transduction into the G$\beta\gamma$-subunit and a GTP (guanosine triphosphate)-activated Gα subunit which can then stimulate intracellular processes.

Even gene storage and their regulation (see Subsection 2.4.3) as well as the process of protein biosynthesis (see Subsection 2.3) heavily rely on such macromolecular assemblies. In the former case, we have for example the octamer-forming histones and the complexes RNA polymerase temporarily forms with DNA and transcription factors. An even more remarkable complex is the ribosome, the complex responsible for translation of mRNA into a protein during protein biosynthesis. It is not only formed by several proteins and a two units of a particular type of RNA, the ribosomal RNA, but is a very stable and active macromolecular assembly.

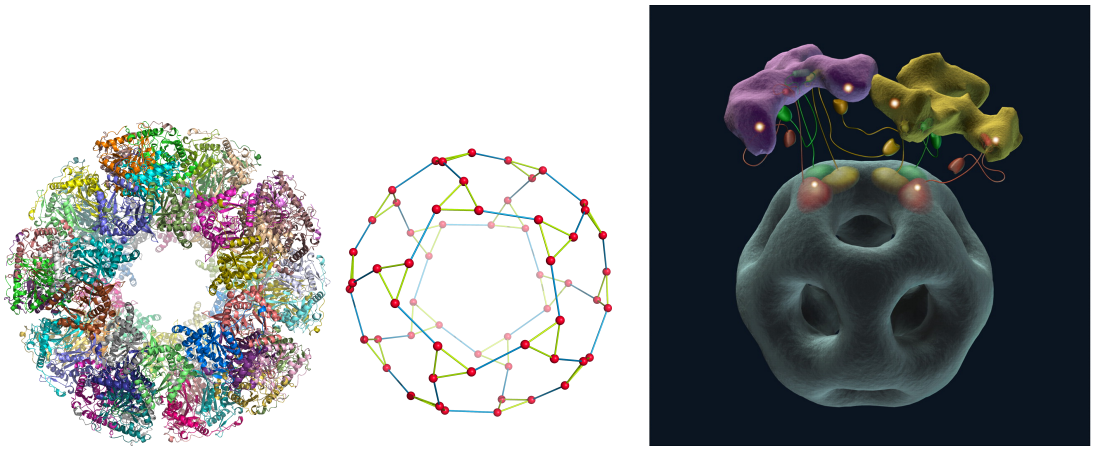
In the following, we want to address some assemblies of special interest with very integral functions in more detail.

2.5.1 *Pyruvate Dehydrogenase Complex*

The pyruvate dehydrogenase complex (PDC) is a large complex of mainly three different protein types that bridges the gap between glycolysis and tricarboxylic acid cycle. It's predominant function is the degradation of pyruvate – an organic acid produced by the glycolysis – in acetyl-CoA which can then enter the tricarboxylic acid cycle where it is further degraded and prepared for the ultimate energy recovery in the respiratory chain [89].

One of the most severe diseases associated with a malfunctioning of PDC is the pyruvate dehydrogenase deficiency caused by a mutation in the E1 alpha gene located on the X-chromosome. The resulting phenotype can take two forms, a metabolic and a neurological one. In the former, it leads to a lactic acidose with some symptoms being lethargy, nausea and tachycardia. In the latter case, the untreated progress of the disease can lead to spasms, blindness and mental retardation [90, 91].

The three main protein types the complex is composed of are the pyruvate dehydrogenase (E1), the dihydrolipoyl transacetylase (E2), and the dihydrolipoyl dehydrogenase (E3). The overall structure of the complex differs with 24 subunits forming a cubic core in gram-negative bacteria such as E. coli and an icosahedral composition (corresponding to a pentagonal dodecahedron) of 60 subunits in gram-positive bacteria and eucaryotes [92].



(a) Structure (pdb code 1B5S) and topology (spheres: protein centroids, edges: interfaces, colors: types) of the PDC core w.r.t. the five-fold axis.

(b) Model of the Pyruvate Dehydrogenase Complex w.r.t. the three-fold axis of the E2 core (gray) and satellite E1 (purple) and E3 (yellow) components[1].

Figure 2.7: The pyruvate dehydrogenase complex.

This core (Fig. 2.7a) is solely composed of dihydrolipoyl transacetylase (E2) proteins with E1 hetero-tetramers and E3 dimers attaching to it as satellite proteins (Fig. 2.7b). Here, three copies of E2 at a time are assembled into a trimeric structure forming one of the corners of the core body and are considered to be the centers of acetyl-CoA synthesis [93]. In eukaryotes, where the complex resides in the mitochondrial matrix, each of these E2 trimers can bind up to a trimer of E1 proteins, while one copy of the E3 dimer can be bound above the center of each of the faces, i.e., the planes surrounded by a pentagon of E2 trimers. In theory, the complex can thus consist of 60 E2 trimer copies, up to 60 E1 tetramer copies and 12 dimer E3 copies in the complex. However, due to mutually exclusive binding of E1 and E3, the total number of proteins forming the complex is typically thought to be 96 with varying relative stoichiometries of E1 and E3 (30:6 in mammals) [94, 95].

1 This research was originally published in Journal of Biological Chemistry. Milne, J. L. S., Wu, X., Borgnia, M. J., Lengyel, J. S., Brooks, B. R., Shi, D., Perham, R. N., and Subramaniam, S. Molecular Structure of a 9-MDa Icosahedral Pyruvate Dehydrogenase Subcomplex Containing the E2 and E3 Enzymes Using Cryoelectron Microscopy. *Journal of Biological Chemistry*. 2006; 281:4364–4370. © the American Society for Biochemistry and Molecular Biology.

The PDC is not only of great interest because of its pharmaceutical relevance but it is also one of the largest presently known complexes, having a diameter of approximately 250Å. It possesses some unusual features, the most remarkable one being the following: the edges between the corners of the complex, i.e., trimers of E2, are formed by very small interfaces between two instances of a trimer (in fact, one monomer in each of the two trimers is involved in forming that bridge). Hence, these brides are very flexible and allow the complex to "breathe" [96].

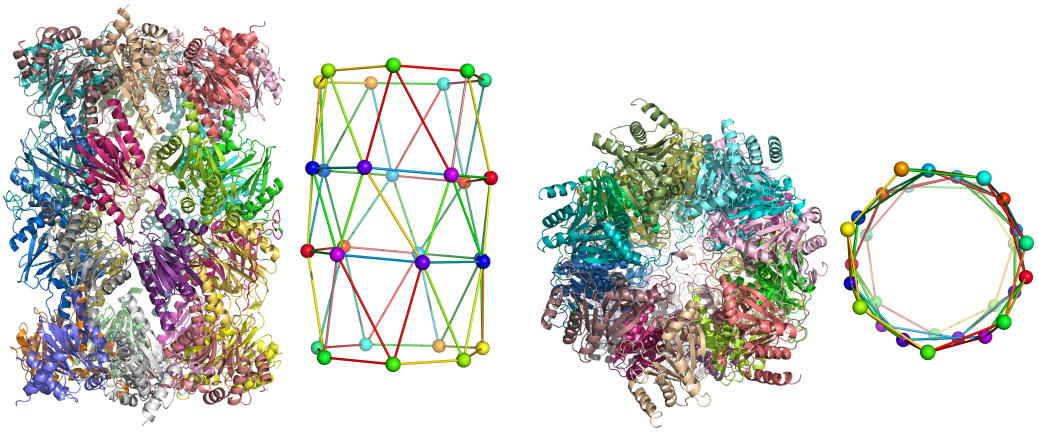
Due to the size of the complex, no high-resolution structure is available at present (a model of the E2 core with an resolution of 4.4Å can be found under pdb code 1b5s), and many details about the mechanisms of pyruvate degradation remain unknown.

2.5.2 *Proteasome*

The function of proteasome complexes, which are present in all eukaryotes as well as some bacteria and archaea, is to acquire and recycle resources from proteins that are no longer required, have been misfolded or damaged for example by a heat shock and are thus unable to further perform their task. The proteins to be degraded carry a certain kind of marker: a so-called polyubiquitin chain that is attached by the ubiquitylation system (which again forms a separate complex) [97, 98].

Proteasomes present a valuable pharmaceutical target, because the efficient and irreversible degradation of proteins by the proteasome can trigger the activation or repression of many processes in the cell: many regulatory proteins depend on normal turnover rates of the proteasome to function properly. In particular, inhibition of the proteasome in healthy cells can stop the cell division [99] or can induce the synthesis of glutathione. An increase of glutathione can help to protect cells from oxidative stress which in turn is assumed to be tightly connected to Parkinson's disease [100].

Cancer cells are especially susceptible to proteasome inhibition: here, blocking the proteasome activity can lead to the induction of a selective apoptosis of cancer cells [101, 99, 102, 103]. Though not completely understood, the inhibition of the proteasome in cancer cells is assumed to restore mechanisms related to cell proliferation and the suppression of apoptosis to normal function.



(a) Side view of the 2OS core particle structure and topology

(b) Top view of the 2OS core particle structure and topology

Figure 2.8: Structure (pdb code 1RYP) and topology (spheres: protein centroids, edges: interfaces, colors: types) of the 2OS proteasome core particle.

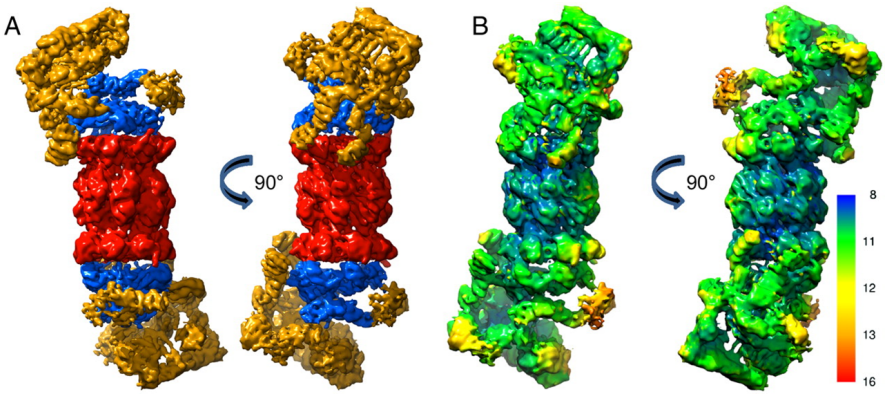


Figure 2.9: Electron density map of the the 26S proteasome (image source: [109]). (A) Components. Red: 20S core particle, blue: regulatory particles, orange: AAA-ATPase, (B) Isosurface colored by local resolution (Å).

The core of the proteasome (CP) is formed by the 20S subunit which consists of four stacked rings, each being a heptamer (Fig. 2.8a). Together, they form a hollow cylindric structure with an approximately 53Å-wide degradation chamber in its center (Fig. 2.8b). The overall size of the 20S subunit is about 150Å×115Å. [104, 105] The composition w.r.t. the involved protein types may differ among the organisms, but the overall topology remains the same: the 28 subunits of the four heptameric rings in the 20S yeast proteasome (e.g. pdb code 1z7q [106]) comprise 14 different protein types, each with a stoichiometry of 2. The outer two rings – the α rings – serve as gatekeepers that restrict the access to the degradation chamber and interact with regulatory components while the inner two β rings perform the catalytic reaction and the proteolysis.

Electron microscopy and X-ray structures of the 20S proteasome first became available in 1986 [107] and 1995 [104], respectively. A more complete structural model – the 26S proteasome (Fig. 2.9) which is composed of one 20S unit and two regulatory 19S caps (RP) and is assumed to be the predominant form of proteasomes in mammals – determined by an integrative approach combining data from various sources, has been proposed recently [108, 109].

Compared to the 20S core, the 19S caps are even more diverse in function and types of involved proteins [110]. Unfortunately, this also makes a direct structural determination much more difficult, because the binding affinities of the different proteins are very heterogeneous. Upon crystallization, the 19S particle dissociates into several sub-complexes and proteins of which some have not yet been structurally determined. Furthermore, some of the subunits are assumed to exhibit a considerable amount of conformational flexibility, making it hard to obtain an interpretable electron diffraction pattern for the these subunits [109].

Many details on the exact function of the involved proteins and their orientation are still unknown at present, hence, the structural prediction and investigation of the proteasome (and the ubiquitine/proteasome pathway) presents a field of intensive research.

2.5.3 *Viral Capsids*

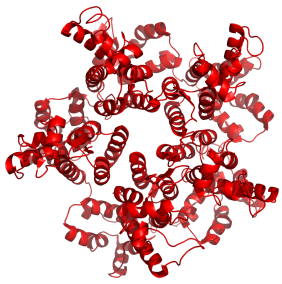
Viruses are the cause for a large number of diseases in all kinds of living systems. They require a host cell in which they can replicate because viruses themselves do not possess the complete replicative machinery nor do they show any kind of metabolic activity or cellular organization. On the other hand, they contain genetic information, can replicate (by using the host infrastructure) and are subject to selective evolutionary pressure. Hence, their membership in the realm of living organisms is a matter of ongoing dispute [111].

A virus typically consists of two components: the genetic material required for the replication and a protective hull of proteins surrounding the genome called capsid or coat. In some cases this coat is surrounded by an envelope of lipids acquired from the host cell. The viral capsid consists of proteins encoded by the viral genome and is synthesized by the infected host cell. They fulfill a variety of different functions, the most important one being the protection of the genetic material, but also the transport and binding to the host cell as well as the packaging of the genetic material. [112] These viral capsids are thus of great pharmaceutical interest as their destabilization and disintegration can severly affect the virus life cycle and destroy it before it is able to infect a new host cell. Hence, the knowledge of their structure can provide valuable information in the process of developing antiviral drugs.

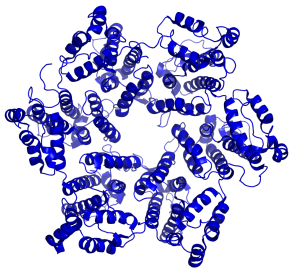
The structure of these capsids is very diverse and is considered to be a result of evolutionary pressure itself with the capsids exhibiting simpler topologies being the fittest [113]. The basic properties of capsid structure have been known since the 1950's [114]: they often assemble from sub-complexes of point-symmetric protein rings of varying stoichiometry, so-called protomers [115]. Spherical capsids, one of the most prominent types, typically consists of 12 pentameric rings (or a pentavalent subunit cluster) and a varying number of interconnecting hexamers (hexavalent subunit clusters) showing various types of symmetry [116, 117].

In recent years, more and more X-ray and electron microscopy structures have become available, allowing for a deeper investigation of the structural assemblies [118], a systematic representation of viral capsids as a kind of periodic table [113] and an assessment of their structural fold space [119]. One of the most recent achievements in capsid structure determination is the modeling of a mature HIV-1 viral capsid, consisting of 12 pentamers (Fig. 2.10a) and different numbers of hexamers (Fig. 2.10b): 216 and 186 (Figures 2.10c and 2.10d) [120].

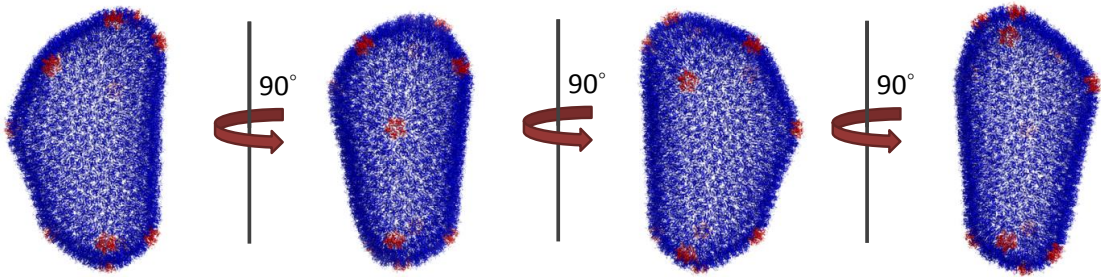
Besides rendering viruses inoperable or at least alleviating their pathogenicity by attacking their viral coat, understanding the principles and dynamics of capsid assembly is of great importance for the pharmaceutical treatment of diseases by the use of viral vectors [121, 122].



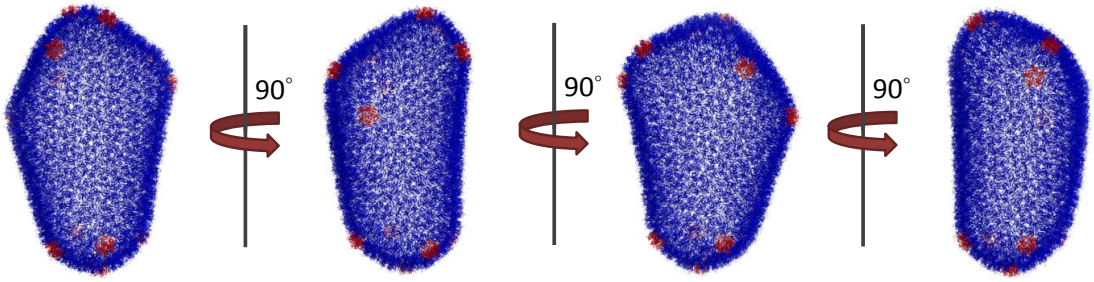
(a) Pentameric protomer of the HIV-1 capsid



(b) Hexameric protomer of the HIV-1 capsid



(c) Model of a mature HIV-1 capsid with 186 hexamers (blue) and 12 pentamers (red), pdb code 3J3Y.



(d) Model of a mature HIV-1 capsid with 216 hexamers (blue) and 12 pentamers (red), pdb code 3J3Q.

Figure 2.10: Protomers and models of the mature HIV-1 capsid.

3

EXPERIMENTAL AND COMPUTATIONAL TECHNIQUES AND RELATED APPROACHES

3.1 PROTEIN STRUCTURE DETERMINATION

The determination of protein structures is a key prerequisite to the work presented in this thesis. To this end, several methods have been developed, each with their own advantages and disadvantages. The two most common techniques are X-ray crystallization and nuclear magnetic resonance (NMR) spectroscopy. We will discuss the workflow of these methods as well as their advantages and shortfalls. Subsequently, we will briefly present some additional methods not yet suitable to compete against the aforementioned ones, but promising w.r.t. future developments.

3.1.1 *X-ray Crystallography*

The most prominent method to determine the structure of proteins, biochemical compounds and other macromolecules, is X-ray crystallography [123]. It is based on the observation that atoms in a crystal form regular, repeating patterns. This observation provides the key prerequisite to X-ray crystallography, because this regular arrangement of the atoms or molecules in the crystal allows for a sufficiently strong diffraction signal to resolve the molecule's structure using X-rays.

However, the crystallization of proteins and other macromolecules is hard to do and often involves many iterations of trial-and-error. The problems and limitations of protein crystallization are depicted in the next section. In this section, we first describe some of the basic concepts relevant to this work; for more detail, the interested reader is referred to [124, 125].

The general procedure is as follows: first, a crystal of the molecule under investigation is grown. Such a crystal must have a sufficient size (> 0.1mm) to afford signals that are strong enough for the resolution of the molecule.

X-ray beams are then sent through this crystal and detected on a screen behind the crystal. During their journey through the crystal, the beams are diffracted by the atoms, or rather their electrons, contained in the crystal. As a result, they produce a scattering pattern on the screen.

The produced diffraction patterns show spots at regular distances (known as reflections), and blank or blurred spaces at others, as predicted by Bragg's law: the scattered waves cancel in most directions, but are amplified in a few directions producing the spots. The amplification can only happen when the lengths of the traveled paths for rays scattered at different atoms under the same incident angle equals an integer multiple of the beam's wavelength. In that case, the beams scattered by the corresponding atoms remain in phase, i.e., they have the same phase when arriving at the screen, which leads to a constructive inference and an amplification of the signal. If that condition is not fulfilled, a phase shift has happened, leading to a less amplified or even canceled signal.

The diffraction pattern measurement is repeatedly performed from different angles, typically the crystal is rotated in between the measurements. From the set of measurements, an electron density map can be computed. Such maps then show the stochastic distribution of electrons, and hence the approximate positions of the corresponding atoms in the crystal.

From this electron density map, an atomic model can then be produced, which amongst others involves the application of chemical information about atomic valences, standard bond lengths and angles, removal of crystal contacts and packing artifacts, as well as the structural refinement and sometimes the fitting of secondary structure elements into the density map.

3.1.1.1 *Limitations of the crystallization process*

For the X-ray crystallization to succeed, sufficiently pure, regular and large crystals have to be grown from a solution containing the molecule in an accordingly large number of copies. However, this crucial step is often very difficult. The larger the molecule under investigation, the higher in general the number of degrees of freedom it exhibits. To obtain a good crystal of large proteins, special care must be taken to reproduce the native environmental conditions of the proteins to prevent the proteins from undergoing large conformational changes or even unfolding.

The crystallization typically starts with a small crystal nucleus around which the crystal growth can then take place. However, the conditions to produce the nucleus are intrinsically required to be different from those of the growth, and a good trade-off must be found in order to produce one single nucleus from which a large crystal can grow instead of many small nuclei or none at all. The determination of these conditions becomes extremely difficult with increasing size of the protein (or protein complex), and some environmental conditions, for example those of transmembrane proteins such as GPCR's which are natively found in a hydrophobic environment, the lipid bilayer of the cell membrane, are especially hard to reproduce. The prediction of good conditions is impossible, hence screening experiments covering a wide range of different conditions are often performed. In some cases, for example GPCR's, stablizing compounds or structural modifications which link certain key residues in the structure have to be introduced [126, 127, 128], if the crystallization is to succeed at all.

In the case of large macromolecular assemblies the problem of crystallization is twofold: first, there are typically no covalent bonds between the individual components of the complexes, and the overall topology is only preserved through non-covalent interactions in the complex interfaces. Hence, wrong conditions can cause the complex to fall apart. Especially in the case of hetero-oligomers, this can prove to be an infeasible problem, because each interface may require different properties, and conditions suiting the needs of all interfaces may simply not exist.

A second problem is the size of the complexes. As a rule of thumb, we can say, the larger a structure under investigation, the lower the resolution. This is due to the many atoms diffracting the X-ray beams. The resulting pattern of spots is smeared, and the computed electron density map is thus blurred, making an accurate estimation of the atomic positions not possible. Hence, the information content of the resulting crystal structure is reduced to where only a fold recognition or only statements about the

general topology but no secondary structure elements or even atomic positions are possible.

3.1.2 *Nuclear Magnetic Resonance Spectroscopy*

Another method to determine protein structures at an atom-resolution level is Nuclear Magnetic Resonance (NMR) [129, 130] spectroscopy. The main advantages of this method compared to X-ray crystallography are that NMR spectroscopy does not require the crystallization of proteins which can be a very difficult process and force the molecule into a nonnative conformation, as we have seen in Sections 3.1.1 and 3.1.1.1. NMR spectroscopy can thus capture protein structures in a more native state.

The underlying quantum-mechanical principle is that of nuclear magnetic resonance: atomic nuclei aligned in a magnetic field are able to absorb and re-emit electromagnetic radiation that is initially emitted by a radiation source perpendicular to the magnetic field. Only atoms and isotopes with a non-zero net magnetic momentum, which corresponds to the sum of magnetic momenta (spins) of the protons and neutrons of the atom nucleus, are detectable by this method.

The resonance of these nuclei arises from the absorption and emittance of energy upon changes in the magnetic momenta, i.e., transitions between spin states. The corresponding resonance spectra and frequencies are characteristic for each type of nucleus. However, in a molecule, shielding effects arising from neighboring atoms may occur: their electrons (or rather the distribution thereof) may induce a local magnetic field that shields the nucleus from the external magnet field. As a result, a variation of the resonance frequency of the nucleus w.r.t. a reference sample is observable, an effect that is called chemical shift. A measurement of the chemical shifts of all nuclei in the molecule at a given magnetic field strength is called an NMR spectrum.

The chemical shift, and thus the NMR spectrum depends on the strength of the applied magnetic field. Furthermore, the responses of nuclei in different neighborhoods to a change of the strength of the magnetic field are different. Hence, measuring the same structure at different strengths can provide information on the composition of that neighborhood, and combined over all nuclei, about the structure of the molecule under investigation.

Further indirect information on the structure can be obtained with methods employing Nuclear Overhauser Effect (NOE) spectroscopy [131]. This effect can be used to determine the distance between pairs of nuclei (most importantly protons) in space from multi-dimensional NMR spectra, with closer nuclei leading to a stronger NOE signal. The corresponding distance geometry information can then be used as additional restraints during structure calculation and optimization.

An additional advantage of NMR spectroscopy over X-ray crystallization is, that it can also be used to study dynamic processes such as conformational changes in the structure, because it investigates proteins in their native state. This is possible because the local neighborhoods of the individual atoms and hence the chemical shifts change when a conformational change occurs. The resulting collection of structures is called an NMR ensemble.

However, NMR spectroscopy is generally limited to smaller protein structures (less than 25 to 50 kDA [131]) and thus less well suited for the structural determination of oligomeric structures. Yet, TROSY-based (Transverse Relaxation Optimized SpectroscopY) NMR methods [132] which are experimentally very demanding (NMR spec-

trometer with > 700 MHz operating frequency with an assumed optimum at 1Ghz, probe temperatures below $-150.0\,^{\circ}\mathrm{C}$ and most protons in the macromolecule replaced with deuterons [131, 133]) can achieve a better sensitivity already for molecules with molecular masses greater than $15 - 20$ kDA and can be used to study complexes with molecular masses of up to 1 MDa, for example the 900K GroEL–GroES complex [134].

3.1.3 *Cryo-EM and Cryo-ET*

Cryo-electron microscropy (cryo-EM) [135, 136] and tomography (cryo-ET) [137, 138, 139] are two related methods that gain more and more relevance in protein structure determination and prediction. Though their resolution is not yet comparable to that of X-ray crystallization or NMR spectroscopy, they nevertheless represent promising approaches due to their ability to determine much larger macromolecular structures as well as cells and even supercellular objects.

To this end, the target structure is cooled down to a cryogenic temperature ($< -150.0\,^{\circ}\mathrm{C}$). Then, similar to X-ray crystallography, electrons emitted by an electron microscope or tomograph are scattered w.r.t. the investigated structure. The three-dimensional structure can then again be obtained from the combination of such scattering patterns taken from different angles.

Currently, their predictive value lies in the fact that they can be combined with high-resolution techniques in a hybrid fashion: the corresponding high-resolution structures can be fitted into the overall topology of, for example, oligomeric assemblies provided by the EM or ET electron maps [140, 141].

Depending on the resolution of the underlying density map, the fitting can be performed manually using visualization tools or algorithmically: in the latter case, the fitting of the high-resolution monomers is for example done by optimization of the cross-correlation between fitted monomer and density map. However, rigidly fitting monomers into an EM map can be hampered by the fact that the monomers in the high-resolution crystal structure may adapt a different conformational state than in the assembly given by the density map. This can for example be due to conformational changes during the assembly process but also depend for example on the crystallization conditions of the high-resolution structure [142]. Hence approaches accounting for flexibility during the fitting procedure, for example by molecular dynamics simulations [143, 141] or elastic network models (ENM; a definition is given in Section 5.1), can be required [144, 145].

The process of fitting high-resolution monomers into low-resolution density maps is to some extent complementary to the research we present in Part iii, i.e. the assembly of macromolecular oligomeric assemblies from pairwise dockings between protein monomers: for example, the complexes obtained with our approach can be post-scored against such density maps using established quality-of-fit measures applied in the above fitting procedures. Furthermore, information about binding modes in the assembly can for example be deduced from density maps and in turn be used to generate pairwise docking poses which can be used in the assembly process.

On the other hand, our approach does not rely on the availability of such density maps but can use a manifold of different data sources to derive pairwise docking poses (see Section 3.5 and Chapter 8) and can thus even be applied in cases where either the fitting procedure fails or low-resolution density maps are not available.

3.2 PROTEIN STRUCTURE CLASSIFICATION

To determine structural relationships and potential common ancestors, a classification of protein structures is often useful. For example, a rule of thumb states that protein function is conserved between proteins with a sequence identity of about 30%–40% or more [146]. However, finer classification schemes are often required. In the following, we address the two widely used classification schemes SCOP [147] and CATH [148], and subsequently shortly discuss further approaches of classification.

3.2.1 *SCOP*

SCOP (Structural Classification of Proteins) is a widely used classification scheme among structural biologists [147]. The rationale behind this scheme is the grouping of proteins according to their functional and evolutionary relatedness. It is mostly manually curated and thus not as extensive w.r.t. the number of classified proteins as other methods. However, the accuracy is generally considered to be higher than that of (semi-)automated methods (see Subsection 3.2.3), because particularly distant evolutionary relationships are difficult to resolve without expert knowledge.

The unit by which SCOP classifies proteins is that of a domain. As stated in Subsection 2.1.3, a unique definition of the term domain does not exist, and hence the SCOP curators have their own notion of a domain, which they define as "an evolutionary unit observed in nature either in isolation or in more than one context in multidomain proteins" [43] and as "a region of the protein that has its own hydrophobic core and has relatively little interaction with the rest of the protein, so that it is essentially structurally independent" even though often evolutionary information must be taken into account for a correct annotation [149]. In particular, the SCOP curators consider most proteins, i.e., those with a small to medium number of residues (though no exact value is given), to consist of a single domain, multiple domains are in general only present in large proteins.

The hierarchical classification scheme of SCOP comprises a bottom-up clustering of protein domains at the following levels of decreasing similarity: family, superfamily, common fold and class. The first two primarily account for structural and evolutionary relationships, while the latter two solely consider structural similarity.

Proteins belong to the same family cluster if they are assumed to have a common evolutionary origin, which is defined by the authors as having more than 30% sequence identity or, if they are known to perform the same function, a sequence similarity between 15% and 30% sequence is sufficient [147]. If these criteria are not met, proteins can still belong to the same superfamily cluster if their structural and functional features indicate a distant common ancestor.

If two proteins are not assumed to share a common ancestor, but exhibit the same secondary/supersecondary structure content and overall structural topology, they are part of the same common fold cluster. Here, the structural similarity is assumed to be due to a preference of certain packings and structural topologies of domains induced by the physics and chemistry of proteins [147].

Two clusters on the common fold level can then be further grouped into the same class, mostly depending on the secondary structure elements they are composed of: i) all-α for domains consisting predominantly of α-helices, ii) all-β for those mainly comprised of β-sheets, iii) α/β for domains with intermixed α-helices and β-sheets, iv)

α+β for those with separated α-helices and β-sheets, and v) multi-domain containing proteins without known homologues and a different fold. The classification scheme at the class level is however somewhat illogical because, in addition to the aforementioned classes that are purely based on the secondary structure element content, the class level also provides several classes based on different criteria: classes accounting for more unusual proteins such as membrane and cell surface proteins, small proteins, peptides, as well as theoretical models, nucleic acids, and carbohydrates.

It is worth noting that on November 29, 2013 a prototype for a new SCOP version, termed SCOP2, became available [150]. The main improvement of SCOP2 is that it essentially replaces the tree-like structure of previous SCOP versions by a directed acyclic graph where each node represents a particular type of relationship, e.g., the evolutionary and structural relationships explained above. Each child node can have more than one parent, hence, SCOP2 can account for more complex relationships between proteins and domains than SCOP. In addition, it differentiates between structural and functional classes and thus compensates the inconsistent classification scheme of previous SCOP versions. However, the work in this thesis relies on the classification scheme as present in SCOP 1.75, hence, for more detailed information on differences between SCOP and SCOP2, the interested reader is referred to *Andreeva et al.* [150].

3.2.2 *CATH*

In contrast to SCOP, CATH (Class, Architecture, Topology, Homologous superfamily) [148] is a semi-automated top-down structure classification method of protein domains, relying mainly on automated techniques for classification and prediction, for example CATHEDRAL [151] for the automated determination of folds and domains from multi-domain protein structures, or Hidden-Markov-Models for the assignment into superfamily classes [152], but it also incorporates expert knowledge on the Architecture level.

In CATHEDRAL, a domain is defined as an individual folding unit of evolution that adopts a specific fold. However, automated domain (boundary) assignment is difficult, because domains can considerably vary in their compactness and their separation in protein structure and sequence. In particular, discontiguous domains where the secondary structure content of the domain is distributed over several discontinuous, discrete parts of the protein sequence [42] pose a hard challenge to fully automated methods. Hence, in CATHEDRAL, domains of a query protein are detected iteratively through structural comparison against the contents of a library of manually curated domains in the CATH database.

The top level in the CATH classification hierarchy is the Class level (C-Level), where domains are classified according to the secondary structure elements they contain, i.e., Class 1 comprises domains consisting mostly of α-helices, Class 2 those that are mainly composed of β-sheets, Class 3 encompasses domains that provide a significant number of both α-helices and β-sheets, and Class 4 is reserved for domains that contain a very small of amount of secondary structure.

Each of these four classes is then split into Architecture (A-Level) categories with each category containing domains with similar three-dimensional arrangement of the secondary structure elements within the domains, regardless of their connectivity. The assignment of the domains to the individual architecture categories is done man-

ually, taking into account the description of the overall structure in the literature (for example barrel-like; compare Fig. 2.5d).

Each architecture category is then further partitioned into fold groups, also called topologies (hence T-level), which in addition to the three-dimensional arrangement of the secondary structure elements on the A-Level also account for their connectivity.

The bottom level is the H-Level where domains from a T-level category are subdivided into homologous superfamilies. Criteria for classification are structural, sequence and functional similarities. Further sub-categories at the H-level exist and depend on the sequence similarity of the contained proteins, yielding families in four different levels: S, O, L, I with all proteins in the same family having $< 35\%$, $< 60\%$, $< 95\%$, and 100% sequence identity, respectively.

Because of the semi-automated nature of CATH, it contains a considerably higher number of domains than SCOP (173,000 domains from 51,334 PDB entries in CATH vs. 110,800 domains from 38,221 PDB entries in SCOP v1.75) and 1313 vs. 1195 folds (though fold classification entails more subjective criteria, and is thus less meaningful) [148]. However, because the SCOP entries are all manually curated, the SCOP database is believed to be more accurate.

Often, differences between SCOP and CATH are due to the philosophy behind the expert-based domain assignment. The alleviation of these differences to some extent is one of the aims of the Genome3D consortium, which was brought to life in January 2013 [153]. It consists, amongst others, of experts from both the CATH and SCOP groups which aim at a more accurate and unified classification of superfamilies as well as providing information on the philosophy behind alternative groupings of related proteins. To this end, Genome3D provides a mapping between SCOP and CATH entries which is approved by curators from both groups.

3.2.3 *Other Structure Classification Methods*

Besides the previously presented classification approaches SCOP (manually curated) and CATH (semi-automated), several fully automated methods exist. One of them, SUPERFAMILY [154] is strongly relying on the corresponding SCOP superfamily classification. Given such a SCOP superfamily classification, Hidden Markov Models (HMM) [155] are learned via an automated post-processing step from multiple sequence alignments of the SCOP superfamilies. Using these HMMs, the superfamily of a given query protein sequence can the be predicted automatically. Following SCOP releases, the latest release is SUPERFAMILY 1.75 [154].

The second method is called FSSP (Family of Structurally Similar Proteins) [156] and contains a collection of fully automated annotations and classifications of structurally superimposed proteins into more than 330 families, each with a representative protein chain. However, the FSSP service retired in 2014, while SUPERFAMILY is still under active development.

3.3 PROTEIN-SMALL MOLECULE AND PROTEIN-PROTEIN DOCKING

In Section 2.2, we have pointed out that the functionality of most proteins is associated with interactions with small biochemical compounds, proteins or other biological macromolecules such as DNA and RNA. However, experimental determination

of such interactions is not always possible or too time-consuming, for example in the process of drug screening, an approach to find novel active compounds (hit structures) for a given protein target in the process of drug development.

In such cases, one aims at a computational prediction of possible interactions and the strength of their binding. Computational methods capable of such predictions are called docking methods. The molecules to be docked are generally named ligand and receptor. The ligand denotes the structure which is kept mobile during docking while the receptor, a protein to which the ligand is docked, is typically kept fixed. In protein-small molecule docking, the ligand is a small chemical, often drug-like compound while in protein-protein docking, the ligand refers to another protein (or peptide).

Docking methods typically consist of two essential steps: A sampling phase and a scoring phase. In the sampling phase, the orientation of the ligand w.r.t. to the receptor (and depending on the approach also the ligand and possibly receptor conformation) is probed, leading to a set of receptor-ligand complex candidates, so-called poses. In the scoring phase, the quality of these poses is then assessed w.r.t. the strength of their interactions. This is typically done by estimating the binding free energy (equations 2.1 and 2.2) by the means of a scoring function [157].

Even for rigid docking methods, where no conformational changes are introduced, sampling and scoring can become computationally very intensive. However, conformational changes are often required because neither ligand nor receptor typically have the same conformation in the bound (holo) and unbound (apo) state. The amount of protein flexibility that needs to be modeled depends on the situation: in the simplest case, the protein is known to be rigid or the used holo conformation (obtained for example from a protein structure database such as the PDB, see Section 3.5.1) can be assumed to be similar to that with the ligand in question bound. This is often the case in drug screening experiments, where large libraries of biochemical compounds are searched for structures that are similar to a reference ligand with known protein holo conformation. However, if no appropriate holo structure is available or if one strives to determine compounds with new binding modes, protein side chain or even backbone flexibility has to be taken into account. This problem is even more imminent in the protein-protein docking case where the number of degrees of freedom can become much greater, depending on the allowed degree of flexibility in both binding partners, and differential binding affinities are smaller, especially given the fact that often proteins are known to interact but the structure and topology of the protein-protein complex is not known.

Hence, to avoid extensive sampling of unfavorable regions in the solution space and thus improve the computational efficiency, the scoring is in general already performed in the sampling stage to guide the sampling towards favorable receptor-ligand interactions. To this end, the score of each pose must be quickly computable, which often restrains the scoring function to comparatively simple sums of terms that approximate the true interaction energies more or less well. Consequently, the poor power of discriminating near-native poses from decoys (incorrect poses) as well as the accuracy in estimating the free binding energy, is still a major problem of scoring functions [158, 159, 160]: while algorithms are very often able to sample a near-native pose, an inaccurate scoring in the presence of a large number of decoys often leads to a bad ranking of near-native poses.

In particular, one typically distinguishes between three types of scoring functions (comprehensive lists of scoring functions and their respective type can for example be found in [161, 162]): force-field based, empirical, and knowledge-based scoring functions. Force-field based functions employ a physics-based approach using molecular mechanics force fields such as CHARMM [25]. They typically include several terms for covalent and non-covalent interactions. Covalent forces comprise terms for deviations in bond lengths, bond angles, and torsional angles, the non-covalent forces a Lennard-Jones potential accounting for attractive and repulsive forces between pairs of uncharged atoms as well as a Coulomb term for attractive forces between pairs of charged atoms. In general, entropic effects have to be considered as well, but are often neglected because they are difficult to model. In comparison, knowledge-based and empirical scoring functions often employ simpler terms. Knowledge-based scoring functions rely on the concept of classical statistical physics, i.e., that the distribution of the interaction geometries (for example atomic distance) between pairs of atom types in a given data set, e.g. the protein-ligand or protein-protein complexes in the Protein Data Bank (see Section 3.5.1) or a set of drugs known to bind to a biological target, can be used to derive pairwise potential functions that reflect the observed interaction geometry distributions for any pair of atom types present in the data set. Using these pairwise potential functions, the score then corresponds to the sum of the contributions of all pairs of atoms a, b between receptor and ligand determined by the respective pairwise potential for the atom types and the observed interaction geometry of a and b. In contrast, empirical scoring functions mainly consist of additive terms that reflect physical effects known to contribute to the binding affinity. The contribution of each term to the docking score is scalable by an individual parameter. The individual scoring functions differ in their parametrization and the modeling of the physical effects but typically include terms accounting for hydrogen bonds, polar interactions, hydrophobic and steric effects as well as the loss of translational, rotational, and torsional degrees of freedom. Furthermore, two additional types of scoring functions are worth mentioning in this context: mixed scoring functions incorporating terms from several of the above classes as well as the so-called consensus scoring functions which calculate the score of a docking pose from the weighted contribution of the scores obtained from several other scoring functions [161, 162].

Many different docking approaches exist, both in terms of the underlying algorithm as well as the type and parametrization of the scoring function. Each of them has its own field of application where it performs well and others where its suitability is questionable. Many studies trying to compare different docking methods have been performed, however there exists no algorithm that is clearly superior to the others [163, 164, 165, 166, 167, 168]. In the following, we present some popular protein-small molecule and protein-protein docking approaches of different complexity that are used throughout this work, further relevant and widely used docking algorithms are shortly addressed in Chapter 4. Subsequently, we present some approaches related to this thesis that algorithmically assemble macromolecular oligomeric assemblies.

3.3.1 *FlexX/FlexE*

FlexX [31] and FlexE [169] are two docking tools used for protein-small molecule docking. The difference between those two is that FlexX keeps the protein absolutely rigid while FlexE is an extension that can additionally handle protein flexibility. How-

ever, it is not capable of introducing that flexibility on its own but rather relies on an externally provided ensemble of different superimposed protein conformations into which the ligand can be docked. The conformational space is hereby mainly restricted to different side-chain orientations but small conformational changes in the backbone are also tolerated.

Both methods implement an iterative so-called incremental construction algorithm: in the preparation phase, the ligand in question is first cut at rotatable bonds into rigid or almost rigid fragments to break down the conformational space that has to be handled. From this set of fragments, a so-called base fragment is determined, which ideally is large, almost rigid and provides many hydrogen bonds. Each ligand atom is assigned an interaction geometry, fragments containing rings are optimized with CORINA [170]. Analogously, interaction geometries are assigned to the proteins.

In the first iteration of the incremental construction, each of a small number of selected base fragments is placed into the binding pocket at positions that maximize the complementarity of interaction geometries and thus the score between protein and ligand. Different scoring functions are available, by default, F-score [31] which is based on Böhm's empirical scoring function LUDI [171, 172] is used. It comprises terms for electrostatic interactions (ionic interactions and hydrogen bonds) as well as entropic, aromatic and hydrophobic effects.

In each of the following iterations, the candidate poses are each extended by another fragment (under the condition that the original covalent bond to the already placed fragment is restored, using appropriate torsion angles from the MIMUBA database [173]) such that additional favorable interactions are established.

Because many solutions obtained in a particular iteration can be expected to be similar, geometric hashing is applied to find these solutions and reduce and diversify the space of candidates, from which only the k best-scoring ones are kept for the next iteration (known as k-greedy scheme).

In FlexE, the principle of ligand fragmentation is also applied to the protein: residues corresponding to conformationally different parts of the binding pocket are cut into fragments at peptide and backbone-side chain bonds. From these fragments a unified protein description is derived. Upon incremental construction, an appropriate protein conformation for each ligand candidate fragment is determined using a self-consistent mean field (SCMF) approach.

3.3.2 *GOLD*

GOLD (Genetic Optimisation for Ligand Docking) [174] employs a genetic algorithm (GA) for protein-small molecule docking with protein and ligand flexibility. In genetic algorithms, the degrees of freedom of a specific problem to be optimized are encoded in a special data structure, that in analogy to genetics is called chromosome. In GOLD, these degrees of freedom correspond to the angles of rotatable bonds, separated into two binary strings (chromosomes): one encoding the rotational angles of the protein, the other of the ligand. Two additional integer strings are used that map possible hydrogen donors in the ligand to acceptors in the protein and vice versa.

Initially, a population of candidates with random values for these features is created and evaluated by a fitness function. This fitness function mimics the process of evolutionary selection. To this end, GOLD first performs a least-square fitting of the mapping of hydrogen bonds, i.e., the computation of an orientation of the ligand w.r.t.

the protein that maximizes the number of hydrogen bonds w.r.t. to the torsional configuration of protein and ligand. Subsequently, the energy of the hydrogen bonding pattern, a complex energy term containing the steric interactions between both partners and the internal energy of the ligand are calculated and comprise the total fitness score. Besides this fitness function, called GoldScore, the current version of GOLD also provides three additional scoring functions: ChemScore, ALP and CHEMPLP [174] as well as the possibility to alter existing or implement new scoring functions.

Once the fitness score for all candidates has been evaluated, the fittest members of the population are chosen to breed children. Children are generated according to the so-called island model which assumes that there exist different sub-populations which are locally optimized w.r.t. to the environmental conditions. In this model, three different evolutionary operators are applied: cross-over, mutation and migration. Mutation is the simplest one, altering one of the degrees of freedom of the parent. In cross-over, two parents are chosen to exchange a certain fraction of their chromosome and in migration a member of one sub-population is copied to another sub-population, providing more diverse features which may prove beneficial upon cross-over in later generations.

The generated children then replace the least fit members, if not already present in the algorithm. This process it then iterated 100,000 times (by default) and the final set of solutions is returned.

3.3.3 *AutoDock*

In its first version, the protein-small molecule docking algorithm AutoDock implemented a Simulated Annealing (SA) [175] approach to solve the problem of protein-ligand docking. Here, the sampling converges from a global search at high temperatures, where large changes in the states of instances of an optimization problem are allowed, to a local search in later iterations, where the magnitude of the transition between two states is bounded by low temperatures.

While retaining the SA functionality, versions 3 and 4 now use a genetic algorithm by default [176, 177]. The modus operandi of a GA has been described in the previous section, introducing the docking program GOLD. In addition, AutoDock 4.2 features, besides ligand flexibility, also the definition of a number of flexible residues of the receptor [178].

However, contrary to GOLD, this genetic algorithm is enhanced by an adaptive local search component, where individual members of the population are not only generated by genetic operators, but are also allowed to locally optimize their features on their own. Through learning from the outcome of previous optimization trials, the local search adapts the step size of the minimization procedure according to whether previous trials were successful or not.

Methods combining GA's with local search (LS) heuristics are called Lamarckian Genetic Algorithms (LGA) [179]. Because they overcome the rather coarse-grained global sampling of conventional GA's by the use of a computationally expensive local optimization, they can be expected to increase docking accuracy. However, depending on the complexity of the scoring function used for local optimization, this may result in a increase in total computational time. Yet, on the other hand, directed optimization, especially when performed on only a small fraction of the population (6% was found to already be sufficient for AutoDock [176]), often helps to reduce the total number of

generations required to find a plausible solution, and can thus also often reduce the total running time of the algorithm.

To restrict the torsional space to be sampled during optimization to reasonable areas and to allow for an easy switch between GA and LS, the torsional degrees of freedom are real-valued as compared to the binary strings employed in GOLD. In addition, translational and rotational degrees of freedom are encoded in the chromosomes.

Further differences to GOLD are that AutoDock does not consider island populations, i.e., there is no migration operator. However, cross-over and mutation are used consecutively, i.e., a child is generated first by cross-over of the chromosomes of two parents and then mutated w.r.t. to translational, orientational and torsional degrees of freedom at random according to a Cauchy distribution. A user-defined number of children can then be optimized using LS. The children then replace the parents in the new iteration as compared to GOLD where the parents are kept if their fitness score permits it.

AutoDock uses a more sophisticated and computationally expensive scoring function than GOLD which implements energy terms for hydrophobic and repulsive (van der Waals) forces, hydrogen bonds, electrostatic, torsional and solvent effects. To reduce the amount of computation time required, AutoDock relies on a predefined grid where for each lattice point, the corresponding energy terms for all pairs of atom types are precalculated.

3.3.4 *RosettaDock*

RosettaDock [180, 181] is a more versatile docking tool than the previously introduced algorithms in the sense that it is mainly designed for the purpose of protein-protein docking, but like the previously presented algorithms can also perform protein-ligand docking. RosettaDock employs a Monte-Carlo (MC) based sampling approach performing rigid-body pertubations of the receptor-ligand complex configuration. The rationale behind this method is that biophysically relevant encounters between receptor and ligand should occur more often than random contacts.

The algorithm consists of two stages which are iteratively repeated and can be individually switched on or off. The first stage is a low-resolution stage that represents the protein side chains by pseudo-atoms called centroids. In the second stage, a full-atom optimization is employed that also allows for side-chain flexibility.

During the low-resolution stage, 500 sampling steps (by default) altering rotational and translational degrees of freedom according to a Gaussian distribution are performed. Complex scores are computed using four terms: a contact term, a term for steric clashes, as well as two terms derived from Bayesian statistics estimating the plausibility of each candidate pose: a residue-residue specific energy term as well as a term accounting for the residue environment. The lowest-energy conformation in this stage is then taken and optimized in the second stage.

Here, the side-chain centroids used in the low-resolution stage are first replaced by the all-atom side-chain conformations from the unbound state of the binding partners. Subsequently, the position of the second binding partner is slightly perturbed according to a Gaussian distribution and minimized. Finally, side-chain optimization is performed by rotamer trials w.r.t. the Metropolis criterion [182] which forces the algorithm to sample low-energy conformations more efficiently and thus leads to a faster convergence of the optimization algorithm.

In the second stage, a high-resolution all-atom scoring function is used. The computed terms account for Van der Waals effects, solvation, hydrogen bonding, electrostatics, side-chain conformation energy, and pairwise residue-residue interactions.

RosettaDock is capable of both local and global docking, i.e., a perturbation around a predefined input dimer configuration or a sampling of the full orientational space of receptor-ligand complexes. However, global docking is only recommendable when protein and ligand have less than approximately 450 residues in total [183].

3.3.5 *CombDock*

CombDock [184, 185] is a combinatorial algorithm for assembling macromolecular homo- and hetero-oligomeric assemblies from all-vs-all pairwise dockings. The algorithm consists of three separate stages.

In the first stage, dockings between all pairs of proteins are performed. In particular, if the complex consists of N components, the number of pairwise dockings to be performed amounts to $N(N-1)/2$. This is also done for homo-multimers, where all components are identical and in principle only one pairwise docking would have to be carried out. The used docking algorithm performs a global docking relying primarily on geometric shape matching. First, for each protein, a molecular grid and surface representation is created, and local features of the protein surface are determined. Subsequently, the binding partners are matched w.r.t. a maximization of complementarity of these local features. Finally, the solutions are clustered and scored w.r.t. shape complementarity. The underlying algorithm combines two different methods [186, 187] and follows a similar principle as PatchDock [188, 189].

From each of the pairwise dockings the K best-scoring solutions (100 by default), represented as transformations, are retained and used during combinatorial assembly. The combinatorial assembly is then treated as graph theory problem: Each protein in the complex is represented by a node, and each of the retained solutions forms an edge between the corresponding vertices, weighted by the score of the solution, yielding a complete multi-graph with K edges between any pair of proteins. Each subset of $N-1$ edges such that all nodes are connected, i.e., a spanning tree, forms a potential complex. However, two restrictions have to be considered: first, edges may be incompatible in that sense that the corresponding transformations induce steric clashes when applied to the corresponding proteins. Second, spanning-trees built from transformations with a good score are considered to be more realistic. Hence a clash-free minimum spanning-tree (MST) is required.

An exhaustive search for the MST is NP-complete, hence spanning-trees are created hierarchically and selected by a greedy approach. Initially, each vertex forms a separate tree, and in each subsequent iteration two trees are joined via an edge and kept, if no significant steric clashes are produced. Since the search space is still very large, only a parameter-controlled subspace is searched.

The final scoring is performed using a function accounting for geometrical and chemical compatibility (non-polar buried surface area). Solutions are then ranked and clustered to remove redundancy.

3.3.6 *HADDOCK*

HADDOCK (High Ambiguity-Driven DOCKing) [190, 191] is a multi-body docking program that can assemble complexes with up to six monomers. The docking process can be efficiently guided by constraints derived from data obtained by experimental or bioinformatics methods. It relies on the assumption that the protein complexes to be assembled exhibit a symmetry, where arbitrary combinations of two-, three-, and five-fold cyclic symmetries are supported.

Based on the supplied interaction data, active residues are defined as those residues taking part in an interaction, passive residues are the solvent-accessible ones around the active ones. In addition, for interacting residues, ambiguous interaction restraints (AIR) are generated, demanding that the corresponding residues should be close in the resulting complex. From these, an AIR network is generated.

The three-stage docking protocol of HADDOCK is as follows: the first stage consists of a rigid-body energy minimization of the molecules to be docked, the second stage performs a refinement of the conformation of the contacting residues expressed as torsional angles, and the last stage comprises a flexible refinement in explicit solvent. The scoring function employed during this protocol contains van der Waals, electrostatic, desolvation, buried surface area as well as AIR and symmetry violation terms.

Finally, a clustering based on an RMSD [192] cutoff of 7.5Å and a final scoring and ranking based on the average score of each cluster is performed.

3.3.7 *ClusPro Multimer Docking*

This N-mer assembly algorithm developed by *Comeau et al.* [193] generates symmetric complexes of homo-oligomeric assemblies of up to six units. Though it does not require any input parameters, it nevertheless assumes that the assembled complex will exhibit a certain kind of symmetry. Consequently, all possible symmetries are tried during the assembly.

The algorithm requires a single monomeric structure and follows a six-stage protocol consisting of the following steps: first, in excess of 20,000 poses between two identical instances of the monomer are generated using the DOT docking program [194], employing a global docking, i.e., an unrestricted sampling of the full rotational and translational space of the ligand molecule around the receptor. From these, the top 500 solutions w.r.t. desolvation energy and the first 1,500 w.r.t. electrostatic interaction energy (using the scoring function implemented in ClusPro [195, 196]) are kept. In step three, depending on the number N of monomers in the complex, all possible symmetries are then tried via translational and rotational search along the corresponding symmetry axes. The total score of each solution corresponds to the sum of scores of the corresponding dimer interactions. Fourth, a clustering is applied according to a 5Å C_α RMSD [192]. Each cluster is then assigned the maximum score over all solutions contained therein. Finally, from each cluster the solution showing the best overall symmetry is retained.

The obtained solutions are then minimized with CHARMM [25] for 300 steps and returned.

3.4 MODELING OF PROTEIN FLEXIBILITY AND PROTEIN COMPLEXES

The structural modeling of proteins is an important area of research relevant for drug design and prediction of protein-protein interactions. Several approaches exist, among them *de novo* methods that do not require any structural information but instead try to predict the protein structure from the amino-acid sequence alone. Though successful applications have been rare for many years, this field has recently received new attention through co-evolution based methods that use statistical approaches to detect significantly co-mutated residues from a large batch of homologous sequences [197, 198, 199, 200]. These co-mutated residues can then be employed to derive distance constraints which can be used to guide the protein folding process.

The other end of the spectrum is marked by homology modeling methods, for example the widely used MODELLER [201, 202]. Homology modeling relies on the availability of sequence and structural data of closely related, homologous, protein structures, so-called templates, to model an unknown target structure. The assumption is that, for proteins with the same or a similar function, the overall protein conformation should not be altered significantly, despite the introduction of, e.g., point-mutations. This assumption is for example justified by the structural classification schemes described in Section 3.2, where structurally and functionally related structures are classified into the same domain category despite potential differences in the amino-acid sequences.

However, structural modeling of proteins is not only required for proteins of unknown structure or with differences in amino-acid sequence. The docking methods presented in the previous section already demonstrate that structural changes, especially conformational adaptions, have to be considered when predicting binding modes between proteins and ligands or other proteins. Another field where conformational changes are important is the study of protein dynamics. For example, trajectories (the movements of one or several objects or atoms trough space over time) obtained from molecular dynamics (MD) simulations [203] may reveal detailed insights of the whole binding process that can not be derived from the snapshot that is given by a docking result. However, all-atom MD simulations of a whole protein, due to their implementation of Newton's laws of motion, often require several days, and are hence mostly inapplicable in docking studies, especially in a high-throughput scenario.

A method suitable for modeling whole-protein dynamics is the so-called normal mode analysis. This method is capable of determining the collective, energetically favorable, motions of a given protein conformation w.r.t. an underlying potential energy. The applicability of normal mode analysis in the context of protein-small molecule docking is in the focus of the study presented in Part ii and will hence be discussed there in detail.

In this section, we describe two backbone-dependent methods used throughout this thesis to predict side-chain rotamers in protein-small molecule and protein-protein docking.

3.4.1 *SCWRL*

SCWRL (Side-Chains With a Rotamer Library) [204] is probably the most prominent approach to model protein side-chain conformations. Existing side chains can also be

replaced by different amino acids, making this tool suitable for homology modeling purposes.

SCWRL relies on Dunbrack's popular backbone-dependent rotamer library (BBDep [205], for example also used in RosettaDock, see 3.3.4) which contains a small number of different side-chain conformations per amino acid. These side-chain conformations are associated with specific backbone-conformations for which they are energetically favorable.

Initially, a protein model is created from the input coordinates by assigning the BBDEP rotamers, furthermore the self-energies of each rotamer and pairwise energies of rotamer pairs are calculated. Both energy terms comprise hydrogen bonding as well as attractive and repulsive vdW terms; the self-energy term in addition accounts for the rotamer probabilities of the respective side chain w.r.t. to the residue's backbone conformation. Subsequently, high-energy rotamers are removed, disulfide bonds are determined and an interaction graph of the residues in the model is constructed (each residue is represented by a vertex, an edge between two vertices is added if at least one pair of rotamers, one from each corresponding residue, has a non-zero interaction energy).

The determination of the optimal side-chain conformations starts with two preprocessing steps: first, edges whose interaction energy deviates less than a predefined threshold from the sum of the interactions of the residues connected by that edge are removed and the self-interactions of the corresponding residues are adjusted accordingly. Furthermore, a dead-end elimination (DEE) algorithm using Goldstein's criterion [206] to efficiently reject rotamers that cannot be contained in the globally optimal solution is applied. The steps are iteratively repeated until no further removal is possible.

From the remaining rotamers, a graph is constructed to divide the side-chain optimization problem into independent sub-problems. First, the disconnected components (called clusters) are determined, subsequently the optimal side-chain conformations in each cluster are determined via tree decomposition (adapted from *Xu* [207]) and dynamic programming.

Finally, the globally optimal side chains from all independent sub-problems are collected and the global minimum energy conformation (GMEC) is returned.

3.4.2 *IRECS*

IRECS (Iterative REstriction of Conformational Space) [208] is an alternative method relying on simulated annealing to predict side-chain conformations of a protein structure. Its advantage over SCWRL lies in the fact that it is also able to generate structures with ensembles of side-chain rotamers, a feature especially useful in docking applications. This can be achieved by specifying a so-called rotamer density, i.e., an average number of rotamers per residue.

In a first step, IRECS discards all side-chain conformations present in the given input protein structure and rebuilds them from the standard parameters for bond lengths and angles as defined in the CHARMM force field [25, 209]. The corresponding side-chain rotamers for each residue are then sampled w.r.t. to the BBDEP. Initially, each rotamer receives a uniformly distributed probability that describes the influence of this rotamer on the other rotamers in the ensemble. In addition, effective energies are assigned to each rotamer taking into account all interactions of the rotamer with

all other rotamers in the ensemble, weighted by the probability of the respective rotamer.

In each iteration, IRECS determines the side chain with the largest range of effective energies according to the ROTA scoring function [210] and removes the rotamer with the worst energy. Subsequently the probabilities of the rotamers of this side chain as well as the effective energies are updated. Hence, the number of active rotamers is iteratively reduced.

If no more rotamers can be removed, either because there is no side chain with more than one rotamer left or because the target rotamer density has been reached, the algorithm stops and generates a pdb file with the structural ensemble.

3.5 PROTEIN STRUCTURE AND INTERACTION DATABASES

In the previous sections we have seen how structural information on proteins and other molecules can be obtained from experiments or computational modeling. Many different databases exist that store and provide information and data on protein structures, conformational ensembles and/or interactions between them. In the following, we present a selection of databases that are either used as data source in this thesis or are otherwise related to the work therein.

3.5.1 *Protein Data Bank*

The major source of protein structural data is the Protein Data Bank (PDB) [211] whose origin dates back to the 1970's [212]. According to the holdings report of July 24, 2014 [213] it contained a total of 101,948 searchable structural entries, 94,415 (92.61%) of which are proteins. Of these, 84,441 were determined using X-ray crystallography (see Subsection 3.1.1), 9,262 by NMR spectroscopy (see Subsection 3.1.2) and 565 by electron microscopy, 61 by hybrid and 86 using other methods (see Subsection 3.1.3).

Table 3.1 contains an excerpt of the content report generated on July 24, 2014 [214] that reveals the rapid growth of the database over the last 10 years.

Y	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	(2014)
A	5176	5355	6464	7194	6936	7362	7878	8039	8905	9607	(5439)
T	28769	34124	40588	47782	54718	62080	69958	77997	86902	96509	(101948)

Table 3.1: Content growth of the PDB over the last 10 years, showing the (Y)ear, the yearly (A)dded and the (T)otal number of structures. This table was generated on July 24, 2014, hence, values for 2014 are preliminary and thus written in brackets.

Besides the actual structures, the Protein Data Bank also provides a comprehensive tool set to find homologous proteins, visualize protein secondary structure on a sequence level, related literature and many more.

3.5.2 *3D Complex*

3D complex [83] is a database that contains topological information on protein complexes from the PDB. The information comprises contact information, as well as sequence and structural similarity measurements with the protein complexes rep-

resented as graph structures that describe the interactions between the individual components of the complex.

Polypeptide chains are represented as nodes, and interactions by edges. The nodes provide additional features: color and shape, with identical chains exhibiting the same color and shape, while homologous chains are differently colored but have the same shape.

The complexes are classified hierarchically according to their quaternary structure topology. The hierarchy contains 12 levels of increasingly strict definitions of similarity according to the following criteria: 1) the complex topology in terms of the number of components and their interactions, 2) the SCOP domain architecture (see Subsection 3.2.1) of each component, 3) the number of different chains per architecture, 4) amino acid sequence similarities of the components to other complexes, and 5) complex symmetry. The fourth criterion is further subdivided into sequence similarities from 20% to 100%, in steps of 10%.

The current version 2.0 features the SCOP hierarchy v1.73 (the follow-up version v1.75 has been available since June, 2009), indicating that the database is somewhat outdated both w.r.t. to the contained structures as well as the structural classification of the available complexes. As of July 24, 2014, the hierarchy contains the following number of entries at each level [215]:

Level	QS Topologies	QS Families	QS	QS30	QS70	QS100	All
Hierarchy 1	200	3785	3849	6441	9173	15270	30001
Hierarchy 2	191	3473	3530	5852	8328	14112	28266

Table 3.2: Number of entries contained in different hierarchy levels of the 3D complex database. Hierarchy 2 disregards all structures that are assigned errors by the manually curated PiQSi [216] database.

3.5.3 *Database of Macromolecular Motions*

The Database of Macromolecular Motions [217] tries to classify conformational changes of protein domains. Given two conformations of a protein structure, molecular modeling techniques are applied to interpolate between those conformations. However, the database is of limited use, because the set of proteins for which conformational changes are available is rather small and in addition depends on the used input conformations. Hence, the covered conformational space is restricted to the conformational difference between both proteins.

3.5.4 *Interactome 3D*

Interactome 3D [218] provides structural pairwise protein-protein interaction data, both experimentally determined and modeled, on a network basis. The database provides different services such as interaction annotation, interaction browsing in protein-protein interaction networks as well as visualization and download of structural information of interactions.

3.5.5 *STRING Database*

Similarly, the STRING database [219] provides information on physical as well as functional annotations of proteins, obtained from genomic context, high-throughput screening, and conserved coexpression experiments as well as from literature, together with a confidence score. As such, both STRING and Interactome 3D might provide a useful starting point to obtain binary interface information for the assembly of oligomeric assemblies from pairwise dockings.

3.6 METHODS OF EVALUATING THE ACCURACY OF STRUCTURAL MODELS

This thesis focuses on the structural modeling of binary protein-ligand complexes and oligomeric protein assemblies. The accuracy of the obtained models is performed using the methods presented in this section.

3.6.1 *Root-Mean-Square-Deviation*

A standard measure for the difference between two protein structures, small molecules, or generally speaking a mapping of two vectors of Cartesian points $S = (S_1, S_2, ..., S_n)$ and $T = (T_1, T_2, ..., T_n)$, each of cardinality n, is the so-called root-mean-square-deviation (RMSD), which is defined as follows [192]:

$$RMSD(S,T) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d(S_i, T_i)^2} \tag{3.1}$$

with d representing the Euclidean distance between two vectors S_i and T_i. In three dimensions, we have:

$$d(s,t) = \sqrt{(s_x - t_x)^2 + (s_y - t_y)^2 + (s_z - t_z)^2} \tag{3.2}$$

Typically, S and T comprise the heavy atoms of the structures to compare, but can, depending on the purpose only contain protein backbone atoms or, even more coarse-grained, only the C_α atoms of the structure.

When developing computational methods to predict protein-small molecule or protein-protein interactions, the RMSD is typically used to compare a computed with a given reference structure. In general, the lower the RMSD of the prediction to the reference, the more accurate the prediction is considered to be. However, sometimes, the restriction of the RMSD to a selected subset of point pairs is more meaningful than that between the whole structures. For example, the quality of the structural alignment and the RMSD between structures with different loop conformations may improve when the points belonging to the loop are excluded. On the other hand, excluding too many atoms may diminish the informative value of the obtained RMSD. Hence, a tradeoff between RMSD and alignment coverage must be found in such cases.

3.6.2 *Interaction RMSD*

The aforementioned standard RMSD provides a simple measure of the deviation between arbitrary coordinate sets involving rotational, translational, as well as conformational changes.

However, sometimes, for example when comparing or classifying interfaces between distant homologous proteins, the interest lies in the overall structural similarity of the interfaces and not in the local conformational differences between the individual homologous dimers.

To this end, *Aloy et al.* derived the interaction RMSD (iRMSD) measure [220]. Here, each monomer is represented by a set of seven standard points: the protein's centroid and six additional points which correspond to displacements by ± 5Å in the direction of x-, y-, and z-axes through the monomer centroid.

The iRMSD between two dimers P_1–Q_1 and P_2–Q_2 is computed as follows: first, the structurally similar proteins P_1 and P_2 as well as Q_1 and Q_2 are optimally superimposed to remove the impact of the proteins' different coordinate frames. Subsequently, the seven standard points for each protein are calculated. Consequently, each dimer consists of 14 standard points, seven for each protein, in the same orientation as the corresponding protein.

The interaction RMSD between the two dimers in standard point representation P_1^S–Q_1^S and P_2^S–Q_2^S is then computed using Equation 3.1 as $RMSD(P_1^S\text{–}Q_1^S, P_2^S\text{–}Q_2^S)$.

Based on iRMSD, we develop a measure called topology RMSD which is described in Subsection 11.2.9 and can be used to compare the topology of protein complexes.

3.7 STATISTICAL METHODS OF ASSESSING THE QUALITY OF A PREDICTION MODEL

In this section, we shortly introduce the measures and methods that are used for the scientific evaluation of the studies carried out in this thesis. For detailed information on the theoretical background of such methods as well as the field of statistical learning, the reader is referred to [221] which also provides the basis for the following sections.

3.7.1 *Cross-Validation*

When developing predictive methods in a supervised learning scenario [221], the methods learn (are trained) from known data, represented by a set of N observations $X = (X_1, ..., X_N)$ and corresponding outcomes $Y = (Y_1, ..., Y_N)$, where each observation $X_i = (X_{1i}, ..., X_{pi})$ is described by a set of p features. The aim of such methods is to generalize the learned knowledge to unseen data by learning an estimator function $\hat{f}$ that predicts Y_i given X_i. These methods typically require, besides the input features, external parameters, e.g., to select the properties of $\hat{f}$ (also called model) that minimizes the training error, which is given as:

$$\overline{err} = \frac{1}{N} L(Y, \hat{f}(X)) = \frac{1}{N} \sum_{i=1}^{N} L(Y_i, \hat{f}(X_i)) \tag{3.3}$$

where Y_i is the true outcome for the i-th observation, $\hat{f}(X_i)$ is the corresponding predicted outcome according to a feature vector X_i and an estimator $\hat{f}$ learned from the set of observations X. L is the so-called loss function which measures the error between true outcome and prediction. Depending on the underlying data, several choices for this loss function are possible, for example the Euclidean distance (cmp. Eq. 3.2) or RMSD (cmp. Eq. 3.1).

Once such an estimator $\hat{f}$ has been trained, the following problem arises: a realistic assessment of the model quality. Obviously, it is a bad idea to use the training error as a measure of accuracy, because it is always possible to achieve a perfect fit: if the feature space is large enough, i.e., $p \geq N$, the model can fit each observation independently and thus correctly predict the whole training set. However, when predicting unseen data X' with known outcome Y', such models can be expected to perform poorly, i.e., they have a high expected prediction error, given as:

$$Err = E[L(Y', \hat{f}(X'))] \tag{3.4}$$

This effect is called over-training, a situation that should be avoided. However, one often encounters data-scarce situations, where an additional set on which the prediction accuracy can be evaluated is not available. For such settings, methods of sample re-use exist, one of them being the so called cross-validation [221].

Here, the available data is first randomly split into K folds $X^1, ..., X^K$ and corresponding outcomes $Y^1, ..., Y^K$. Each of of the K folds is then predicted in turn, the k-th fold by a model $\hat{f}^{-k}$ trained on the remaining $k-1$ folds. The cross-validation error is then given as:

$$CV = \frac{1}{K} \sum_{k=1}^{K} L(Y_k, \hat{f}^{-k}(X_k)) \tag{3.5}$$

A question that often arises is how many folds should be used. For $K = N$, i.e., when each fold consists of a single observation, we speak of leave-one cross-validation (LOOCV). However, such an approach often introduces an artificially low cross-validation error. For smaller K and thus smaller training sets, the cross-validation error is considered to be more realistic and often 5-fold or 10-fold cross-validation runs are assumed to be good choices [221].

3.7.2 *Receiver Operator Characteristic (ROC) Curve*

The receiver operator characteristic (ROC) curve can be used to demonstrate the discriminative power of a prediction method on a set X of N observations given a binary ground truth label y_i for each element $x_i \in X$ [221]. To this end, the true positive rate (TPR) is plotted against the false positive rate (FPR) for a varying discrimination threshold. In a ranked scenario, as for example in docking scenarios where the generated poses are typically ranked according to the score obtained from a scoring function, this threshold typically corresponds to the best n solutions for varying n.

Let $Y = (y_1, y_2, ..., y_N)$ be a vector of ground truth labels that correspond to the sorted best N elements of X starting with the best element. Then, the number of true positives TP_n for the first n solutions (where n is a selectable threshold) is given as:

$$TP_n = \sum_{i=1}^{n} y_i \tag{3.6}$$

Analogously, the false positives FP_n, the number of wrong solutions among the first n predictions, are given by:

$$FP_n = \sum_{i=1}^{n} 1 - y_i \tag{3.7}$$

The true negatives TN_n comprise the wrong solutions that are beyond rank n and hence correctly rejected:

$$TN_n = \sum_{i=n+1}^{N} 1 - y_i \tag{3.8}$$

Analogously, false negatives FN_n are correct solutions which are not considered because they are beyond rank n:

$$FN_n = \sum_{i=n+1}^{N} y_i \tag{3.9}$$

From these values, the true positive and false positive rates for the first n solutions, TPR_n and FPR_n respectively, can be calculated:

$$TPR_n = \frac{TP_n}{TP_n + FN_n} \tag{3.10}$$

and

$$FPR_n = \frac{FP_n}{FP_n + TN_n} \tag{3.11}$$

The corresponding values TPR_n can then be plotted against FPR_n.

3.7.3 *Area Under the ROC Curve (ROC AUC)*

While a ROC curve provides a graphical interpretation of the discriminative power of a prediction method or model, one is often interested in a single value reflecting the general behavior of the curve. This is possible by computing the area under the ROC curve w.r.t. to set of N ROC points $\{(x_1, y_1), ..., (x_N, y_N)\}$, sorted by increasing FPR, through linear interpolation between subsequent points:

$$AUC = \frac{1}{2} \sum_{i=1}^{N-1} (x_{i+1} - x_i)(y_{i+1} - y_i) \tag{3.12}$$

If $FPR_N = 0$, i.e., all solutions are correct, the AUC is set to 1. Analogously, if $TPR_N = 0$, AUC is set to 0.

Part II

MODELING BACKBONE FLEXIBILITY IN PROTEIN-LIGAND DOCKING

4 INTRODUCTION

The molecular basis of diseases resides in processes pertaining to the function and interaction of proteins and other biological macromolecules. The aim of computer-aided drug design is to develop drugs that influence the activity of these molecules and lessen or neutralize the effects of functional disorders. Such drugs can, for example, stimulate signal-transduction pathways, inhibit a protein's catalytic function, modulate protein-protein interactions, or change the rate at which a gene is transcribed [222, 223, 224, 225].

However, the accurate and fast prediction of promising candidate molecules and the correct protein-ligand complex conformation – preferably combined with an accurate estimation of the respective binding affinities – is still a largely unsolved problem. To be useful in a high-throughput virtual screening, i.e., the *in silico* testing of chemical compounds from a large virtual library for their suitability as potential drugs w.r.t. to a biological target [226], docking must be computationally very efficient but still produce reliable results [227, 228]. However, fast and accurate scoring functions that efficiently guide the search for the final protein-ligand complex for arbitrary protein-ligand complexes are currently not available, and are not likely to be discovered in the near future. To make matters worse, the energy landscape of the protein changes through the binding of a ligand: on the one hand, the ligand itself changes the landscape by establishing interactions with the protein. On the other hand, both protein and ligand are not rigid bodies but are able to undergo substantial conformational changes [229, 230, 231]. To tackle these problems in a reasonable computation time, docking algorithms are forced to apply simplifications which, however, typically reduce the accuracy of the predictions.

In this context, the treatment of ligand- and protein flexibility is usually different. While ligands are chemically much more diverse than proteins, they contain significantly fewer degrees of freedom. Thus, approaches to handling ligand flexibility use different global optimization schemes, such as Monte Carlo methods, e.g., ICM [232] or LigandFit [233], genetic algorithms such as AutoDock [234] or GOLD [174], incremental construction like FlexX [31], or grid-based methods like Glide [235, 236]. Protein flexibility, in contrast, requires a different strategy, since here the number of degrees of freedom becomes forbiddingly large, even for small proteins. In a first step, the protein's movements are decomposed into side-chain re-arrangement and backbone movement. Side-chain conformers can be described suitably by a discrete set of rotamers [205], but optimizing their arrangement via exhaustive sampling incurs the risk of combinatorial runtime explosion. Most of the earliest methods able to capture small induced fit effects [17] thus concentrated on locally sampling the side-chain conformations of the active site during the docking process [237, 238, 239, 240, 241, 242, 190, 243, 244]. Others used pre-generated protein ensembles and, in some cases, also considered different backbone conformations [245, 169, 246, 177, 247, 248, 249, 250]. Such molecular structure ensembles are assumed to be representative for the conformational space the protein can explore, but

they usually fail when large-scale backbone movements are involved, as for example observed in HIV-1 protease [251] or aldose reductase [252].

To describe such conformational alternatives of proteins, different techniques can be applied. One possibility is to carry out a principal component analysis of movements, determined by MD simulations [253, 254]. These so-called essential modes have been shown to improve the docking performance significantly [255, 256]. But because MD simulations are computationally expensive, approximating the global dynamics of proteins by normal mode analysis (NMA) on the basis of coarse-grained elastic network models (ENM) [257], i.e., models considering only a subset of atoms of the protein structure, has become increasingly popular over the last years. Due to its ability to reproduce the collective (or global) motions of proteins, i.e., motions involving a large part of the protein, without significant loss of accuracy [258, 259, 260, 261, 262, 263, 264, 265, 266], NMA has been applied to many different problems, for example protein domain decomposition [259, 267, 268], guiding MD simulations along normal modes [269, 270], or fitting proteins into electron density maps obtained from cryo-EM or X-ray crystallography [271, 272, 273, 274].

The benefit of coarse-graining compared to the use of an all-atom representation is that a large number of the normal modes corresponding to local movements, so-called non-collective or local normal modes, which involve only a small number of atoms, in particular those not incorporated in the coarse-grained representation, do not need to be computed. Consequently, the number of modes that need to be considered for conformational sampling is substantially reduced. In this context, several levels of coarse-graining exist: besides the convenient strategy to use the C_α trace of the backbone, sets of atoms or residues can be combined into blocks [275, 276, 277, 278, 279]. Other approaches explore subsets of protein components [280, 281, 282], use additional grains that represent side-chain centroids [283], or employ a mixed coarse-graining with a higher resolution in important protein regions [284]. So far, several studies have established the ability of elastic network models to also predict conformational changes during protein-protein docking [285, 286, 287, 288]. Moreover, a recent paper has proposed a sophisticated method to sample protein conformations using an ENM [289].

However, in protein-ligand docking, the binding interfaces are typically much smaller than in protein-protein docking and there are theoretical arguments for the assumption that normal modes may not be suitable to model backbone movements involved in ligand binding: on the one hand, the primary purpose of normal modes is to describe large-scale collective motions of a system and thus they may not be well-suited to model the more local movements related to ligand binding using only a small number of normal modes. On the other hand, differences between several conformations of the same protein which are interpreted to be due to protein motion may, in fact, be the result of uncertainties in the coordinates of experimentally determined structures. Thus it may not be adequate to use normal modes to interpret such differences. However, despite these assumptions, normal modes have already been successfully applied in select cases of protein-ligand docking as, for example, in two studies using normal modes from heavy-atom and all-atom ENM [290, 291].

These studies indicate that backbone conformational changes observed in protein-small molecule binding may, in some cases, only be modeled when using non-collective modes accounting for local movements of small parts of the protein backbone. The question arises how this observation translates to coarse-grained ENMs using C_α atoms, where the number of modes that have to be considered is drasti-

cally reduced, and how suitable they are for such problems where alternative protein conformations are needed to improve docking results, for example when only the conformation of the protein in the unbound (apo) state is available but the protein may undergo conformational changes upon ligand binding. In this context, *May et al.* have shown in a cross-docking study with six different CDK2 inhibitors [292] that NMA can significantly improve docking results while *Cavasotto et al.* have reported similar results in their study with cAPK [293].

In this study, we thus investigate on a larger scale how suitable binding-pocket restricted normal modes from a C_α-ENM are for protein-ligand docking, with a focus on high-throughput applications. By establishing a best-case scenario for conformational sampling on a diverse data set derived from the Astex Diverse [294] and Non-Native [295] Set, we evaluate how the number of modes used to reproduce a ligand-bound (holo) conformation from its respective unbound (apo) state influences the docking accuracy. The corresponding ligands are docked into the reproduced holo structures using AutoDock [234], GOLD [174], and FlexX [31].

5

MATERIALS AND METHODS

5.1 NORMAL MODE ANALYSIS FOR ELASTIC NETWORK MODELS

The aim of performing a Normal Mode Analysis (NMA) on biological macromolecules [296, 297] is to determine the global and energetically most favorable motions of a system close to its energetic minimum. Here, the assumption is that a protein in an energetically stable conformation oscillates harmonically around this equilibrium. A normal mode represents a collective motion within such a system at a certain oscillation frequency. Each mode has its own unique frequency which is proportional to the energy required w.r.t. the underlying potential to perform a unit length motion along the mode.

To determine these modes, we use an Elastic Network Model (ENM) [261], an extension of the Gaussian Network Model (GNM) [258, 259, 260, 262] that accounts for the anisotropy of motions of a system's components in Cartesian space. Here, an artificial harmonic potential V is constructed around an assumed minimum energy protein conformation $\mathbf{R}^0$ consisting of N point masses (C_α atoms in our case), such that $\mathbf{R}^0$ becomes the minimum conformation of V (Fig. 5.1). For a conformation

$$\mathbf{R} := \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_N \end{bmatrix} \in \mathbb{R}^{3N} \tag{5.1}$$

where $\mathbf{R}_i \in \mathbb{R}^3$ is a column vector representing the coordinates of the i-th point mass, $i \in \{1, ..., N\}$, the potential energy in this elastic network model is given by:

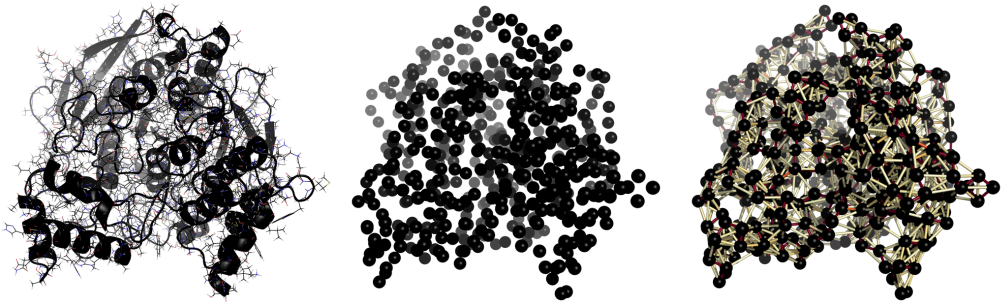
$$V(\mathbf{R}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=i+1}^{N} k\left(\left|\mathbf{R}_i^0 - \mathbf{R}_j^0\right|\right) \left(\left|\mathbf{R}_i - \mathbf{R}_j\right| - \left|\mathbf{R}_i^0 - \mathbf{R}_j^0\right|\right)^2 \tag{5.2}$$

The quadratic term describes a spring between two point masses i and j that is relaxed in $\mathbf{R}^0$. It can be easily seen that $V(\mathbf{R}) = 0$ when $\mathbf{R} = \mathbf{R}^0$ and $V(\mathbf{R}) > 0$ anywhere else. Furthermore, to scale the influence of each spring according to the distance of the participating point masses in the minimum conformation, each spring is assigned a spring constant by a function $k(d)$ which decreases with the distance d between the involved point masses. In this way, close spatial neighbors can be made to contribute more strongly to the potential than remote atoms.

To apply NMA to an ENM, one assumes that $V(\mathbf{R})$ can be approximated by quadratic Taylor Expansion around $\mathbf{R}^0$:

$$V(\mathbf{R}) \approx \frac{1}{2} \left(\mathbf{R} - \mathbf{R}^0\right)^{\mathrm{T}} \mathbf{H} \left(\mathbf{R} - \mathbf{R}^0\right) \tag{5.3}$$

where $\mathbf{H} := H_V(\mathbf{R} = \mathbf{R}^0)$ is the Hessian matrix containing all second partial derivatives of $V(\mathbf{R})$ evaluated at $\mathbf{R} = \mathbf{R}^0$ (a detailed derivation of the Hessian in an ENM



(a) (b) (c)

Figure 5.1: Exemplary construction of a C_α-atom ENM: (a) from the all-atom structure (pdb code 1gpk), (b) the C_α trace (one sphere corresponds to one C_α atom) is extracted, and (c) the springs with corresponding strengths between the atoms are calculated (sticks: springs, stick color: spring strength from weak (ivory) to strong (red); for illustrative purposes, only a subset of stronger springs is shown).

can, for example, be found in *Atilgan et al.* [261]). $H_V(\mathbf{R})$ can be interpreted as a block matrix of 3×3 submatrices of the form:

$$H_V(\mathbf{R}) = \begin{bmatrix} H_{11}(\mathbf{R}) & \dots & H_{1N}(\mathbf{R}) \\ \vdots & \ddots & \vdots \\ H_{N1}(\mathbf{R}) & \dots & H_{NN}(\mathbf{R}) \end{bmatrix} \tag{5.4}$$

where each submatrix $H_{ij}(\mathbf{R}), 1 \leq i, j \leq N$ contains the second partial derivatives of $V(\mathbf{R})$ w.r.t. the x-, y-, and z-components of point masses i and j:

$$H_{ij}(\mathbf{R}) = \begin{bmatrix} \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{ix} \partial \mathbf{R}_{jx}} & \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{ix} \partial \mathbf{R}_{jy}} & \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{ix} \partial \mathbf{R}_{jz}} \\ \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{iy} \partial \mathbf{R}_{jx}} & \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{iy} \partial \mathbf{R}_{jy}} & \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{iy} \partial \mathbf{R}_{jz}} \\ \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{iz} \partial \mathbf{R}_{jx}} & \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{iz} \partial \mathbf{R}_{jy}} & \frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{iz} \partial \mathbf{R}_{jz}} \end{bmatrix} \tag{5.5}$$

For example, according to the above elastic network model, the second mixed-derivative w.r.t. to the x-components of i and j, $i \neq j$, i.e., the upper left entry of $H_{ij}(\mathbf{R})$ for $\mathbf{R} = \mathbf{R}^0$, is given by:

$$\left.\frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{ix} \partial \mathbf{R}_{jx}}\right|_{\mathbf{R}^0} = -k\left(\left|\mathbf{R}_j^0 - \mathbf{R}_i^0\right|\right) \left.\frac{\left(\mathbf{R}_{jx} - \mathbf{R}_{ix}\right)\left(\mathbf{R}_{jx} - \mathbf{R}_{ix}\right)}{\left(\left|\mathbf{R}_j - \mathbf{R}_i\right|\right)^2}\right|_{\mathbf{R}^0} \tag{5.6}$$

The other entries of $H_{ij}(\mathbf{R})$ for $\mathbf{R} = \mathbf{R}^0$ are calculated analogously. Likewise, the second order derivative of point mass i w.r.t. its x-component is given as:

$$\left.\frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{ix}{}^2}\right|_{\mathbf{R}^0} = \sum_{j=1, j\neq i}^{N} k\left(\left|\mathbf{R}_j^0 - \mathbf{R}_i^0\right|\right) \left.\frac{\left(\mathbf{R}_{jx} - \mathbf{R}_{ix}\right)\left(\mathbf{R}_{jx} - \mathbf{R}_{ix}\right)}{\left(\left|\mathbf{R}_j - \mathbf{R}_i\right|\right)^2}\right|_{\mathbf{R}^0} = -\sum_{j=1, j\neq i}^{N} \left.\frac{\partial^2 V(\mathbf{R})}{\partial \mathbf{R}_{ix} \partial \mathbf{R}_{jx}}\right|_{\mathbf{R}^0} \tag{5.7}$$

By construction, $\mathbf{H}$ is positive semi-definite and hence has real eigenvectors and all eigenvalues are either positive or zero. The normal modes are defined as the eigenvectors $\mathbf{U}$ of $\mathbf{H}$, which, together with the eigenvalues $\mathbf{\Lambda}$, are obtained by carrying out an eigenvalue decomposition on the Hessian:

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}} \tag{5.8}$$

Usually, the normal modes are sorted in ascending order w.r.t. their eigenvalues as these correspond to the energetic cost required to perform a unit length movement along a mode in the energetic model of the ENM: assume, we are given a movement along a normal mode $\mathbf{u}_i$ with $|\mathbf{u}_i| = 1$ and eigenvalue λ_i. The obtained conformational change is then given by:

$$\mathbf{R} = \mathbf{R}^0 + \mathbf{u}_i \tag{5.9}$$

Insertion into Eq. 5.3 yields:

$$V(\mathbf{R}) = \frac{1}{2}\mathbf{u}_i^{\mathrm{T}} \cdot \mathbf{H} \cdot \mathbf{u}_i = \frac{1}{2}\mathbf{u}_i^{\mathrm{T}} \cdot \lambda_i \cdot \mathbf{u}_i = \frac{1}{2}\lambda_i \cdot \mathbf{u}_i^{\mathrm{T}} \cdot \mathbf{u}_i = \frac{1}{2}\lambda_i \tag{5.10}$$

Modes that require little energy (i.e., a unit-length movement along such modes imposes little stress on the spring network) are associated with collective motions that involve a large part of the underlying system, in contrast, modes requiring large amounts of energy correspond to non-collective, local motions. The first six modes have zero eigenvalues and correspond to the translational and rotational degrees of freedom of the whole system – in the case of an elastic network model, these obviously require no energy.

5.2 EXTRACTING BINDING POCKET NORMAL MODES

In protein-ligand docking we are particularly interested in the conformational changes of the binding site. But by calculating the normal modes for the whole protein, we will obtain many normal modes that are associated with collective movements elsewhere in the protein. Hence, to restrict the normal mode set to those modes that are collective for the binding pocket and thus relevant for protein-ligand docking, we use an approach described in *Zheng et al.* [280] and *Ming et al.* [281]: we divide the protein into two components, the binding pocket and the remaining protein, by rearranging $\mathbf{H}$ such that we obtain four submatrices:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\mathrm{pp}} & \mathbf{H}_{\mathrm{pe}} \\ \mathbf{H}_{\mathrm{ep}} & \mathbf{H}_{\mathrm{ee}} \end{bmatrix} \tag{5.11}$$

$\mathbf{H}_{\mathrm{pp}}$ and $\mathbf{H}_{\mathrm{ee}}$ contain the couplings within binding pocket and environmental protein, respectively, $\mathbf{H}_{\mathrm{pe}}$ the stress imposed on the environment by changes in the pocket and for $\mathbf{H}_{\mathrm{ep}}$ vice versa. Let $\mathbf{r}_{\mathrm{p}}$ and $\mathbf{r}_{\mathrm{e}}$ be the conformational changes in the pocket and the environment, insertion into Eq. 5.3 yields:

$$\begin{aligned} V\left(\begin{bmatrix} \mathbf{r}_{\mathrm{p}} \\ \mathbf{r}_{\mathrm{e}} \end{bmatrix}\right) &= \frac{1}{2}\begin{bmatrix} \mathbf{r}_{\mathrm{p}} \\ \mathbf{r}_{\mathrm{e}} \end{bmatrix}^T \begin{bmatrix} \mathbf{H}_{\mathrm{pp}} & \mathbf{H}_{\mathrm{pe}} \\ \mathbf{H}_{\mathrm{ep}} & \mathbf{H}_{\mathrm{ee}} \end{bmatrix} \begin{bmatrix} \mathbf{r}_{\mathrm{p}} \\ \mathbf{r}_{\mathrm{e}} \end{bmatrix} \\ &= \underbrace{\frac{1}{2}\mathbf{r}_{\mathrm{p}}{}^T\mathbf{H}_{\mathrm{pp}}\mathbf{r}_{\mathrm{p}} + \frac{1}{2}\mathbf{r}_{\mathrm{p}}{}^T\mathbf{H}_{\mathrm{pe}}\mathbf{r}_{\mathrm{e}}}_{V_{\mathbf{r}_{\mathrm{e}}}(\mathbf{r}_{\mathrm{p}})} + \underbrace{\frac{1}{2}\mathbf{r}_{\mathrm{e}}{}^T\mathbf{H}_{\mathrm{ep}}\mathbf{r}_{\mathrm{p}} + \frac{1}{2}\mathbf{r}_{\mathrm{e}}{}^T\mathbf{H}_{\mathrm{ee}}\mathbf{r}_{\mathrm{e}}}_{V_{\mathbf{r}_{\mathrm{p}}}(\mathbf{r}_{\mathrm{e}})} \end{aligned} \tag{5.12}$$

where $V_{\mathbf{r}_{\mathrm{e}}}(\mathbf{r}_{\mathrm{p}})$ and $V_{\mathbf{r}_{\mathrm{p}}}(\mathbf{r}_{\mathrm{e}})$ denote the contributions of $\mathbf{r}_{\mathrm{p}}$ and $\mathbf{r}_{\mathrm{e}}$ to the overall potential energy, respectively.

We can now assume that upon a given conformational change $\mathbf{r}_p$ in the binding pocket, the environmental residues perform an adaptive movement $\mathbf{r}_e$ that minimizes the total energy in the elastic network, i.e. $V_{\mathbf{r}_p}(\mathbf{r}_e) = 0$. $\mathbf{r}_e$ is given by (the final equation is also presented in [280]):

$$\begin{aligned} & V_{\mathbf{r}_p}(\mathbf{r}_e) &&= 0 \\ \Leftrightarrow\ & \tfrac{1}{2}\mathbf{r}_e{}^T\mathbf{H}_{ep}\mathbf{r}_p + \tfrac{1}{2}\mathbf{r}_e{}^T\mathbf{H}_{ee}\mathbf{r}_e &&= 0 \\ \Leftrightarrow\ & \tfrac{1}{2}\mathbf{r}_e{}^T\mathbf{H}_{ee}\mathbf{r}_e &&= -\tfrac{1}{2}\mathbf{r}_e{}^T\mathbf{H}_{ep}\mathbf{r}_p \\ \Leftrightarrow\ & \mathbf{H}_{ee}\mathbf{r}_e &&= -\mathbf{H}_{ep}\mathbf{r}_p \\ \Leftrightarrow\ & \mathbf{r}_e &&= -\mathbf{H}_{ee}^{-1}\mathbf{H}_{ep}\mathbf{r}_p \end{aligned} \tag{5.13}$$

Substitution of $\mathbf{r}_e$ in $V_{\mathbf{r}_e}(\mathbf{r}_p)$ by the right-hand side of Eq. 5.13 yields:

$$\begin{aligned} V_{\mathbf{r}_e}(\mathbf{r}_p) &= \tfrac{1}{2}\mathbf{r}_p{}^T\mathbf{H}_{pp}\mathbf{r}_p + \tfrac{1}{2}\mathbf{r}_p{}^T\mathbf{H}_{pe}\mathbf{r}_e \\ &\overset{\text{Eq. 5.13}}{=} \tfrac{1}{2}\mathbf{r}_p{}^T\mathbf{H}_{pp}\mathbf{r}_p + \tfrac{1}{2}\mathbf{r}_p{}^T\mathbf{H}_{pe}\left(-\mathbf{H}_{ee}^{-1}\mathbf{H}_{ep}\mathbf{r}_p\right) \\ &= \tfrac{1}{2}\mathbf{r}_p{}^T\mathbf{H}_{pp}\mathbf{r}_p - \tfrac{1}{2}\mathbf{r}_p{}^T\mathbf{H}_{pe}\mathbf{H}_{ee}^{-1}\mathbf{H}_{ep}\mathbf{r}_p \\ &= \tfrac{1}{2}\mathbf{r}_p{}^T\left(\mathbf{H}_{pp} - \mathbf{H}_{pe}\mathbf{H}_{ee}^{-1}\mathbf{H}_{ep}\right)\mathbf{r}_p \\ &= \tfrac{1}{2}\mathbf{r}_p{}^T\mathbf{H}_{\mathit{eff}}\mathbf{r}_p \end{aligned} \tag{5.14}$$

Consequently, under the assumption $V_{\mathbf{r}_p}(\mathbf{r}_e) = 0$, the effective Hessian for our binding pocket is defined as [280, 281]:

$$\mathbf{H}_{\mathit{eff}} = \mathbf{H}_{pp} - \mathbf{H}_{pe}\mathbf{H}_{ee}^{-1}\mathbf{H}_{ep} \tag{5.15}$$

The normal modes $\mathbf{U}_{\mathit{eff}}$ for the binding pocket can then be obtained from $\mathbf{H}_{\mathit{eff}}$ using Eq. 5.8. The resulting modes provide an orthonormal basis set which is again sorted in ascending order according to the modes' eigenvalues and thus to their degree of collectivity. Hence, the first modes describe the most collective motions within the subsystem represented by $\mathbf{H}_{\mathit{eff}}$. The adaptive modes for the remaining protein can be calculated from $\mathbf{U}_{\mathit{eff}}$ and Eq. 5.13.

In the following, we describe how we establish a best-case scenario to demonstrate how well the binding-pocket normal $\mathbf{U}_{\mathit{eff}}$ modes can be used in docking applications to model the induced-fit backbone conformational changes in the binding pocket upon ligand binding: we establish a diverse benchmark set of pairs of apo/holo conformation which we use to generate intermediate structures that optimally reproduce the holo conformation w.r.t. increasing subsets of the most-collective modes of $\mathbf{U}_{\mathit{eff}}$ obtained from the apo conformation. The respective ligands from the holo conformation are then docked into these intermediate structures to assess the improvement in docking performance w.r.t. to the size of the normal-mode subset used to generate the intermediate conformation.

5.3 DATA SET

The data set we used is derived from the Astex Diverse [294] and the Astex Non-Native Set [295]. The Astex Diverse Set comprises 85 diverse protein crystal structures, bound to drug-like ligands, with a resolution of less than 2.5Å. The structures have been automatically and manually checked for structural problems, i.e., clashes,

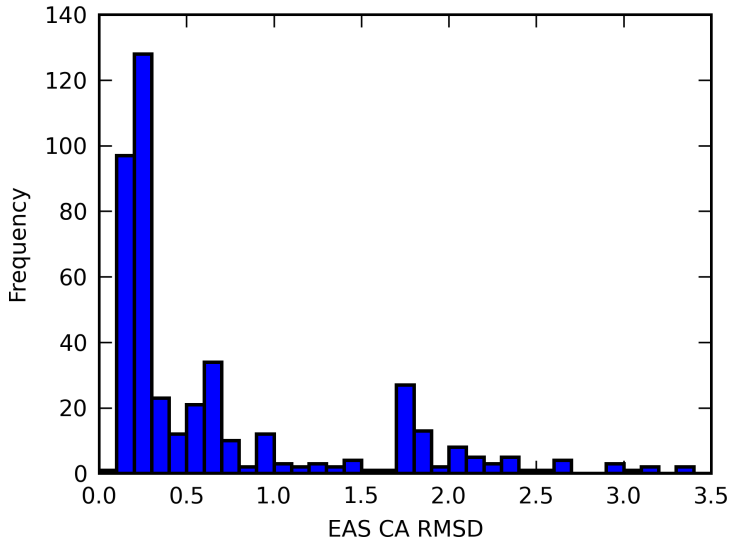


Figure 5.2: Distribution of C_α-RMSDs [Å] in the extended active site.

interactions with symmetry units, and dubious ligand binding. The Non-Native Set was set up analogously and consists of 1112 non-native (alternative) protein conformations, apo structures as well as holo conformations with corresponding ligand, for 65 of the reference structures contained in the Diverse Set. The binding pockets of the non-native structures are unmutated w.r.t. those of the reference structure and have been superimposed onto the reference pocket for a better comparability of the results of cross-docking studies.

In our study, the aim is to address the question whether and if so, how well, reduced sets of normal modes can model conformational changes observed during binding of small molecules. We thus chose only reference structures with at least one apo structure in the Non-Native Set to avoid bias of the protein backbone conformation towards any ligand.

For each of the 29 remaining reference structures, we determined the residues that are involved in substrate binding in any of the corresponding holo structures and are thus associated with possible conformational changes in the protein binding site: we first independently defined the binding pocket for each holo structure as consisting of those residues that have at least one heavy atom within a distance of 6.0Å to a heavy atom of the corresponding ligand. These pockets were then aligned and merged into one extended active site (EAS) that comprises for each conformation all the residues that are in contact with any of the ligands. The respective residues make up the residues contributing to $\mathbf{H}_{\mathit{eff}}$; the remaining residues form the environment which is assumed to perform an adaptive movement that minimizes the global energy required for the conformational changes in the binding site. For each of the 29 reference structures we only kept those apo/holo pairs that have no mismatches or indels in the EAS between apo and holo structure. The resulting data set consisted of 283 apo/holo pairs from 20 reference structures, with 260 having a C_α-RMSD below 0.5Å in the EAS.

To gain more data on structural differences exceeding a C_α-RMSD of 0.5Å while keeping the effect of structural mutations on the protein dynamics small, we decided to also incorporate apo/holo pairs with a C_α-RMSD of at least 0.5Å and at most five mismatches which, however, must not occur within the EAS. In this way, we

augmented our data set by 150 additional apo/holo pairs while ensuring that the docking results do not suffer from mutations in the binding pocket. The complete data set contains 433 apo/holo pairs from 21 different reference structures. For each apo/holo pair, the respective apo structure was optimally superimposed onto the corresponding holo conformation w.r.t. the EAS C_α atoms. A distribution of the C_α-RMSDs can be found in Figure 5.2. Each structure in this data set was converted into pdb format, missing atoms and side chains were added with BALL [298]. The corresponding ligands were converted into mol2 format using OpenBabel [299].

5.4 ESTABLISHING A BEST-CASE SCENARIO

From the apo/holo pairs of the data set derived in the previous section, we then generated intermediate structures that optimally reproduce the holo conformation. For each apo structure, we first calculated the effective modes $\mathbf{U}_{eff}$ for the residues contained in its EAS as described in the previous sections and established subsets of the first m (m =10%, 20%, ..., 100%) of these modes.

Let $\mathbf{S}$ be the matrix containing the first m modes,

$$\mathbf{S} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 & \dots & \mathbf{U}_m \end{bmatrix} \tag{5.16}$$

we then orthogonally projected the conformational difference between apo and holo binding pockets, $\mathbf{R}_\mathrm{A}$ and $\mathbf{R}_\mathrm{H}$ onto $\mathbf{S}$:

$$\mathbf{P} = \mathbf{S}^\mathrm{T}(\mathbf{R}_\mathrm{H} - \mathbf{R}_\mathrm{A}) \tag{5.17}$$

From this projection, we can then obtain the amplitudes $\mathbf{A}$ for the modes that minimize the distance between this projection and the space $\mathbf{S}$:

$$\mathbf{A} = (\mathbf{S}^\mathrm{T}\mathbf{S})^{-1}\mathbf{P} \tag{5.18}$$

We can then generate an approximate conformation $\mathbf{R}^*_\mathrm{H}$ of the holo binding pocket from that of the apo structure w.r.t. the underlying space $\mathbf{S}$ by:

$$\mathbf{R}^*_\mathrm{H} = \mathbf{R}_\mathrm{A} + \mathbf{S}\mathbf{A} \tag{5.19}$$

Due to the orthogonal projection, the distance between the C_α trace of the holo conformation, $\mathbf{R}_\mathrm{H}$, and $\mathbf{R}^*_\mathrm{H}$ is minimal w.r.t. $\mathbf{S}$. Thus, a conformational sampling in the same subspace can never yield a conformation that is closer to the original holo structure than our intermediate structure. Our intermediate structures can thus be considered as an upper bound for the accuracy achievable with conformational sampling algorithms w.r.t. the underlying normal mode subspace.

Applying the above procedure to each of the generated mode subsets leads to increasingly well reproduced holo C_α conformations. This makes it possible to investigate how the docking performance relates to the number of modes used to reconstruct the holo conformation and to estimate how many modes are needed to sufficiently reproduce the conformational change upon ligand binding.

To prepare these conformations for docking, the all-atom structures were reconstructed by translating the side chains and remaining backbone atoms according to the displacement of the corresponding C_α atom. We then relaxed the resulting structure for 0, 10, and 50 steps using the AMBER96 [300] force field and an L-BFGS

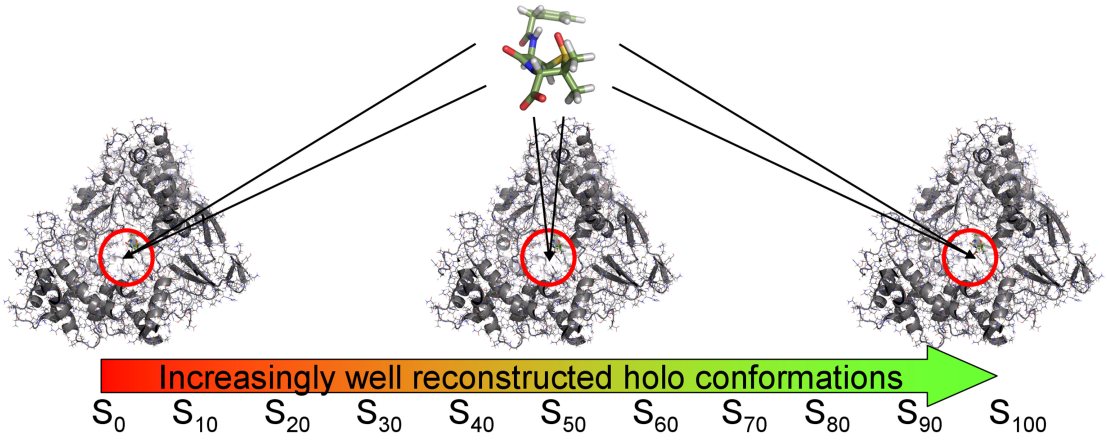


Figure 5.3: Schematic description of our best-case scenario (shown for acetylcholinesterase, pdb code 1gpk). The native ligands are sequentially docked into a set of increasingly well reconstructed holo conformations to assess how the docking performance depends on the number of modes used for the reconstruction.

minimizer [301, 302] to resolve possible steric clashes between side chains and/or backbone atoms while keeping C_α atoms fixed. In this context, when using all available modes ($m = 100\%$), we obtain a so-called 100% reconstructed holo conformation, i.e., the C_α trace of the holo conformation binding pocket is exactly reproduced while the remaining part of the protein (side chains, non-C_α backbone atoms) may show deviations from the original holo structure due to the reconstruction procedure. A validation of this reconstruction procedure can be found in Section A.2. With this procedure, we obtained 33 intermediate structures per apo/holo pair, in total. The full data set to be docked contained 15,304 protein conformations derived from 433 apo/holo pairs and the original holo structures (see Section A.1 for more details on the data set composition). A schematic representation of our best-case scenario is given in Figure 5.3.

5.5 DOCKING EXPERIMENTS

To analyze the quality of the structures w.r.t. docking, we performed two different docking rounds. In the first round, we investigated the capability of normal modes without considering the side-chain conformations. We established six different docking protocols, consisting of a standard and a soft docking setup for each of the docking programs AutoDock [234], GOLD [174], and FlexX [31]. The standard protocols used the default parameters of the respective docking program; for AutoDock and GOLD, the number of runs was set to 25 in both the standard and soft docking protocol. Furthermore, for the soft docking protocols, we adjusted the parameters to reduce the impact of steric clashes. In AutoDock, FE_coeff_vdW was reduced by a factor of 0.5 while in GOLD, start_vdw_linear_cutoff was set to 4 and the binding pocket residues were assigned a 2-4 vdW potential. In FlexX, MAX_OVERLAP_VOL and DOT_OVERLAP_VOL were increased by a factor of 1.5.

In the second round, side-chain flexibility was explicitly taken into account as it can have a significant impact on the docking performance. We therefore selected those apo/holo pairs from the first round for which the ligand could be successfully re-docked into the original holo structures (i.e., with a minimum pose RMSD below

2.0Å) with at least one of the above described docking protocols, but failed to do so for the 100% reconstructed holo conformation. We established three additional docking protocols that account for side-chain flexibility. AutoDock directly incorporates side-chain flexibility (at the cost of greatly increased running times), but the maximum number of torsions is restricted to 32. We thus iteratively chose binding pocket residues with growing distance to the ligand as long as the total number of torsions did not exceed this threshold. The second protocol uses FlexX with binding pocket side-chain conformations generated with SCWRL [204], the third employs FlexE [169] with side-chain ensembles derived with IRECS [208] (rotamer density 3 and at most 3 additional side-chain conformations per binding pocket residue). All protocols used the default parameters of the respective docking algorithm; the number of runs for AutoDock was again set to 25.

In contrast to the EAS which is used to define the residues supposedly involved in the backbone movements relevant for the binding pocket, for both docking rounds, we used the smaller, ligand-specific binding pocket which we defined as consisting of all residues with heavy atoms within a distance of at most 6.0Å to any of the ligand's heavy atoms.

The resulting docking poses were evaluated by calculating the symmetry-corrected RMSD to the crystallized ligand structure using the *smartrms* program included in GOLD. However, the obtained docking poses also depend to some degree on the used protein conformation: the reconstructed structures used for docking are not identical with the original holo structure, and the RMSD between the ligand in the crystal structure and the docked pose may thus be slightly biased towards the quality of the superimposition between crystal structure and the input protein conformation. But the ligand may nevertheless be able to adapt itself to slightly different protein conformations and establish the same interactions as present in the crystal structure. We thus additionally calculated the symmetry-corrected fraction of native ligand contacts of the crystal structure realized in each docked pose. The ligand contacts were determined using HBPLUS and HBADD [303] as implemented in LigPlot [304], the symmetry-corrected fraction of a pose is given as the maximum fraction of native ligand contacts over all its automorphisms as calculated by OpenBabel [299].

6

RESULTS AND DISCUSSION

6.1 SELECTING A SPRING FORCE FUNCTION

Several types of Elastic Network Models have been proposed, the main difference lying in the choice of the spring force function and its parameters. We tested five different spring force functions for their ability to concentrate the motions of interest within the first few modes:

$$
\begin{aligned}
k_1(d) &= \left(\tfrac{d}{\mathrm{d_0}}\right)^{-6} && \mathrm{d_0} = 3.8\text{Å} \\
k_2(d) &= \mathrm{e}^{-\left(\frac{d}{\mathrm{d_0}}\right)^2} && \mathrm{d_0} = 3.0\text{Å} \\
k_3(d) &= \mathrm{e}^{-\left(\frac{d}{\mathrm{d_0}}\right)^2} && \mathrm{d_0} = 7.0\text{Å} \\
k_4(d) &= \left(\tfrac{1}{d}\right)^2 && \\
k_5(d) &= \begin{cases} 1 & \text{if } d < \mathrm{d_0} \\ 0 & \text{else} \end{cases} && \mathrm{d_0} = 15.0\text{Å}
\end{aligned}
$$

The behaviors of these functions w.r.t. the atomic distance are shown in Figure 6.1. While k_5 is a step function that equally weighs all atom pairs that have a distance of less than 15Å [260], the other four functions decrease with a growing distance between the atoms. k1 is a modified version of a function presented by Kovacs et al. [305]: the original function includes an additional term $a \cdot s_{ij}$ scaling the influence of the normalized residue contact area s_{ij} [306] between pairs i, j of residues by a factor a, which was fitted on a small benchmark of only ten proteins and which we thus set to zero. This function by design pays special attention to C_α atoms that are consecutive in the protein backbone. Their average distance is 3.8Å, which exactly corresponds to this function's distance threshold d_0. k_2 uses an inverse exponential with a distance threshold of 3.0Å [259] and generally assigns weaker forces to springs than k_1. k_3 uses the same function but with a threshold of 7.0Å [307] that imposes a stronger force on more distant atoms than the other distance-dependent functions. k_4 is a parameter-free spring force function that does not employ a distance threshold and uses an inverse quadratic function to model the force values [308].

We have investigated how well each of the five functions is able to capture the conformational change from apo to holo structure in the first 10, 25 and 50 non-zero modes. The results are given in Table 6.1. The left-hand side shows the mean performance in terms of the fraction of the conformational change from apo to holo conformation that could be achieved using the respective number of modes. The righthand side shows the average rank when ranking the performances of each function on a per-apo/holo-pair basis. For comparison, the results (based on the binding pocket RMSD) using a full Hessian approach instead of an effective Hessian are also shown.

Regarding the mean performance, two groups are observable: the first one comprising k_1, k_3, and k_2, the second one consisting of k_4 and k_5. The mean performance in each group is similar; a comparison with the behavior of these functions (Figure 6.1) provides an explanation for these results: the first group contains those functions

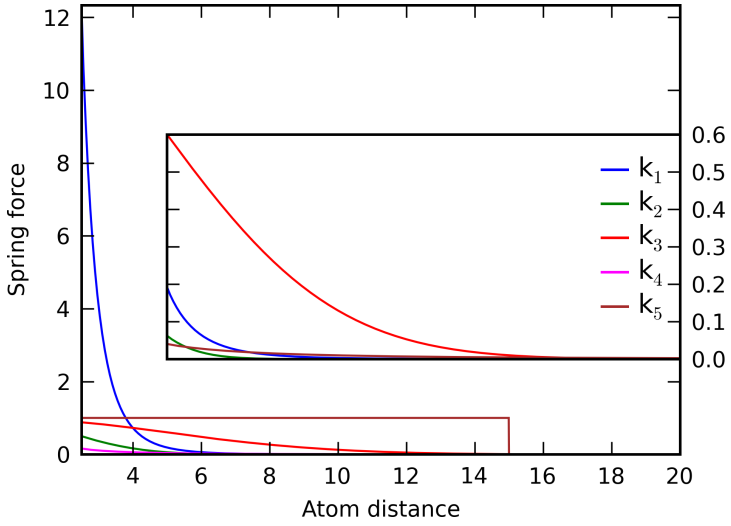


Figure 6.1: Spring force as a function of the atom distance. The insert shows a magnification of the curves beyond a distance of 5Å.

that clearly favor springs between close atoms while the second group is made up of those that hardly differentiate between springs for neighboring and remote atoms. The performance of group 1 is approximately 16-18% better than that of group 2. This clearly demonstrates the importance of adapting the spring strength to the distance between the participating atoms.

The table contains the average performances and ranks of each spring force function over the complete data set. Data rows and columns are set up to facilitate the comparison of performance according to the aforementioned criteria for the first 10, 25, and 50 modes.

Furthermore, it can be observed that the models based on the effective Hessian are superior to the models based on the full Hessian. This result is expected, as the normal mode space of the effective models is much smaller than that of the full models. The moderate difference between effective and full models shows that at least some of the conformational change in the binding pocket can be attributed to global, collective motions. However, some of the first modes in the full models describe movements elsewhere in the molecule and thus negatively influence the performance.

Because the results are quite similar among the models in group 1, the decision on which force function to use in the upcoming experiments was made by additionally determining the mean rank of each function over all apo/holo pairs. Here, a clear preference for k_1 is observable for all three mode sets, having a mean rank of about 1.9 - 2.0. While k_3 behaves similarly for the first 10 modes, the difference of mean ranks to k_1 increases with large mode sets. k_2 shows a similar behavior to k_3 for the larger sets while the ranking of k_4 and k_5 is clearly worse than that of the former three functions.

We thus decided to use k_1 for the upcoming experiments. This choice seems reasonable as k_1 strongly penalizes relative movements of subsequent C_α atoms, allows a certain amount of movement for close atom pairs, and lessens the impact of springs representing long-range interactions.

		Mean performance (fraction of RMSD reduced)						Mean ranks					
		Effective Hessian			Full Hessian			Effective Hessian			Full Hessian		
	#modes	10	25	50	10	25	50	10	25	50	10	25	50
k_1	10	0.266			0.140			1.994			2.145		
	25		0.411			0.307			1.871			2.256	
	50			0.560			0.482			1.907			2.160
k_3	10	0.264			0.130			2.079			2.849		
	25		0.400			0.288			2.234			2.769	
	50			0.556			0.468			2.422			2.621
k_2	10	0.252			0.156			2.546			2.356		
	25		0.402			0.317			2.303			2.168	
	50			0.549			0.478			2.446			2.082
k_4	10	0.144			0.094			4.549			4.062		
	25		0.282			0.238			4.452			4.155	
	50			0.463			0.401			3.968			4.041
k_5	10	0.178			0.117			3.833			3.588		
	25		0.311			0.255			4.140			3.653	
	50			0.459			0.410			4.256			4.097

Table 6.1: Overview of the performances for the investigated spring force functions and Hessian models.

6.2 COMPARISON WITH NORMAL MODES FROM A MOLECULAR MECHANICS FORCE FIELD

The main focus of this study is the investigation and application of normal modes in a high-throughput scenario where computational efficiency is of great importance. Elastic Network Model normal modes are especially suitable for this task as they do not require any preprocessing of the input structures. In contrast, to obtain normal modes from a molecular mechanics force field, a thorough energy minimization of the input structure is necessary, because the underlying assumption of normal mode analysis is that the protein oscillates harmonically around an energetically stable minimum conformation.

Nevertheless, the comparison of the performance of normal modes obtained with the spring force function we selected in the previous section with those derived from a molecular mechanics force field may gain additional insights into the suitability of elastic network normal modes to model binding pocket related backbone movements.

The generation of force field normal modes was carried out using the GROMOS G53a6 [309] united-atom force field as implemented in GROMACS 4.5 [310]. Parameters and topologies for hetero groups that are not defined in the force field were obtained from the PRODRG2 Server [311]. Seven of 69 apo structures were removed from the data set because they contain prosthetic groups, cofactors, or other hetero groups with elements that PRODRG2 cannot handle, e.g., the Nickel-reconstituted heme group in 1qsi and 1qsh. The remaining apo structures were fully minimized to a maximum force $F_{max} < 0.001$.

During the energy minimization of the remaining 416 apo/holo pairs, the C_α RMSD between the minimized apo and the crystal holo conformation ($RMSD_{min}$) usually increased in comparison with that of crystal apo and holo structures ($RMSD_{cryst}$); the corresponding distribution of $\Delta RMSD = RMSD_{min} - RMSD_{cryst}$ is given in Figure 6.2. It can be clearly seen that there is a significant difference in most cases, with 250 pairs

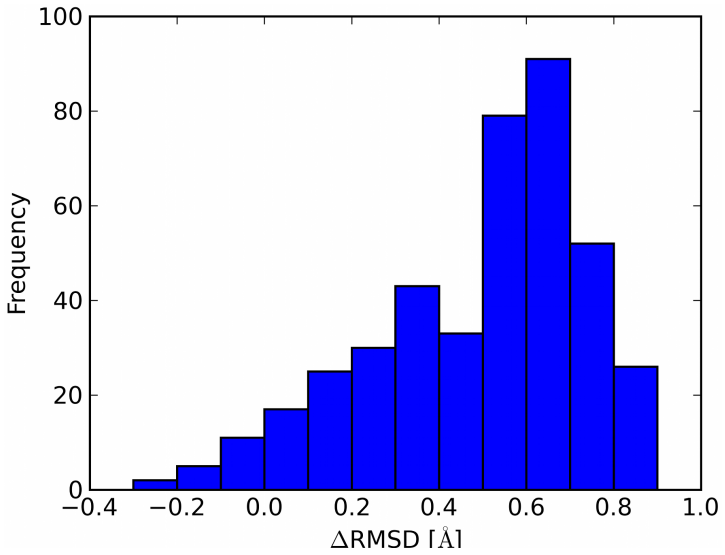


Figure 6.2: Distribution of difference between RMSD of minimized apo/cystal holo and cyrstal apo/crystal holo.

#modes	ENM modes on crystal apo	ENM modes on minimized apo	Force-field based modes (C_α)	Force-field based modes (all-atom)
10	0.266	0.159	0.044	0.040
25	0.412	0.314	0.112	0.105
50	0.561	0.509	0.242	0.269

Table 6.2: Mean reconstruction performance of the different approaches in terms of fraction of C_α RMSD that could be reduced using the different mode subsets.

having a $\Delta\text{RMSD} > 0.5$, which makes a direct comparison of the performance of our ENM-based approach to force-field based normal modes difficult.

On the minimized apo conformations, we thus computed normal modes using three different approaches:

- a force-field based effective Hessian with only the extended active site (EAS) C_α atoms
- an effective Hessian of the force-field comprising all EAS atoms (side chain and backbone)
- a recomputation of the effective Hessian of the ENM containing the EAS C_α atoms to account for the ΔRMSD of the apo/holo pairs as shown in Figure 6.2

The results are given in Table 6.2. For reasons of a better comparison, the ENM results on the corresponding 416 crystal apo/holo pairs are also included.

It can be seen that, for all three mode subsets, the performance of the ENM normal modes on the minimized apo structure is clearly superior to that of both approaches using force-field based normal modes. It is interesting to note that the performance of the ENM modes on the minimized apo conformation rapidly approaches that on the crystal apo structure. In contrast, the performance of both approaches using force-field based normal modes is already very similar when using the subset containing

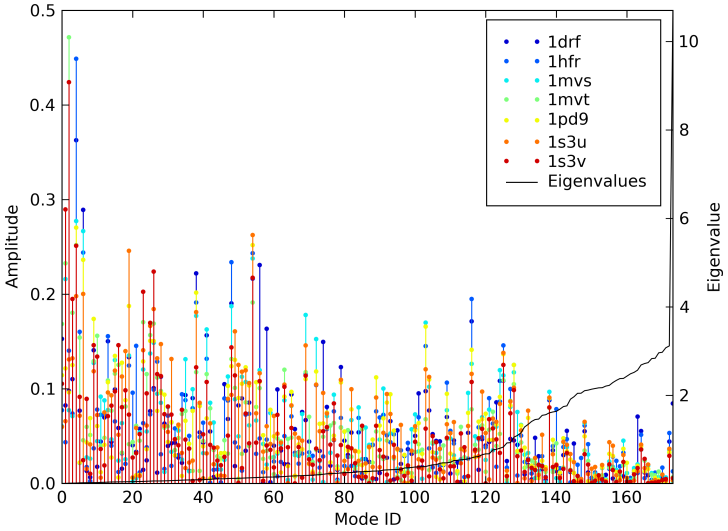


Figure 6.3: Mode amplitude spectra for a reconstruction of different dihydrofolate reductase holo structures from one single apo structure (1pdb). The modes are sorted by increasing eigenvalues. Modes and amplitudes (dimensionless scalar factors) differ for a reconstruction of the individual holo conformations.

only 10 modes. A possible reason for this may be the derivation of the effective Hessian: $\mathbf{H}_{ee}$, the matrix containing all couplings of atoms outside the EAS (see Section 5.2) is very large. Inverting such a large matrix can incur numerical problems, which may be reflected in the above results. Moreover, all these detailed couplings may add a certain amount of randomness to the motion of the C_α atoms in the binding pocket and thus also to the normal modes. The normal modes in a C_α ENM where such details are neglected may thus be better able to capture binding-pocket related collective motions. A more detailed investigation of the reasons for the poor performance using force-field based binding pocket normal modes seems interesting, but would be beyond the focus of this study.

6.3 ANALYSIS OF NORMAL MODE AMPLITUDE SPECTRA

The general assumption of using normal modes in protein-protein docking as well as conformational studies of proteins is that only a few modes are required to reproduce most collective, global conformational changes that the protein is able to perform.

To investigate whether this assumption also holds in the protein-small molecule docking case, we thus first compared the mode amplitude spectra for a full reconstruction of the C_α trace of different holo structures using the normal modes obtained from the effective Hessian of one common apo conformation. If the initial assumption also holds for protein-small molecule docking, the used modes and the corresponding amplitudes should be similar for a reconstruction of different holo conformations. Figure 6.3 shows such a spectrum for the protein dihydrofolate reductase (for a better insight into the differences, the absolute amplitude values are shown). The modes are sorted by increasing eigenvalues, such that only the first few modes should suffice to represent a conformational change if the fundamental assumption behind the normal modes procedure is valid.

The main difference between the displayed structures comprises a conformational change in two loops, while the rest of the system remains relatively rigid. While these movements do not involve the entire protein, they exhibit a certain amount of collectivity on the scale of the binding pocket, on which we focus with our effective Hessian approach. Hence, the spectrum should contain regions with a clearly similar behavior for all holo structures.

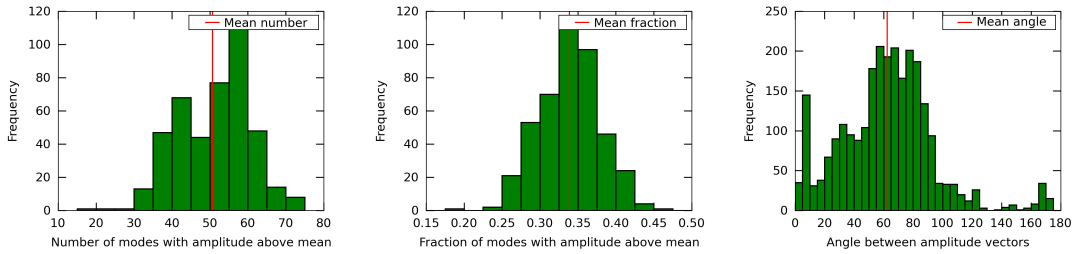
It can be clearly seen that this is not the case: not only do the amplitudes differ, but also modes that strongly contribute to reproducing one conformation have almost no influence for other conformations and vice versa, a fact which makes an a priori selection of relevant modes difficult. For example, a conformational change from 1pdb to 1s3v (red) requires a large amplitude for mode 23, while a reconstruction of 1drf (blue) can be performed very accurately without using that mode. On the other hand, generating the backbone conformation of 1drf cannot be achieved without using mode 58 (with an even higher amplitude than mode 23 for 1s3v) while this mode plays almost no role for 1s3v.

One reason for this difference in the relevance of modes is that the eigenvalues which correspond to the energy required to perform a movement along a mode are very similar for a large fraction of the modes: a protein that is excited by a certain amount of energy distributes this energy evenly among all its degrees of freedom. Movements that require less energy thus dominate those that need much energy.

However, if there are many energetically similarly demanding modes, the space of possible motions grows exponentially and the conformations become more diverse as a result. In addition, elastic network models do not account for anharmonic motions, which may become especially important for small-scale, local backbone movements; a fact that may also contribute to the different relevance of the modes. Furthermore, a bound ligand can shift the minima on the protein energy landscape and, thus, conformations that are less likely in absence of a ligand may attain a lower overall energy in the bound complex due to favorable interactions with a ligand. This is in concordance with the observation that, as soon as the eigenvalues increase significantly (approx. mode 130 in the case of 1pdb), the mode amplitudes decrease for the whole set of reconstructed holo conformations.

The results for the whole set of apo/holo pairs are summarized in Figure 6.4 and confirm the findings detailed for dihydrofolate reductase: the distribution of the number of modes with an absolute amplitude greater than average indicates that a sampling in a small set of modes is not sufficient to reconstruct a holo structure with high accuracy. On average, 51 modes are responsible for the largest part of the conformational shift, an observation that is consistent with the results shown in Figure 6.3. The corresponding fractions of modes (Figure 6.4b) to be used range from 0.19 to 0.46 with a mean value of 0.34. Accordingly, at least one third of all modes must be considered in a conformational sampling.

However, an a priori selection of modes seems hard at best, as the amplitude vectors greatly differ in their composition between complexes with different ligands. Figure 6.4c shows the distribution of pairwise angles between amplitude vectors for a reconstruction of different holo structures from the same apo conformation. The main fraction of angles lies between 50 and 90 degrees (mean value 63.1), a fact that not only confirms the assumption that the relevance of modes highly depends on the bound ligand, but also shows that many amplitude vectors are almost perpendicular to each other and that modes which are switched off in one complex are essential in another.



(a) **(b)** **(c)**

Figure 6.4: Distributions of the number (a) and fraction of modes (b), as well as the pair-wise angles between amplitude vectors (c) for a reconstruction of different holo structures from the same apo structure.

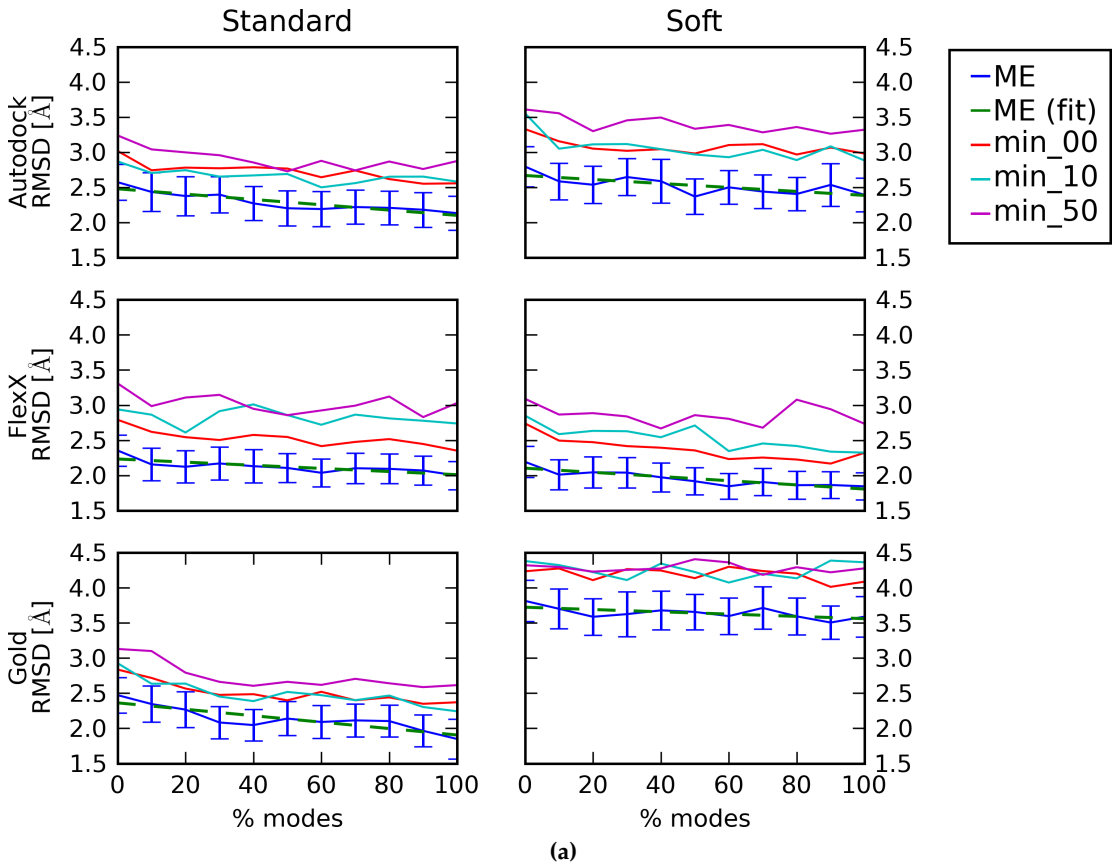
Thus, in the normal case that the bound conformation is unknown and an a priori predictor for the modes relevant for individual ligands does not exist, the number of modes that has to be sampled can be expected to be well above the average number of 51.

6.4 DOCKING INTO RECONSTRUCTED HOLO STRUCTURES

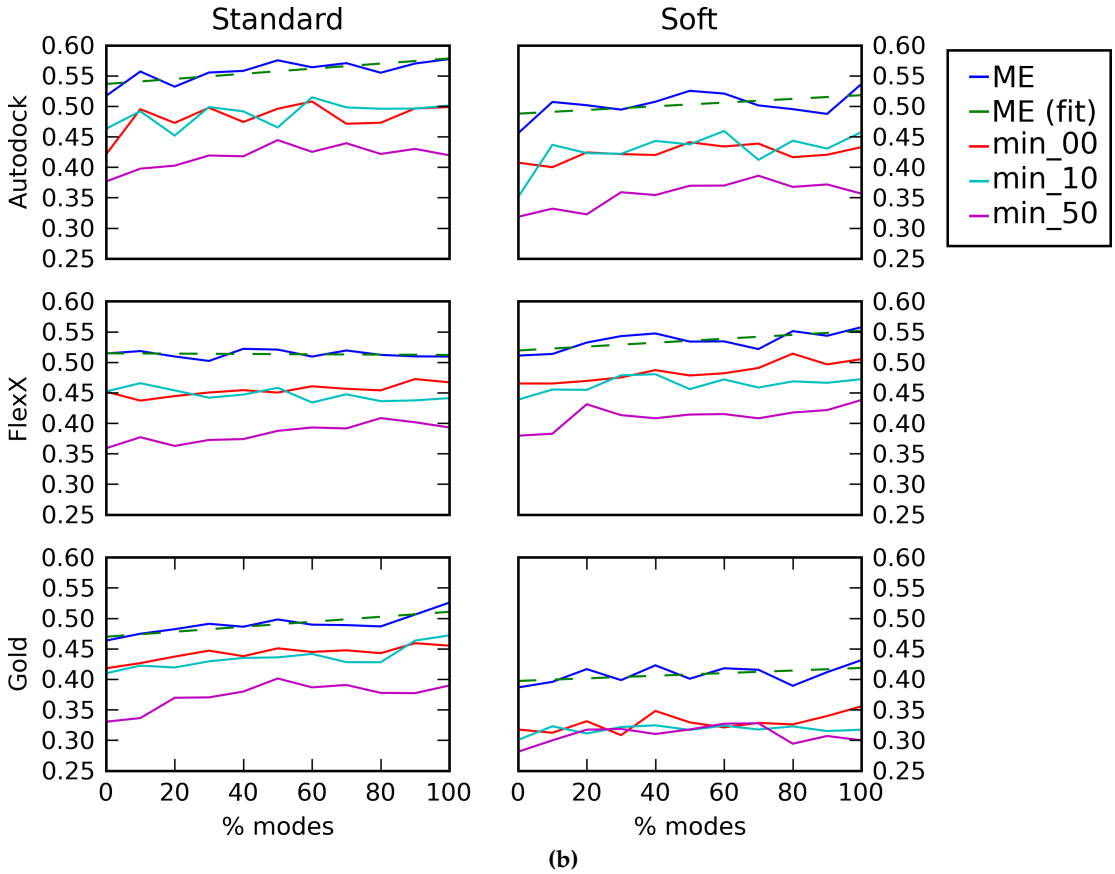
The analysis of the mode amplitudes has shown the highly diverse nature of transitions from apo to different holo structures and demonstrated that the important modes are distributed over almost the full range of modes. However, it has yet to be clarified whether all these modes or only a subset of the most collective ones (i.e., those with the lowest eigenvalues) are required to achieve a conformational change that results in a successful docking of the ligand.

We established 18 docking series, consisting of 6 different protocols using AutoDock, GOLD, and FlexX each in a standard and a soft setup, for each of the 3 different minimization lengths (0, 10, and 50 steps). While many (partially contradictory) studies that compare the performance of different docking tools exist, our primary aim of using different docking programs here is to ensure that the obtained results are not due to peculiarities of any of these tools. We thus do not compare the actual performances of the different programs but rather use them to frame a stable picture of the capability of normal modes to improve small-molecule docking. In some cases, the docking failed due to structural problems, e.g., when the reconstruction procedure produced irresolvable clashes or the atom types could not be assigned properly. The missing results were interpolated using natural splines; eight apo/holo pairs were excluded because they produced five or more missing values in at least one of the docking series.

Figure 6.5 illustrates the docking results for the remaining 425 pairs for each docking series. The results have been normalized to account for the unbalanced distribution of the number of apo conformations associated with each holo structure and the number of holo structures per protein. In addition to the data series for the three minimization protocols, a minimum envelope (*ME*) curve which considers only the optimum value obtained from the three protocols for each apo/holo pair and a linear least-squares fitted line for the *ME* curve are shown in each plot. For the data points on the *ME* curve in the pose RMSDs, the standard errors in the mean are also shown.



(a)



(b)

Figure 6.5: Best pose RMSD (a) and maximum fraction of contacts (b) for holo reconstructions with an increasing normal mode subset size averaged over the full data set.

In the ideal case, one would expect a steep decrease of the minimum RMSD for the most collective modes that diminishes as the normal mode subspace grows. This would indicate that only the first modes are required to produce a conformational change that is sufficient for a successful docking.

However, in our best-case scenario, the overall drop in minimum RMSD is small and essentially linear (Figure 6.5a). This tendency is observable in each of the 18 docking series and even more imminent in the *ME* curves. Regardless of the actual docking performance of the different protocols, the reduction in RMSD compared to the docking results with the apo structure is at most 0.6Å (mean value 0.4Å) when including 100% of the modes in the holo reconstruction; for the first 20% of the modes the average reduction amounts to only 0.26Å. This behavior is also reflected by the fitted lines: the steepest decay was found to be -0.0046 with a residual sum of squares (RSS) of 0.07. The standard errors of the means for the 100% (and 0%) reconstruction in the standard protocols of AutoDock, FlexX and GOLD are 0.24 (0.26), 0.20 (0.22), and 0.28 (0.25), respectively. The values for the soft protocols are comparable, indicating a significant improvement in all six protocols.

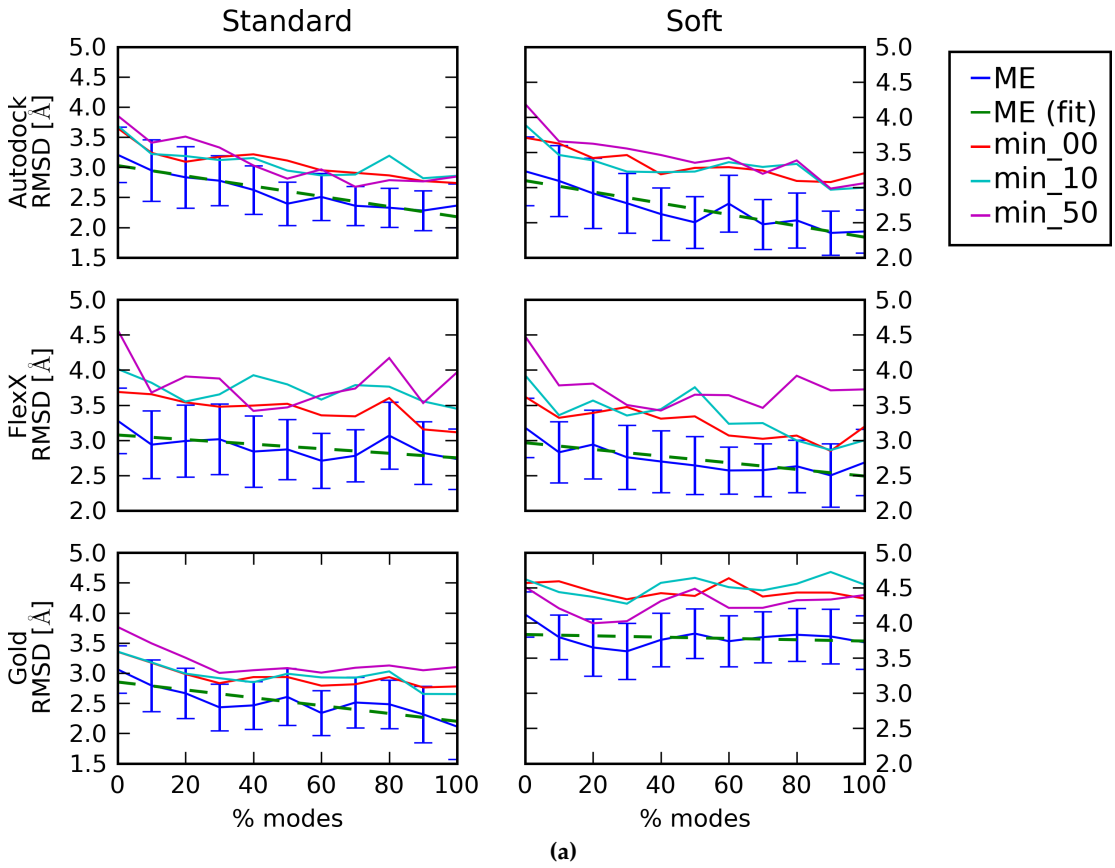
Similarly, the maximum fraction of native contacts (Figure 6.5b) grows linearly and increases by at most 0.081 when including all modes, and only by 0.049 for the first 20% of the modes. The fitted *ME* lines have a maximum slope of 0.00042 with an RSS of 0.0015.

We also investigated the top scores and the corresponding poses (data not shown), and the results reveal another factor that negatively affects the usability of normal modes for sampling binding pocket conformations: the top scores differ by ≈8%, on average, between apo and the reconstructed holo structures, which is, from our experience, far below the standard deviation of scores obtained from a typical docking run. A linear least-squares fit gave a maximum decay of -0.023 with an RSS of 1.68, showing that there is basically no decline in the top scores. Likewise, the top-pose RMSDs were reduced by at most 0.5Å using a full reconstruction and only 0.17Å for the first 20% of the modes. These results show that, even in the case that a conformational sampling in the most collective modes would reproduce the original holo structure, it will be difficult to find the correct protein-ligand complex in the set of generated protein conformations with today's scoring functions, unless additional terms that estimate the plausibility of the different protein conformations are incorporated.

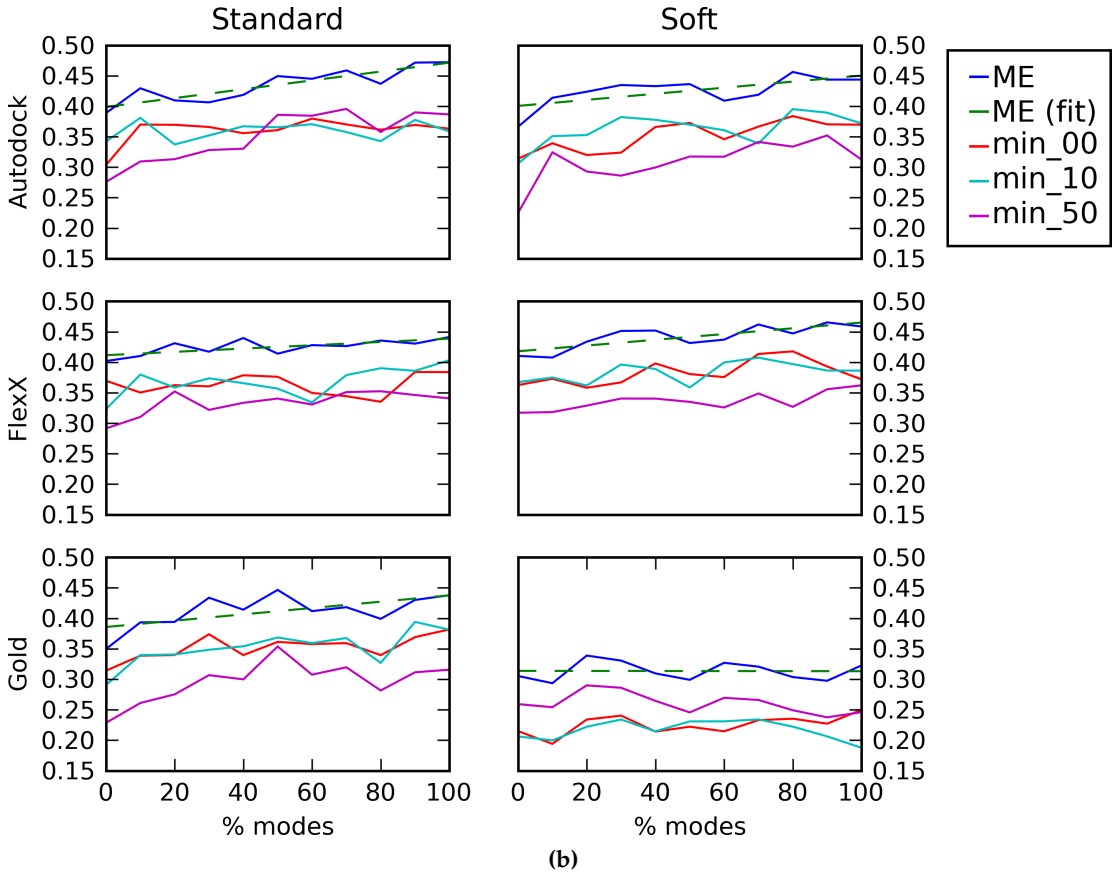
Because normal modes are expected to reproduce the large-scale motions of a protein especially well, we also investigated the subset of 165 apo/holo pairs with a C_α RMSD > 0.5Å (Figure 6.6). The results are mostly comparable to those on the full data set: Due to the larger conformational difference between apo and holo structure, the best-pose RMSDs obtained from docking into the apo conformation are larger on average than for the full data set (cmp. Figure 6.5).

The standard errors of the means for the 100% (and 0%) reconstruction in the standard protocols of AutoDock, FlexX and GOLD have values of 0.36 (0.46), 0.43 (0.47), and 0.54 (0.39), respectively. This implies that, at least in some cases, the significance of improvement in docking performance is questionable, however, both AutoDock protocols and the GOLD standard protocol can be considered to achieve a significant improvement.

In comparison to the full data set, the best-pose RMSD decreases faster as more modes are used to reconstruct the holo structures. But the decay in RMSD is, in essence, still linear (maximum slope of the linear least-squares fit -0.0085, RSS 0.12)



(a)



(b)

Figure 6.6: Best pose RMSD (a) and maximum fraction of contacts (b) for holo reconstructions with an increasing normal mode subset size averaged over the subset of apo/holo pairs with a C_α RMSD > 0.5.

and the best-pose RMSDs for the 100% reconstructed holo conformations are 0.36Å greater than those of the full data set, on average.

The best results for a single protein (data not shown) were obtained for aldose reductase (pdb code in the Astex Diverse Set: 1t40) where the best-pose RMSD resulting from docking into the apo structure consistently dropped from values of between 3.89Å and 4.34Å to below 2.0Å after an inclusion of 50% of the modes in four of the six docking protocols. Although such an increase in performance seems encouraging, at first sight, using 50% of the modes is still an infeasible task for a conformational sampling, especially in a high-throughput setting.

These findings indicate that, even for larger C_α RMSDs, the movements in the binding pocket upon ligand binding are not collective enough to be represented by a small set of normal modes, in general. Thus it may be indispensable to use the ligand in some way to perform a pre-selection of the required modes, or to directly guide the conformational change upon ligand binding. To do so seems difficult, however, given the problems of today's scoring functions in discriminating between correct poses and decoys.

6.5 DOCKING WITH SIDE-CHAIN FLEXIBILITY

In this section, we study how strongly the previous results depend on the side-chain conformations in the binding pocket. To this end, we selected the 59 apo/holo pairs from our data set that were successfully redocked into the original holo structure with a best-pose RMSD of less than 2.0Å in at least one of the docking series, but failed to do so for the corresponding 100% reconstructed holo structure. Because the C_α trace of this structure is identical to that of the crystal holo structure, the problem reduces to non-C_α backbone atoms and, more importantly, the side-chain conformation.

Figure 7 shows the results for the three additional docking protocols: AutoDock with flexible sidechains, FlexE with IRECS-computed side-chain ensembles and FlexX using side chains generated with SCWRL. AutoDock with flexible side chains as well as SCWRL+FlexX show no clear tendency towards an improvement, the fitted *ME* lines have slopes (RSS) of -0.0005 (0.46) and 0.0016 (0.49), respectively. In contrast, the line for IRECS+FlexE demonstrates that using a side-chain ensemble may help to improve the docking performance on a protein conformation generated using normal modes. The corresponding slope and RSS are -0.0134 and 0.97 respectively.

Partial improvements over the original best-pose RMSD and successful dockings could be obtained with all docking protocols, as can be seen in Table 6.3. In total, improvements in best-pose RMSD were achieved for 46 of the 59 apo/holo. AutoDock and FlexE both gave better results in 25 cases, FlexX in 14 cases. Altogether, at least one docking pose with an RMSD below 2.0Å could be obtained for 26 of the 59 apo/holo pairs. FlexE was most successful with 19 poses in total and had the absolute minimum RMSD in comparison to the other docking results in 15 of these cases (last column). For the AutoDock protocol these numbers were 8 and 7, respectively, for FlexX they were 10 and 4, respectively. Nevertheless, not all docking results could be improved. The side-chain rotamers are backbone-dependent, but even in the 100% reconstructed conformations, where the C_α trace is equal to that of the original holo structure, the conformation of the non-C_α backbone atoms can differ slightly from that of the original structure since the elastic network only acts on the C_α atoms and the

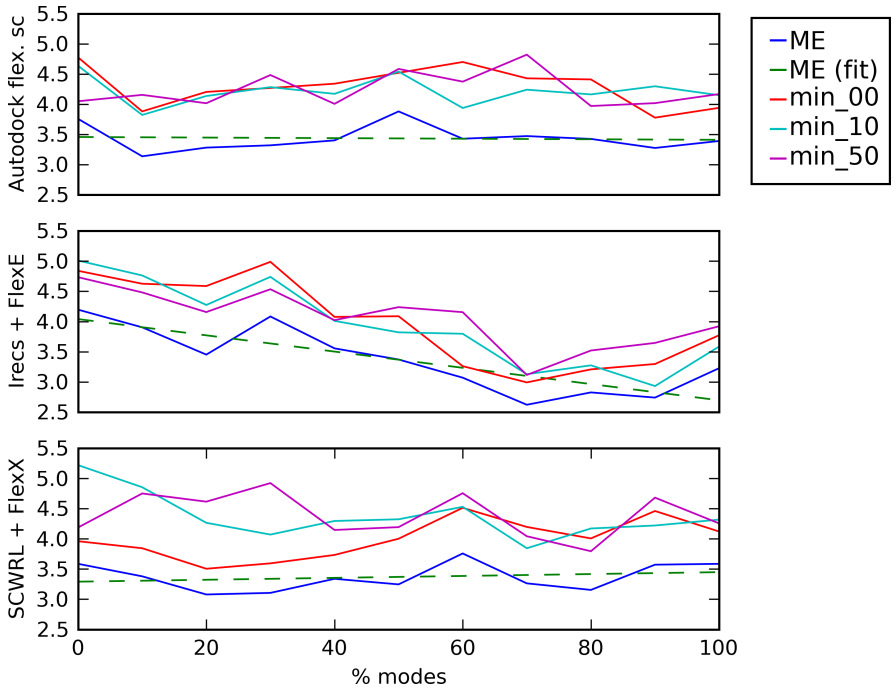


Figure 6.7: Performances of the three docking protocols explicitly accounting for side-chain flexibility.

Protocol	Improvements	Successes (best pose RMSD < 2.0Å)	
		overall	best in comparison
AutoDock, flex. side chains	25	8	7 (27%)
Irecs + FlexE	25	19	15 (58%)
SCWRL + FlexX	14	10	4 (15%)
	46 (78%)	26 (44%)	

Table 6.3: Results for the docking protocols with flexible side chains on the fully reconstructed holo structures.

remaining ones are approximately reconstructed from these (see Section 5.4). Using normal modes obtained from a backbone heavy-atom ENM instead yields the correct backbone conformation for a 100% reconstruction; however, doing so increases the set of resulting normal modes by a factor of 4, which leads to the conclusion that the chance to achieve better results in such a scenario is small, at best.

These results imply that the poor docking performance on normal-mode generated protein conformations may be improved when including side-chain flexibility. For docking algorithms that do not model side-chain flexibility explicitly, it may even be necessary to include not only one side-chain rotamer but an ensemble thereof to effectively increase the chances of a good docking result. However, this can greatly increase the computational effort required for conformational sampling, both for the generation of protein conformations and the dockings to be performed. But even when including side-chain flexibility, 70% of the modes were required, on average, to obtain a pose with an RMSD below 3.0Å. This shows that the capability of normal modes to model binding pocket rearrangements is strongly limited even when accounting for side-chain conformations.

7 CONCLUSIONS

The aim of this study was to empirically gain insight into the usability of binding-pocket normal modes obtained from C_α-ENMs in protein-small molecule docking. We have established a scenario that provides an upper bound for conformational sampling algorithms: for known holo structures, we have generated optimal reconstructions w.r.t. differently sized normal mode subspaces retrieved from corresponding apo structures.

The analysis of mode amplitude spectra and the subsequent docking experiments have shown that the use of normal modes in protein-small molecule docking is limited: the amplitude vectors to be used differ greatly when reconstructing holo structures for different ligands from the same apo structure. This may not always be the case. If the conformation changes globally upon ligand binding, an improvement in docking accuracy can be achieved with a small set of modes, as shown in *May et al.* [292] and *Cavasotto et al.* [293]. In this study, Cavasotto et al. also introduced a measure of relevance to determine the modes that are involved in binding pocket conformational changes. This method makes is possible to narrow down the sampling space to a small set of modes and does not necessarily require the calculation of the effective Hessian as done in our study. Ensembles generated from such mode sets in this study have shown to improve docking results for several ligands of cAPK Kinase ligands. This approach is especially powerful if the mobility of a binding pocket is well-defined and mostly independent of the ligand, as for example in the conformational selection stage during protein movement. However, in case of local ligand-specific induced fit movements, if the binding pocket motions are unknown or cannot be well captured by a small set of relevant modes, this method is difficult to apply.

Even if the conformational change is not fully represented by the most collective modes, the amplitude vectors for the conformational change upon binding different ligands may show a high degree of similarity. For example in the case of calmodulin, a protein that changes its tertiary structure from an elongated form to a globular conformation when binding a ligand [312], the amplitude vectors are very similar - even if the ligands are highly diverse - due to the dominating complexity and the highly distinct conformations in the bound and unbound state [291]. However, in our study, the conformational changes are less extensive, and our results give rise to the assumption that, in such cases, the ligand information is of great importance when selecting the relevant modes, as most of the modes are energetically almost equivalent and nearly equally likely to be activated when binding a ligand.

But the problem how to include information on the ligand in the selection procedure is unsolved: state-of-the-art scoring functions are hardly accurate enough to even reliably select the original pose from the set of generated solutions in the redocking case. Hence, their general usability in helping to find the relevant modes for the given ligand during a conformational search is more than questionable. To our knowledge, approaches to directly use the ligand as a predictor for the relevant normal modes do not exist, and the question whether this is possible at all has not even been approached.

Due to the mutual dependence of finding the correct ligand conformation and determining the true protein conformation, researchers are currently forced to apply conformational sampling strategies. Our docking results show that sampling with a fraction of only the first few, most collective modes is not sufficient to significantly improve the docking performance in general, i.e., when no large-scale motions are involved in ligand binding. Furthermore, although sampling with large fractions of the normal mode space can improve the results, the computational effort increases exponentially with the number of modes and thus the docking may become infeasible.

The additional docking experiments accounting for flexible side chains show that it is often indispensable to adjust the side-chain conformations upon backbone movement and that doing so can enhance the docking performance when applied in combination with normal modes. But while the docking results could be improved using flexible side chains, the number of modes required for obtaining reasonable results was still too large to be applicable in high-throughput settings.

Summarizing these observations, the general reduction in the complexity of modeling protein flexibility with normal modes in protein-ligand docking is small if relevant modes cannot be determined from some external criterion, because even in our best-case scenario, where the actual holo conformation and the path from the apo conformation are known, the gain in docking performance is small and will be hard to achieve in an actual sampling scenario. Moreover, structural uncertainties in the atom coordinates or the fact that normal modes are designed to mainly detect collective motions of a system and that a large number of normal modes is typically required to describe local changes involving single atoms can cause the normal modes to fail in protein-ligand docking. This leads to the strong assumption that the use of normal modes in protein-small molecule docking may be restricted to select cases where only few collective motions are responsible for binding a ligand.

Part III

ASSEMBLING MACROMOLECULAR COMPLEXES BASED ON PAIRWISE DOCKINGS

8 INTRODUCTION

In Sections 2.3 and 2.4 we have already seen that the biochemical processes in cells are the result of the complex interplay between DNA, RNA, and proteins as well as other macromolecules and chemical compounds. In Section 2.5, we have addressed some examples of how proteins arrange themselves into multimeric protein assemblies to enable a highly efficient metabolism, for example regarding the conversion of energy, to synthesize and decompose molecules, effect communication, or to carry out protective functions.

To this end, the individual protein subunits of the complex must be able to assemble and to find their respective position in the complex (sometimes with assistance of molecular chaperones) and under the complicating conditions of the crowded interior of cells. However, the concepts of self-recognition and whether and how different hierarchical sub-stages during assembly are involved are still largely unknown, though hydrophobic and sometimes electrostatic interactions are deemed to be the major driving forces [313, 314, 315, 316, 317]. An experimental determination of such interactions and the resulting complexes can be expensive and time-consuming. Here, the use of bioinformatics methods and algorithms, such as docking, alignment and assembly tools in this context, can help to efficiently guide or sometimes even replace such experiments and to enhance the rational understanding of the processes involved in complex assembly.

The diversity of protein complexes can be assumed to be immense, as we have already seen in Sections 2.3 to 2.5. They can greatly differ in size of the individual monomers, the size of the total complex, the number of components as well as the number of different protein types involved, the contacts established between the monomers and their symmetry properties. This assumption is not only confirmed by the 3D complex database [83], but also shown by the following examples: the yeast ribosome which is responsible for protein biosynthesis (see Section 2.3) contains 79 proteins which are all unique (and in addition four different rRNA molecules; distributed over pdb codes 3U5E, 3U5F, 3U5G, 3U5H) and consist of 46 to 387 amino acids [318]. It has a diameter of $\approx$ 30nm and exhibits no symmetry. The nuclear pore complex (NPC) which serves as a gate that restricts the exchange of macromolecules and chemical compounds between nucleus and cytoplasm of eukaryotic cells is even larger, in fact probably the largest protein complex in the cell: in vertebrates the complex has a diameter of $\approx$ 145nm, a molecular mass of $\approx$ 125 MDa, and consists of more than 450 proteins of about 30 distinct protein types [319, 320]. Even though the NPC contains almost six times as many proteins as the ribosome, it comprises not even half as many distinct protein types, but shows an octagonal symmetry. The range of diversity observable in protein complexes can even be expected to increase with the technological progress of current protein structure determination methods (see Section 3.1) and the availability of newly resolved complex structures.

From an algorithmic perspective, assembling oligomeric complexes from their monomers can be perceived roughly as solving a three-dimensional jigsaw puzzle. However, in contrast to a real jigsaw puzzle, the interfaces are not so well defined

in terms of the complementarity of both their surfaces and their biochemical properties. In addition, conformational changes upon assembly can alter the interfaces to some extent. If the interfaces are roughly known, for example from site-directed mutagenesis [321], studies on correlated mutations [322, 323, 197, 324, 199], cross-linking experiments [325, 326, 327], or databases such as Interactome3D [218], one typically employs docking methods for local sampling, yielding a set of hundreds to thousands of plausible docking poses. While state-of-the-art docking algorithms are able to find and sample near-native binding modes, scoring and ranking the solutions appropriately still presents a major problem [158].

Hence, computationally, the assembly of large protein complexes poses a challenging combinatorial problem which has recveived attention from the algorithm development community only in the last decade. Multi-body docking approaches are scarce (see Sect. 3.3), the most prominent being HADDOCK [191], an information-driven docking algorithm that facilitates the simultaneous docking of up to six proteins (see Sect. 3.3.6). Most approaches rely on prior symmetry information: e.g., the multi-body docking algorithm implemented in ClusPro uses pairwise dockings and symmetry constraints to assemble homo-oligomeric complex [193] (see Sect. 3.3.7). For reasons of computational complexity, both HADDOCK and the ClusPro multi-docking algorithm limit the size of the complexes that can be assembled to a maximum number of six components. A recent approach uses particle swarm optimization and additionally employs molecular dynamics conformational sampling to predict symmetric homo-oligomers, the largest with 24 subunits [328]. Other approaches that rely on symmetry information are SymmDock [189] and Rosetta's symmetry docking protocol [329]. DockTrina [330] does not pre-suppose any symmetry and can predict non-symmetric trimers by scanning pairs of pairwise dockings via an RMSD-based test. CombDock [184, 185] (see Sect. 3.3.5) and an ant-colony approach proposed by *Venkatraman et al.* [331] both combine pairwise dockings to generate clash-free minimum weight spanning trees.

8.1 PROBLEM STATEMENT

In this work, we develop 3D-MOSAIC, a novel combinatorial algorithm that employs a tree-based greedy scheme to iteratively assemble protein complexes from binary docking data. Contrary to approaches that rely for example on electron density maps or other low-resolution information on the full complex topology to generate a corresponding high-resolution model of the respective assembly, the focus of our work is the stepwise assembly of such complexes from the monomeric proteins.

The iterative approach acknowledges the fact that the formation of macromolecular complexes most likely does not happen by a spontaneous and simultaneous assembly of all involved monomers. Rather, analogous to the mechanisms involved in folding of polypeptide chains, it can be assumed that the strongly interacting monomers of the complex assemble first to provide one or several stable cores. After these core components have formed, they may associate over a set of more weakly interacting interfaces and additional monomers may be attached until the full complex has been established (compare Sections 2.1.3 to 2.2).

With our method, we aim at modeling complexes for which no structural information of the full assembly, for example in the form of low-resolution data, can be obtained, but high-resolution data of the individual components (monomers) of the

complex is available. Because there exist a manifold of different possibilities to obtain information on interactions between the monomers of a complex (see Section 3.5), we can assume that at least vague information on the interfaces and binding modes between the individual monomers of the complex is available, for example in terms of potential interactions between individual residue pairs. These assumed binding modes can then be sampled using a pairwise docking algorithm of the user's choice. Several ways of providing the information on such interface locations and binding modes to the docking algorithm are possible: for example through interaction constraints derived from this information or by providing start dimers to the algorithm in which the respective interface areas of the monomeric proteins are roughly oriented towards each other. In addition, the parameters of the docking algorithm can be adjusted such that only the local neighborhood of the start dimer or areas in the search space for which the interaction constraints are fulfilled are sampled.

Furthermore, incorporating both side-chain and backbone flexibility in docking methods still represents a major problem, because it greatly increases the computational time required both during the sampling space, as conformational degrees of freedom must also be taken into account, and second, also during scoring, because often, more expensive scoring functions must be employed and the individual energies arising from the internal interactions of each protein conformation must also be computed to obtain a reasonable docking score [288, 332]. We thus assume that the conformations of the proteins used during docking do not deviate too much from those in the complex and that the set of obtained docking poses contains dimers that are similar to the respective binding modes in the complex. Small conformational differences in side-chain or backbone orientation are tolerated and can be dealt with by allowing for a certain amount of penetration of the individual complex partners. The models generated by our approach can then be used as an input for more compute-intensive methods such as all-atom energy minimizations and MD simulations using molecular mechanics force fields.

To make the algorithm applicable to a wide range of different scenarios, it should use only a minimum amount of information: high-resolution structural data for the involved protein types and their stoichiometries as well as the rough knowledge of the binding modes present in the complex, represented by sets of pairwise docking poses as explained above.

However, commonly used scoring functions are generally limited in their power of ranking near-native binary poses and discriminating them from decoy poses [160]. To deal with this ranking problem, we introduce a novel measure, called *transformation match score* (herafter, *tms*), which scores (sub-)complexes based solely on the mutual compatibility of docking poses obtained by the employed docking algorithm (RosettaDock [180, 181] in this study).

In Nature, complexes assemble without the intrinsic objective of being symmetric. Though the formation of symmetric protein complexes is assumed to have several different beneficial reasons, for example stability, error control in translation, finiteness of the assembly and folding efficiency [333, 334], symmetry is not a prerequisite for function and can even be disadvantageous, for example in the case of ribosomes or polymerases where symmetry would counteract the directionality associated with reading nucleotide sequences. The symmetry in viral capsids is also often broken to allow for the insertion of additional monomers and thus an increase of the volume available for storage of genetic material encompassed by the capsid [334] (compare

Section 2.5.3). Hence, our algorithm should not rely on *a priori* symmetry information. Even if a complex is known to be symmetric, the type of symmetry present in the complex is often unknown and the potential number of different symmetries grows with the number of monomers in the complex. Hence, either the type of symmetry must be known beforehand or all possible symmetries must be tried. In addition, the generation of partial complexes might be difficult or impossible with these methods, even though such complexes might still be of use for further studies. Consequently, we decided not to incorporate a restriction regarding symmetry in our algorithm, however, if a near-symmetric complex is generated, the algorithm infers the symmetry from the assembly and optimizes the complexes accordingly. Furthermore, symmetric binding modes can be identified from the provided docking poses.

Our development constitutes a major step forward from previous approaches in terms of both the number of distinct protein types in the assembly and the total number of monomers comprising the complex, which are often only able to assemble complexes with a small number of monomers (typically six for computational reasons), consider only homo-oligomers and often take symmetry information into account, as described in the introductory part of this chapter. However, complexes can be very diverse, as described in Sections 2.3 to 2.5 and the introduction to Chapter 8. Our approach should thus be able to account for this diversity and will be tested on a diverse benchmark set of 308 protein complexes we will derive throughout the remainder of this thesis, yielding symmetric and asymmetric complexes with 6 to 60 monomers, 1 to 15 distinct protein types, and 1-50 different binding modes.

Furthermore, we want to assist in cases where integrative approaches are applied to predicting the structure of protein complexes, and the integration of the various sources of information is performed (at least partially) manually or in a semi-automated fashion [335]. For such application scenarios, we want to provide a fast and automatic algorithm with which the manual intervention is reduced to a minimum and is only required to generate starting dimers.

Summarizing, we can state the aims of our approach as follows:

- Considerably extend the scope of current algorithms to the modeling of oligomeric macromolecular assemblies with many more than six monomers
- Require only a minimum of information for the complex assembly: the protein types and representative high-resolution structure for each type, their respective stoichiometries and docking poses sampling each of the assumed native complex binding modes
- Handle both homo- and heteromeric complexes with a large number of distinct protein types
- Reduce the amount of required manual intervention to data collection and input preparation
- Do not assume a complex symmetry beforehand
- Provide a flexible algorithm that can handle a broad range of diverse complexes in terms of size, composition, topology, (a)symmetry

In the following chapters, we will describe and evaluate how these goals are accomplished: we develop a novel, efficient scoring function, called transformation match

score (*tms*), and 3D-MOSAIC, a combinatorial algorithm that iteratively assembles protein complexes from binary docking data using a greedy scheme based on *tms*. Furthermore, we establish a diverse benchmark set of protein complexes to evaluate the performance of our algorithm.

9 PRELIMINARIES

This chapter defines the concepts used during development of 3D-MOSAIC.

9.1 RIGID TRANSFORMATIONS

To determine the structure of macromolecular oligomeric assemblies, we require a description of the placement, i.e., the position and orientation of each of the complex monomers w.r.t. to a base configuration.

Assuming that the monomer is a rigid body, i.e., no conformational changes occur during placement, the description of its placement in three-dimensional Euclidean space requires two distance-preserving (i.e., isometric), so-called proper *rigid body transformations* [336]: a proper rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ (an orthonormal matrix that does not allow reflections, i.e., $det(\mathbf{R}) = +1$) and a translation $\mathbf{t} \in \mathbb{R}^3$.

These two parameters determine how points in one reference frame are represented in another reference frame: let P, P' be the position of the same point, once measured in the reference frame $\{\mathbf{F}\}$ and once in the reference frame $\{\mathbf{C}\}$, the transformation of P from $\{\mathbf{F}\}$ to P' in $\{\mathbf{C}\}$ is given by [336]:

$$P' = \mathbf{R} \cdot P + \mathbf{t} \tag{9.1}$$

To simplify the notation of proper rigid transformations and combinations thereof, $\mathbf{R}$ and $\mathbf{t}$ can be combined into a single matrix $\mathbf{T} \in \mathbb{R}^{(4,4)}$ using homogeneous coordinates (see [337]): in such matrices, not only the combination of rotations but also translations and thus arbitrary rigid transformations can be performed via a single matrix multiplication, as opposed to the matrix multiplication and vector addition required in the classical formulation. Such a matrix $\mathbf{T}$ has the form:

$$\mathbf{T} = \begin{pmatrix} \mathbf{R}_{xx} & \mathbf{R}_{xy} & \mathbf{R}_{xz} & \mathbf{t}_x \\ \mathbf{R}_{yx} & \mathbf{R}_{yy} & \mathbf{R}_{yz} & \mathbf{t}_y \\ \mathbf{R}_{zx} & \mathbf{R}_{zy} & \mathbf{R}_{zz} & \mathbf{t}_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{9.2}$$

where $\mathbf{R}_{xx}, \mathbf{R}_{xy}, ..., \mathbf{R}_{zz}$ denote the components of $\mathbf{R}$ and $\mathbf{t}_x, ..., \mathbf{t}_z$ the components of $\mathbf{t}$.

When applying a transformation $\mathbf{T}$ to a point P, $\mathbf{T}$ is decomposed again into the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ and can then be applied as defined in Eq. 9.1 (technically, the transformation is applied in four-dimensional projective space, where each point is extended by a fourth coordinate with value 1, however, rigid transformations yield the above result in Euclidean space) [337].

In the following, the term *transformation* always refers to proper rigid transformations as expressed in Eq. 9.2.

9.2 RIGID DOCKING POSES

Typically, in rigid binary protein-protein docking, a docking pose comprises a docking score or (estimated) interaction energy and a transformation describing the placement of one of the binding partners in the bound configuration (the other binding partner is typically kept fixed) relative to its configuration in the unbound state (see Section 9.1). However, when assembling macromolecular oligomeric complexes using pairwise dockings, we require the following additional information for each docking pose to decide which docking poses may be considered during a particular stage of the assembly process.

Firstly, protein complexes often consist of a number of different monomeric proteins, i.e., proteins with different amino-acid sequences. To distinguish between them, we introduce the concept of different protein types: given a complex consisting of proteins with n different amino-acid sequences, let $seq(a)$ be the amino-acid sequence of protein a in the complex. Two proteins a and b are labeled with the same protein type $p \in \{1, ..., n\}$ if and only if $seq(a) = seq(b)$. We denote the set of protein types by $\mathfrak{P} \subseteq \mathbb{N}^+$.

Secondly, two proteins that interact with each other typically can do this in one of a small number of distinct orientations, called binding modes, and a unique identification of distinct binding modes is necessary during the assembly process. In a binding mode, each protein contributes its interface which is the surface patch of the protein that is in contact with its binding partner. The two interfaces are complementary w.r.t. their shapes and biochemical properties. The information on potential binding modes can for example be obtained from studies on correlated mutations [322, 323, 197, 199] or cross-linking experiments [326]. Computationally, such binding modes can be sampled using a docking algorithm. To attribute each docking pose to one of the n unique (supposedly) native binding modes in a complex, we equip each binding mode between a pair of protein types with a unique id $b \in \{1, ..., n\}$. The set of ids corresponding to binding modes occurring in a complex is denoted by $\mathfrak{B}$.

Thirdly, to determine compatible docking poses, we require the artificial concept of directionality of interactions and interfaces. In protein-protein docking, the larger protein is often considered the receptor, yielding a transformation for the smaller protein, the ligand. However, in the context of this thesis, we consider both protein monomers M_1 and M_2 of a docking pose alternatively to be receptors and ligands, regardless of their size.

We thus obtain two transformations $\mathbf{T}_1$ and $\mathbf{T}_2$ with $\mathbf{T}_2^{-1} = \mathbf{T}_1$. $\mathbf{T}_2$ describes the placement of M_2 w.r.t. M_1 and $\mathbf{T}_1$ the placement of M_1 w.r.t. M_2, as shown in Fig. 9.1. Each transformation in a set of docking poses describing the placement of M_2 is labeled with a directionality $+1$, whereas each inverse transformation placing M_1 w.r.t. M_2 is labeled with directionality -1.

Using the discrimination between protein types and binding modes, as well as the directionality of interactions, the interfaces present in a macromolecular oligomeric complex can be uniquely described as follows: given two proteins with types $p_1, p_2 \in \mathfrak{P}$, we denote the directed interface that p_1 provides for the interaction with p_2 in the binding mode represented by id $b \in \mathfrak{B}$ by a tuple $i^+ := (+1, b, p_1, p_2)$. Analogously, the complementary (reverse) interface that p_2 provides for contact with p_1 in the same binding mode b is given as $i^- := (-1, b, p_2, p_1) = -i^+$. The set of directed interfaces present in a complex is denoted by $\mathfrak{I}$.

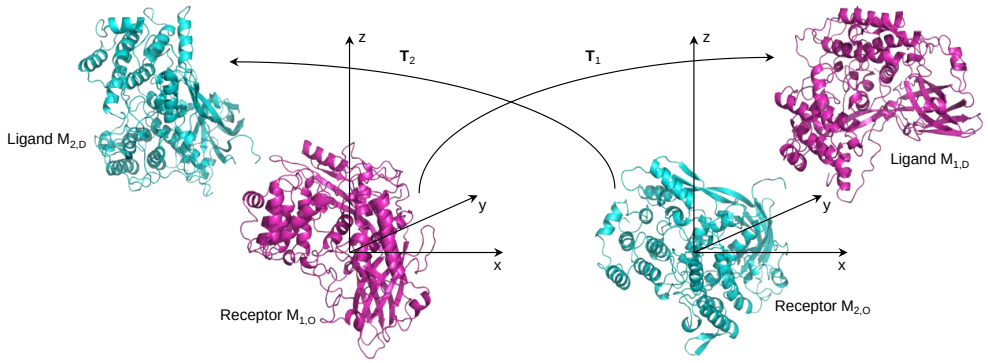


Figure 9.1: The two rigid transformations obtained from a binding mode: Each of the two monomers (M_1, magenta and M_2, cyan) is alternatively considered receptor and ligand. The respective receptor is first superimposed to a copy centered at the origin ($M_{1,O}$ resp. $M_{2,O}$). The transformation of the corresponding ligand in the docking pose ($M_{2,D}$ resp. $M_{1,D}$) is then calculated w.r.t. its counterpart at the origin ($M_{2,O}$ resp. $M_{1,O}$).

Having introduced the above notation of interfaces in a macromolecular oligomeric complex, we define a docking pose as used for the assembly of such a complex as follows:

Definition 9.1 (Docking pose)**.** *A docking pose is represented by an ordered triple* $d := (i, e, \mathbf{T})$ *comprising the following elements: a directed interface* $i \in \mathfrak{I}$*, an interaction energy* $e \in \mathbb{R}$*, and a rigid transformation* $\mathbf{T} \in \mathbb{R}^{(4,4)}$*. The set of docking poses obtained from the docking runs sampling the (assumed) binding modes in a complex is denoted by* $\mathfrak{D}$*.*

The interaction energy e represents the net score for the interaction between the two protein monomers M_1 and M_2 in the docking pose:

$$e = e_{M_1+M_2} - (e_{M_1} + e_{M_2}) \text{ (compare Eq. 2.1)} \tag{9.3}$$

where $e_{M_1+M_2}$ corresponds to the score of the protein monomers M_1 and M_2 in the pose obtained from binary docking and e_{M_1}, e_{M_2} account for the scores of both monomers in their respective unbound states which can be non-zero, depending on the scoring function used during docking.

The use of the net interaction energy e instead of $e_{M_1+M_2}$ is necessary to correctly determine the total interaction energy of a particular complex, the so-called complex energy, which is the sum of the interaction energies of all docking poses used during the assembly of that particular complex. Using $e_{M_1+M_2}$ alone, the unbound-state scores of some monomers might be added multiple times during the assembly of macromolecular oligomeric complexes from pairwise dockings: for example, when two monomers are attached via docking poses to the same monomer, the score of that monomer from the unbound state would be counted twice. This is an undesirable artifact of the assembly procedure, leading to a wrong complex energy as well as potentially to a wrong ranking of the assembled complexes w.r.t. their complex energies. We thus use the net interaction energy e as defined in Eq. 9.3.

For a docking pose $d \in \mathfrak{D}$, the following labels are used in Chapters 10 and 11:

- $I(d) \in \mathfrak{I}$ is the directed interface of d.
- $R(d) \in \mathfrak{P}$ the type of the protein considered as the receptor of d (Fig. 9.1).

- $L(d) \in \mathfrak{P}$ the type of the protein considered as the the ligand of d (Fig. 9.1).
- $E(d) \in \mathbb{R}$ is the interaction energy of d.
- $T(d) \in \mathbb{R}^{(4,4)}$ is the transformation describing the placement of the ligand monomer associated with d.
- $L(i) \in \mathfrak{P}$ the protein type of the ligand of interface $i = I(d)$ of d.

9.3 COMPLEX CANDIDATES

In this section, we introduce the information that is required for assembling protein complexes using binary docking data which we will describe in Section 11.2.

Firstly, each complex consists of one or more protein types, each type occurring at a certain multiplicity. The stoichiometry of a complex associates each type with the respective multiplicity and is represented by a map $S : \mathfrak{P} \to \mathbb{N}^+, p \mapsto n$, assigning each protein type $p \in \mathfrak{P}$ a multiplicity $n \in \mathbb{N}^+$. The complex size K in terms of the number of proteins contained therein is the sum of the multiplicities of the individual protein types:

$$K = \sum_{p \in \mathfrak{P}} S(p) \tag{9.4}$$

Secondly, we require information on the protein type and placement of each monomer in the complex which we can represent as follows:

Definition 9.2 (Complex Monomer)**.** *A complex monomer is represented by an ordered tuple $m := (p, \mathbf{T})$ where $p \in \mathfrak{P}$ denotes a protein type and $\mathbf{T} \in \mathbb{R}^{(4,4)}$ a transformation of the monomer from its unbound state to its bound state in the complex. We denote the set of complex monomers by $\mathfrak{L}$.*

The iterative assembly process we will describe in Section 11.2 yields in each iteration a set of solutions, each solution represented by a so-called complex candidate, which represents a (partial) complex and may be considered in the subsequent iteration of the algorithm. Each new complex candidate is derived from a solution of the previous iteration, i.e., a parent complex candidate p and extends that parent solution by exactly one new (ligand) complex monomer $l \in \mathfrak{L}$ (Def. 9.2); consequently the iterative scheme induces a sequence in monomer attachment and thus a sequence of ancestor complex candidates for each new complex candidate where each ancestor complex candidate represents the attachment of one new complex monomer.

The attachment of complex monomer l can happen to any complex monomer represented by one of the ancestor complex candidates of p (including p itself); the ancestor complex candidate representing the monomer to which l is attached is called receptor complex candidate.

Furthermore, for the purpose of describing the algorithm in Section 11.2, we require the following (redundant) information to be associated with a complex candidate: a unique id, the complex match score (see Sections 10.3 and 11.2.7) and interaction energy (the sum of the interaction energies of all docking poses used for the assembly) of the complex candidate, and the number of symmetry mappings (see Def. 9.8) determined for the complex candidate. Finally, we require information on the number of

steric clashes in the complex, i.e., the number of atoms with a distance less than a certain threshold that indicates a significant penetration of the volumes of the respective atoms, leading to large repulsive forces between both atoms (compare Section B.4).

Definition 9.3 (Complex Candidate)**.** *We denote by a complex candidate an ordered tuple* $c := (i, l, p, r, s, e, c, o)$ *comprising the following elements: a unique id* $i \in \mathbb{N}$*, the ligand complex monomer* $l \in \mathfrak{L}$ *(Def. 9.2), the unique ids* $p, r \in \mathbb{N}$ *of the parent and receptor complex candidates respectively (*0 *only if none is present),* $s, e \in \mathbb{R}$ *the total score (Sect. 10.3) and interaction energy of the complex candidate, respectively, as well as* $c, m \in \mathbb{N}$ *the number of steric clashes and symmetry mappings (Def. 9.8), respectively, in the (sub-)complex represented by* c*.*

Let c be a complex candidate and m the complex monomer represented by c, we use the following labels throughout Chapter 11:

- $M(c) \subseteq \mathfrak{L}$ is the set of complex monomers over c and its ancestor complex candidates.
- $P(c) \subseteq \mathfrak{P}$ contains all protein types $p \in \mathfrak{P}$ whose count over the complex monomers represented by c and its ancestor complex candidates does not exceed their respective allowed stoichiometry $S(p)$.
- $S(c) \in \mathbb{R}$ is the complex match score of c (see Sections 10.3 and 11.2.7).
- $E(c) \in \mathbb{R}$ is the total interaction energy of c.
- $C(c) \in \mathbb{N}$ is the number of clashes between monomers of c.
- $T(m) \in \mathbb{R}^{(4,4)}$ is the transformation describing the placement of m.

9.4 MAPPING COMPLEXES

When considering docking poses or protein complexes assembled by sequentially attaching new monomers w.r.t. such docking poses, the pairwise similarity of some solutions will inevitably be greater than that of others. The consideration of solutions that are similar w.r.t. a certain similarity measure during the course of an algorithm is typically undesirable, because it often leads to an increase in computation and a comparatively low gain in information. Hence, the determination of the similarity of a set of solutions and the removal of the most similar ones to only consider a set of diverse solutions is an important step during the course of an iterative complex assembly algorithm that can avoid becoming stuck in local optima. In addition, in a benchmark scenario where the native reference complex is known, the comparison of an assembled complex with that reference complex based on a similarity measure is useful. Finally, the determination of similarity of a complex to itself under a set of (symmetry) mappings (see Section 9.5) can be used to optimize the complex structure.

We thus need to derive a definition of *complex similarity*. The simplest possibility would be to use the standard RMSD (see Section 3.6.1). Because complexes are composed of several monomers, we can do this using a slightly modified version of the RMSD for two complexes C_1 and C_2, each with m monomers:

$$RMSD_{\mathbf{T}}(C_1, C_2) = \sqrt{\frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n_i} |(\mathbf{R} \cdot x_j^{M_{i,1}} + \mathbf{t}) - x_j^{M_{i,2}}|^2} \tag{9.5}$$

where n_i denotes the number of atoms of the equally-sized monomers $M_{i,1}, M_{i,2}$ of complex C_1 and C_2, respectively. $x_j^{M_{i,k}}$ represents the atomic position of atom j in monomer i of complex C_k. $\mathbf{R}$ and $\mathbf{t}$ denote the rotation and translation given by a transformation $\mathbf{T}$ that can be applied to superimpose C_1 onto C_2, if required.

However, when iteratively assembling protein complexes (see Section 11.2), it may happen that two equal (or similar) protein complexes are generated that differ in the ordering of their complex monomers, which is an artifact resulting from the sequential attachment of new complex monomers during assembly as well as the docking poses used for the attachment (compare Section 9.3).

We thus need to find a mapping between the monomers in two complexes (or a set of mappings) under which the two complexes are similar w.r.t. a given RMSD threshold d_{max} (an example is presented in Figures B.2a and B.2b).

In a first step, we can determine those proteins in two complexes that are mappable onto each other, i.e., those that have the same protein type:

Definition 9.4 (Equivalence-Mapping)**.** *Given two complexes C_1, C_2 each consisting of m proteins. Let $l_{i,1}, l_{i,2} \in \mathfrak{P}$ be the protein type of the i-th protein, $i \in \{1, ..., m\}$, in complex C_1 and C_2 respectively. A bijective function $\varphi : \{1, ..., m\} \to \{1, ..., m\}$ is called equivalence-mapping from the proteins of C_1 onto those of C_2 if $l_{\varphi(i),1} = l_{i,2}\ \forall i \in \{1, ..., m\}$. The complex C_1 with proteins reordered according to φ is denoted by C_φ.*

To determine whether there exists an equivalence mapping under which both complexes reveal structural similarity, we adapt a concept from graph theory, which is related to our problem, called graph isomorphism: two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are isomorphic if there exists a bijective mapping φ between the vertices of both graphs, such that an edge $(u, v) \in E_G$ if and only if $(\varphi(u), \varphi(v)) \in E_H$ [338].

However, we do not have strict adjacency between the proteins: the information on whether two proteins are in contact (or adjacent), e.g., whether a number of atom pairs, one atom per pair from each protein, are closer than a certain distance (and thus interacting), depends on the chosen thresholds for the distance as well as the number of interactions above which the two proteins are considered to be in contact. Two monomers being in contact w.r.t. to such thresholds in one complex might not be in contact in another, even though the overall complex structure is similar.

We thus adapt the notion of graph isomorphism to our problem by replacing the adjacency information by information on the overall complex similarity using the RMSD measure between the complex monomers under a particular mapping:

Definition 9.5 (Complex Similarity Mapping)**.** *Given two complexes C_1, C_2 each consisting of m proteins. An equivalence-mapping φ from the proteins of C_1 onto those of C_2 s.t. a transformation T_φ that optimally superimposes C_φ onto C_2 yields an $RMSD_{T_\varphi}(C_\varphi, C_2) \leq d_{max}$ for a given threshold d_{max} is called a complex similarity mapping. T_φ is called a similarity transformation. Depending on the selected threshold d_{max}, several mappings might exist. If at least one such complex similarity mapping can be found for two complexes C_1 and C_2, we consider them similar under d_{max}.*

9.5 APPROXIMATE COMPLEX SYMMETRY

Symmetry is a concept inherent to many objects in Nature [339] and with small perturbations to the observed symmetry pattern (approximate symmetry) also found in

protein complexes [334]. An object is symmetric if it is invariant under a particular transformation [340, 341].

To determine whether such transformations exist for a particular complex, we adapt the notion of automorphisms, i.e., isomorphisms between a graph and itself [338], from graph theory to our problem, using the notion of complex similarity mappings given in Def. 9.5.

However, depending on the selected RMSD threshold d_{max}, the set of obtained complex similarity mappings might contain elements φ_1, φ_2 with $\varphi_1 \neq \varphi_2$ such that $\varphi_1(i) = \varphi_2(i)$ for at least one $i \in \{1, ..., m\}$ where i corresponds to the i-th of m proteins of the complex under consideration. However, given ideal symmetry, such two mappings for a protein complex cannot exist: first, due to the chirality of the amino acids proteins are composed of, the only symmetry operations for protein assemblies are rotations [342]. Second, because (acyclic) proteins are intrinsically asymmetric, no rotational symmetry operations (except identity) are possible that map a protein onto itself. Consequently, no two mappings φ_1, φ_2 with $\varphi_1 \neq \varphi_2$ with $\varphi_1(i) = \varphi_2(i)$ might exist under ideal symmetry.

However, given approximate symmetry, such two (or more) mappings might indeed be found, depending on the chosen RMSD threshold. Yet, for the above reasons, only one of them can be valid. In such cases, we consider the one with the minimum RMSD the better one and discard the others.

To describe when a set of symmetry mappings is valid, we thus define when two mappings are disjoint:

Definition 9.6 (Disjointness of Complex Similarity Mappings)**.** *Given two complexes C_1, C_2 each with m proteins. Two mappings φ_1 and φ_2 from C_1 to C_2 are disjoint if and only if $\varphi_1(i) \neq \varphi_2(i)$ $\forall i \in \{1, ..., m\}$.*

A valid set of symmetry mappings can then be defined as follows:

Definition 9.7 (Symmetry Mappings)**.** *Let Φ contain all complex similarity mappings w.r.t. a given threshold d_{max} between a given complex C of m proteins and itself, ordered by increasing RMSD. The set of symmetry mappings $S \subseteq \Phi$ is constructed from Φ as follows: for each $\varphi \in \Phi$, φ is added to S if and only if φ is disjoint from all mappings already present in S. The corresponding set of similarity transformations is called symmetry transformation set and is denoted by T_S, with $T_S(\varphi)$ being the symmetry transformation corresponding to mapping φ.*

In the ideal case, the symmetry mappings form a group, however depending on the threshold d_{max} and the geometry of the complex C, the requirement for closure (see [343]) may not be fulfilled. Obviously, the set of symmetry mappings always contains the identity mapping from each monomer in C onto itself, because $RMSD(C, C) = 0$. Consequently, symmetries in a complex can only occur if further symmetry mappings in addition to the trivial one are found. Hence, we define the concept of complex symmetry based on a set of symmetry mappings as follows:

Definition 9.8 (Complex Symmetry)**.** *A complex C is considered to be symmetric if there exists a set S of symmetry mappings for C that contains at least one mapping apart from the trivial one, i.e., $|S| > 1$. Each such mapping $s \in S$ induces a symmetry operation, i.e., a transformation that optimally superimposes the monomers in the complex w.r.t. to the mapping s.*

10 DEVELOPING THE TRANSFORMATION MATCH SCORE

As explained in Sect. 8.1, the aim of this project is the assembly of oligomeric protein complexes on the basis of pairwise docking data. Intuitively, this task can be considered similar to solving a three-dimensional jigsaw puzzle. In the following section, we will address the basic properties both problems have in common and the differences between them.

In the subsequent section, we demonstrate on a set of complexes exemplarily assembled from pairwise docking data, which differences arise in the actual assembly process compared to a jigsaw puzzle, where pitfalls arise and how they can be dealt with.

In the concluding section of this chapter, we will present a novel scoring function that does not rely on scores and ranking of the obtained pairwise dockings, but rather makes use of the mutual compatibility of docking poses w.r.t. the orientation of different monomers in a protein complex.

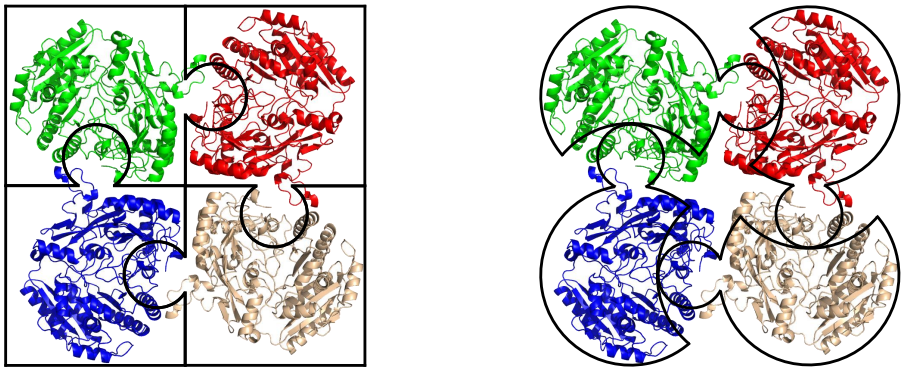
10.1 MACROMOLECULAR COMPLEXES AS THREE-DIMENSIONAL JIGSAW PUZZLES

The process of assembling a complex from its monomers is faintly reminiscent of solving a three-dimensional jigsaw puzzle: the monomers are the pieces to be put together, while the information on the interface locations and binding modes can be considered the equivalent to the tabs and blanks, i.e., the prominent and cut-out areas of the individual pieces of the puzzle, respectively, that are interlocked upon solving the puzzle. In the following, we will use that analogy to demonstrate the relevant properties of macromolecular assemblies.

In the ideal scenario of assembling macromolecular complexes, one would have perfect knowledge of the binding interfaces, corresponding to a puzzle with perfectly fitting pieces (Fig. 10.1a) and the effort of modeling the complex structure is comparable to that of solving a 3D-jigsaw puzzle with as many pieces.

However, in more realistic cases the information on the location of the interfaces is too fuzzy to precisely infer the interlocking of the corresponding pieces. Moreover, the shape and biochemical properties of the binding site(s) might change due to conformational rearrangements during the assembly of the complex. Speaking in terms of a puzzle, this scenario corresponds to roughly fitting pieces, where the tabs and blanks are only crudely complementary, as shown in Fig. 10.1b: due to the poorly defined interfaces, multiple different orientations where the interface areas of the two pieces are in contact can be considered equally valid without further knowledge.

In order to find the true or a near-native binding mode based on this approximate knowledge of the binding site(s), docking methods are commonly applied to extensively sample the assumed interface location(s) of each pair of interacting monomers for low-energy poses. However, this approach also produces many false positive poses which cannot be reliably filtered out using state-of-the-art scoring functions. Owing to the necessity of being fast, scoring functions typically employed in docking algo-



(a) The ideal scenario: The interfaces are well-known, represented by perfectly matching pieces.

(b) The real scenario: The interfaces are only roughly known. The uncertainty of the exact binding mode is expressed by roughly fitting pieces.

Figure 10.1: Quality of available interface information and corresponding representation of puzzle pieces.

rithms are often not very accurate in predicting binding affinities and often fail to rank the near-native docking poses among the top ranks. Clustering and re-scoring of the obtained poses with a computationally more expensive scoring function are often applied as a post-filter step. Such a practice can help to alleviate this problem to some extent and removes the most implausible solutions [344, 345, 346, 347]. However, the remaining number of putative candidates is typically still so large that a near-native solution is unlikely to be found among the top ranks.

We are thus left with an ensemble of dockings per interface, from which we ultimately have to chose a single pose for the attachment of the next puzzle piece. This problem is further aggravated by the following: in contrast to a traditional jigsaw puzzle where each piece is typically unambiguously described by a unique combination of shape and a section of the image to be put together (in a typical puzzle, at least either shape or image section are different), a biological assembly often contains multiple copies of each protein type involved in forming the complex. This means that, in contrast to the jigsaw puzzle where each piece can only be used once, each sampled pose can be used multiple times to assemble the complex.

Let us consider an example: given a homomeric complex with m monomers, we need to establish at least $m-1$ contacts between the monomers to generate a complex in which each monomer is bound to another. If we have d docking poses available to attach two monomers of the same kind, an exhaustive search for the best solution by trying all docking poses and combinations thereof entails $d^{(m-1)}$ distinct solutions. For $m = 11$ and $d = 100$ (a comparatively small number of sampled poses for a roughly known interface), we would have to consider $100^{10} = 10^{20}$ theoretically possible solutions.

Typically, most of the generated solutions exhibit a severe overlap between two or more monomers and are thus implausible. While such solutions can easily be filtered out, the reduced space of solutions is typically still too large to be computationally tractable. Fortunately, a general property of complexes can help to avoid considering the majority of implausible solutions: the fact that every monomer usually establishes connections to several other monomers in the complex, leading to compact assemblies

with a dense network of connections between the individual components [83]. In the following sections, we describe how we can make use of that observation and provide a simple yet effective score for the quality of (sub-)complexes when generating macromolecular assemblies from binary dockings.

10.2 OBSERVATIONS ON ASSEMBLING COMPLEXES FROM PAIRWISE DOCKINGS

To illustrate our score, assume that we are faced with the task to assemble a homomeric complex C with four subunits, i.e., a complex with one protein type $p \in \mathfrak{P}$ with corresponding multiplicity $S(p) = 4$. The simplest way to form an assembly with these criteria is when there exists only one asymmetric binding mode between two instances of p.

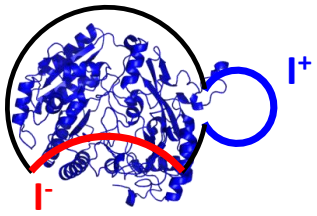


Figure 10.2: Interfaces I^+ (blue) and I^- (red).

For asymmetric binding modes to be established between monomers of the same protein type, the protein must provide two complementary interfaces I^+ and I^- (or at least two different ways of binding to the same interface), as depicted in Fig. 10.2.

According to our assumption from Section 8.1, none of the interfaces is exactly known. Hence, we need to perform a local docking between two instances M_A and M_B of the same protein, where M_A provides I^+ and M_B provides I^- as docking interface (or vice versa). By considering each of the two monomers once as receptor and the respective other as ligand, in analogy to our puzzle example, we obtain two sets of puzzle pieces, denoted by D^+ and D^-, one w.r.t. I^+ and one w.r.t. I^-, respectively, as described in Sect. 9.2. In each of the two sets, the pieces overlap and form a docking pose ensemble, as shown in Fig. 10.3.

Let us now consider a simple example how to iteratively assemble such a complex from an initial, centered monomer M_1: in every iteration $i \in \{2,3,4\}$, we select a docking pose d_i with corresponding transformation $\mathbf{T}_i$ from pose set D^- (from the red interface) and attach a new ligand monomer M_i by $\mathbf{T}_i$ w.r.t. to the orientation of receptor monomer M_{i-1} (the ligand in the previous iteration). This process is repeated until all four monomers have been put into place.

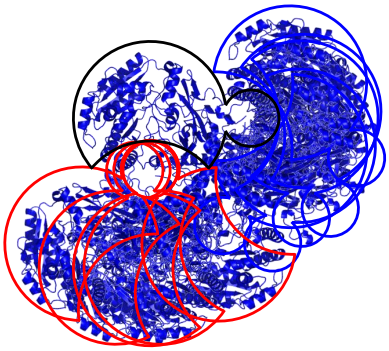


Figure 10.3: Exemplary docking pose sets D^+ (blue) and D^- (red) for I^+ and I^-.

Three exemplary complexes are given in 10.4. Obviously, complexes C2 and C3 establish an additional interaction between M_1 and M_4, and thus should represent energetically (cmp. Section 2.2) more favorable complexes than C1, where this additional contact is not found. Furthermore, complex C3 is more regular than C2 and provides a 4-fold rotational symmetry. Given the observation that many (of the known) complexes in Nature exhibit at least partial symmetries [339] and that symmetry is assumed to be the result of a bias towards low-energy complexes [348, 334, 333], we can assume C3 to be energetically more favorable and thus superior to C2 [333].

An accurate energy function should be able to rank the three complexes according to that observation, i.e., $\Delta G(C3) > \Delta G(C2) > \Delta G(C1)$ (cmp. Section 2.2). However,

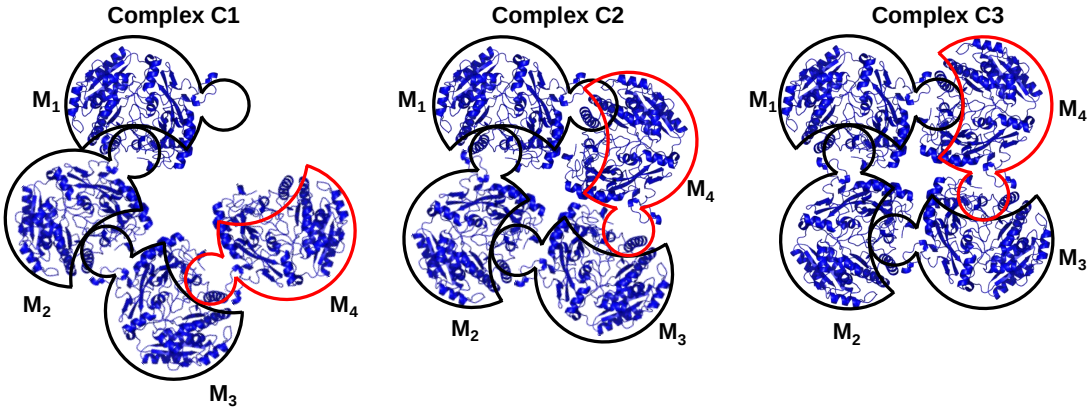


Figure 10.4: Three exemplary complexes, generated by subsequently attaching a new monomer M_i to monomer $M_{i-1}, i \in \{2,3,4\}$ according to one of the docking poses from interface I^-.

accurate all-atom energy calculations are very expensive and not applicable for large complexes, especially when taking into account that an additional energy minimization might be necessary to remove steric clashes introduced by the transformations. Standard docking scoring functions are computationally less intensive but lack the power to discriminate between near-native and decoy poses and thus to determine additional interfaces.

Hence, we need a fast scoring function to determine whether an additional interaction has been established by the iterative assembly of the complexes.

10.3 THE TRANSFORMATION MATCH SCORE

Fig. 10.5 provides a more formal representation of the situation described in the previous section (cmp. Fig. 10.4): in complexes C2 and C3 we observe a ring closure: when monomer M_4 is attached, it comes close to interface I^+ (blue) of M_1. In turn, M_1 is close to the red interface (I^-) of monomer M_4 (cmp Fig. 10.2). It is thus intuitive to search the set of M_1's docking poses at interface I^+ for a pose with a transformation $\mathbf{T}_{Dock}$ that is similar to the transformation $\mathbf{T}_{As}$ of M_4 induced by the assembly, as shown in Fig. 10.5. While $\mathbf{T}_{Dock}$ can be directly obtained from the dockings (for a centered monomer), $\mathbf{T}_{As}$ is the result of cascading transformations that put the individual antecedent monomers in place.

Formally, we can define $\mathbf{T}_{As}$ as follows: given n monomers with monomer M_{i-1} being the receptor of monomer M_i for $i \in \{2,...,n\}$, let $\mathbf{T}_i$ be the transformation of a docking pose $d \in I^-$ that positions the i-th monomer w.r.t. to its receptor. Because the first monomer M_1 in an assembly is always centered at the origin, $\mathbf{T}_1$ is the identity transformation. $\mathbf{T}_{As}$ is then obtained by subsequently applying the transformations on the *transformation path* $P_{\mathbf{T}} := [\mathbf{T}_2, ..., \mathbf{T}_n]$:

$$\mathbf{T}_{As} := \prod_{i=2}^{n} \mathbf{T}_i \tag{10.1}$$

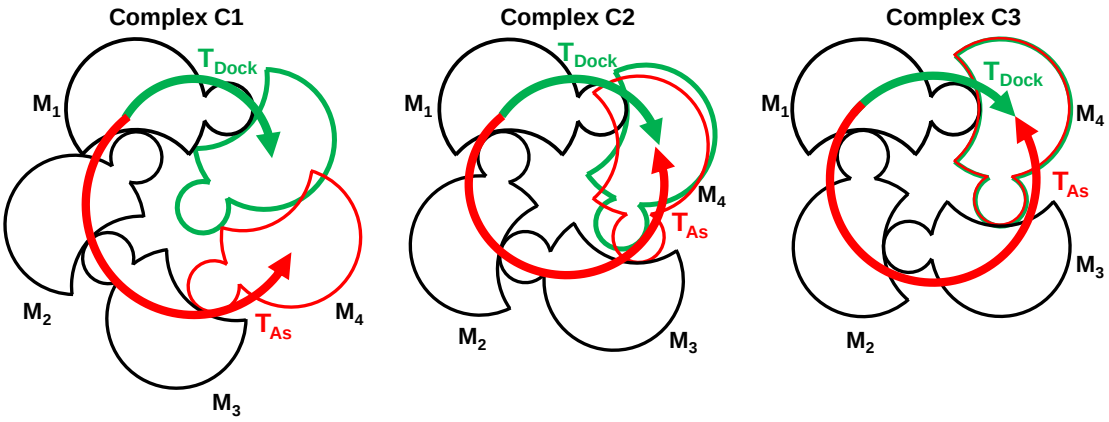


Figure 10.5: The three exemplary complexes from Fig. 10.4, represented as jigsaw puzzle. In each of the three complexes, $\mathbf{T}_{As}$ corresponds to the transformation induced by the assembly for the respective monomer M_4. $\mathbf{T}_{Dock}$ represents the most similar transformation obtained from the dockings at interface I^+ of monomer M_1 in each case.

In the general case, i.e., when considering a monomer M^* which is not centered at the origin, $\mathbf{T}_{As}$ relative to the placement $\mathbf{T}_{M^*}$ of M^* can be computed as:

$$\mathbf{T}_{As} := \mathbf{T}_{M^*}^{-1} \cdot \prod_{i=2}^{n} \mathbf{T}_i \tag{10.2}$$

In the example above, the pairwise docking transformation $\mathbf{T}_{Dock}$ that is most similar to the generated $\mathbf{T}_{As}$ is obtained from a docking pose d_{Dock} from D^+ at interface I^+ of M_1 as follows:

$$\mathbf{T}_{Dock} := T(d_{Dock}) \quad \text{with} \quad d_{Dock} := \arg\max_{d \in D^+} sim(T(d), \mathbf{T}_{As}) \tag{10.3}$$

where $sim : \mathbb{R}^{(4,4)} x \mathbb{R}^{(4,4)} \rightarrow \mathbb{R}$ represents a similarity function between $\mathbf{T}_{Dock}$ and $\mathbf{T}_{As}$ that we will now derive.

Intuitively, one would compare the root-mean-square deviation (see Section 3.6.1) between the placement and the docking pose. However, depending on the number of atoms in the protein, a naïve calculation of the RMSD according to Eq. 3.1 leads to a large number of arithmetic operations to be performed and is thus computationally too expensive.

Eq. 10.3 indicates that the comparison of the corresponding transformations should be sufficient to appropriately determine similar transformations. However, while several general similarity measures for such transformations exist, they are often defined for very specific purposes, yield different results for the same transformations and often lack an intuitive interpretation (an overview of some popular measures is given in *Huyn* [349]). We thus propose two novel, easily interpretable, cutoff-based similarity measures relevant for structural modeling based on rigid transformations.

For the remainder of this section, let $\mathbf{t}^A$ and $\mathbf{t}^B$ be the translations and $\mathbf{R}^A$ and $\mathbf{R}^B$ the rotations given by two rigid transformations $\mathbf{T}_A$ and $\mathbf{T}_B$, respectively.

The first measure we propose is a heuristic measure comparing the translational displacement between two transformations as well as their angular difference. Let l_{max} and a_{max} be the respective maximum thresholds for displacement length and rotation

angle between two transformations. The heuristic similarity measure we propose, S^{da} is defined as:

$$S^{da}(\mathbf{T}_A, \mathbf{T}_B) := \max\left(1 - \frac{|\mathbf{t}^{AB}|}{l_{max}}, 0\right) \cdot \max\left(1 - \frac{\alpha^{AB}}{a_{max}}, 0\right) \tag{10.4}$$

where $\mathbf{t}^{AB}$ is the difference of the translations $\mathbf{t}^B$ and $\mathbf{t}^A$ and α^{AB} denotes the angular deviation between the two rotations $\mathbf{R}^A$ and $\mathbf{R}^B$, given by [350]:

$$\alpha^{AB} = \arccos\left(\frac{tr((\mathbf{R}^A)^{-1}\mathbf{R}^B) - 1}{2}\right) \tag{10.5}$$

where $tr(M)$ denotes the trace, i.e. the, sum of the diagonal elements of a matrix M.

With these two thresholds, it is easily possible to decouple the influence of the angular and translational deviation between two transformations when looking for matching transformations and adapt the parameters individually to the complex to be assembled, if required. While default parameters should already be applicable in most cases, different applications for such a decoupling are thinkable: for example, depending on the size and shape of the monomers as well as the complex topology, angular deviations between matching transformations might be more tolerable than translational differences or vice versa. Furthermore, if additional information on the sampling parameters of the docking algorithm used to generate the transformations is available, e.g., if the rotational degrees of freedom are sampled more densely than the translational ones, the allowed maximum angular deviation a_{max} could, e.g., be set to smaller values while l_{max}, the cut-off for the translational distance, might require larger values to determine matching transformations.

Later, we could show that the RMSD of a protein P under two rigid transformations can be calculated in constant time, solely based on the two transformations and the covariance matrix $cov(P)$ of the protein's atomic positions. The proof and the results of the corresponding benchmark simulations we performed can be found in our related work [351]. The closed formula for the constant-time RMSD measure is as follows:

$$RMSD(\mathbf{T}_A, \mathbf{T}_B) = \sqrt{|\mathbf{t}^{AB}|^2 + \frac{1}{n} tr(\mathbf{R}^{AB} \cdot cov(P))} \tag{10.6}$$

with n being the number of atoms in protein P. Let $rmsd_{max}$ be the threshold for the RMSD between the two transformations $\mathbf{T}_A$ and $\mathbf{T}_B$, we define the RMSD-based transformation match score as follows:

$$S^{rmsd}(\mathbf{T}_A, \mathbf{T}_B) := \left(\max\left(1 - \frac{RMSD(\mathbf{T}_A, \mathbf{T}_B)}{rmsd_{max}}, 0\right)\right)^2 \tag{10.7}$$

Both scoring functions return a similarity score of 1 for identical transformations and yield a score of 0 obtained when any of the cutoffs is reached or exceeded. While S^{da} decreases linearly w.r.t. the rotational and angular deviation, S^{rmsd} decreases harmonically w.r.t. the RMSD. An exemplary decay of the RMSD-based score and the

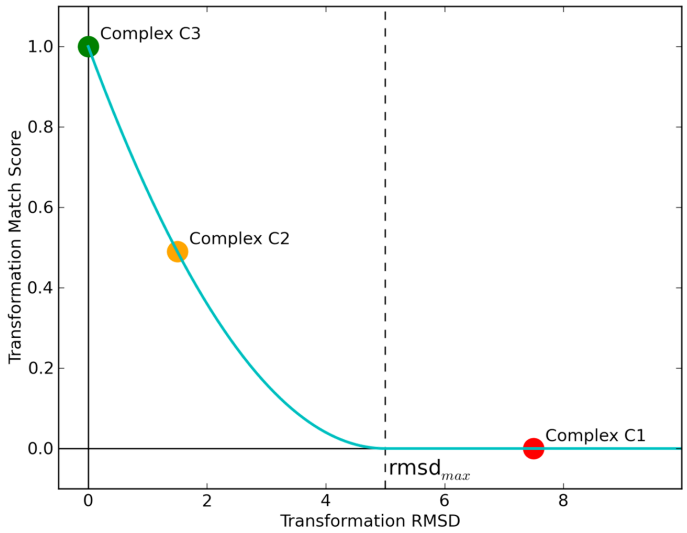


Figure 10.6: An exemplary decay of the RMSD-based transformation matching score and the values of the complexes C1, C2, and C3. $rmsd_{max}$ is set to 5Å. C3 achieves an optimal score of 1 because an identical transformation could be identified in the set of docking poses (cmp. Fig. 10.5). For C2, a roughly matching transformation was found, corresponding to a reduced score. The best transformation found for C1 achieves an RMSD that is beyond $rmsd_{max}$ and thus yields a score of 0.

corresponding values for the three complexes C1, C2, and C3 with an (arbitrarily chosen) cutoff of 5Å for $rmsd_{max}$ is shown in Fig. 10.6.

Contrary to the heuristic deviation/angle-based measure, the RMSD-based score is an exact expression of protein distances, even for proteins with principal axes of different length, and uses an established measure to describe the deviation between two poses. In contrast, the angle-/displacement-based score provides the possibility to independently adjust the impact of angular and translational deviation. Because it does not account for a rotation axis and instead implements only an angular deviation cutoff, it disregards the actual shape of the protein subject to the transformation. Both scores are implemented and will be discussed in the Results.

11

ALGORITHMIC MODELING OF OLIGOMERIC PROTEIN ASSEMBLIES FROM BINARY DOCKING DATA

In the previous chapter, we have derived two similarity measures to score matching transformations obtained when assembling protein complexes from pairwise docking data. In the following sections, we will now present how these measures can be used to solve the problem of assembling a macromolecular oligomeric complex of K proteins from binary dockings.

Based on the transformation matching score, we first formally define our assembly problem in terms of an Integer Quadratic Program which describes theoretically how oligomeric protein complexes can be assembled. However, we will demonstrate several challenges for setting up and applying the IQP formulation in practice.

Consequently, in the concluding section of this chapter, we will present our heuristic, greedy, iterative assembly algorithm that is able to assemble a large variety of different protein complexes. The algorithm is called 3D-MOSAIC (3-Dimensional Modeling of Oligomeric Structural Assemblies based on pairwise Interaction Combination).

11.1 AN INTEGER QUADRATIC PROGRAM FORMULATION OF THE COMPLEX ASSEMBLY PROBLEM

To formally describe the problem of assembling macromolecular oligomeric assemblies from pairwise dockings, we can represent it as a discrete optimization problem in the form of an Integer Quadratic Program (IQP) [352]. The aim of our IQP is to determine a complex of size K, i.e., containing K monomers in total, that yields a maximal overall transformation match score of the involved monomers w.r.t. the underlying set of pairwise docking poses $\mathfrak{D}$ from which the complex is assembled. In the following, we describe the prerequisites and the definition of an IQP to solve the the complex assembly problem.

11.1.1 *Prerequisites*

In this section, we formally define the representation and properties of the monomer placements that can be obtained for a complex of size K on the basis of a set $\mathfrak{D}$ of docking poses and which are required for the IQP.

11.1.1.1 *Monomer Placements*

Each pairwise docking pose $d \in \mathfrak{D}$ with $d = (i, e, \mathbf{T})$ (Def. 9.1) w.r.t. to the directed interface $i = (z, b, p_1, p_2)$ (see Section 9.2) describes the orientation of a ligand protein of type $p_2 \in \mathfrak{P}$ w.r.t. to a receptor protein of type $p_1 \in \mathfrak{P}$ in a binding mode labeled with id $b \in \mathfrak{B}$, according to transformation $\mathbf{T}$.

The assembly of complexes using such pairwise docking poses requires the placement of an initial monomer which does not depend on any docking pose. The placement of such a monomer can be described through introduction of additional artificial

start docking poses (cmp. Def. 9.1), one per protein type $p \in \mathfrak{P}$. Such poses do not correspond to a complex binding mode ($b = 0$), require no receptor protein type ($r = 0$), no direction ($z = 0$), no interaction energy ($e = 0$) and no explicit transformation ($\mathbf{T} = I$). These poses comprise the identity set $\mathfrak{D}^0$:

$$\mathfrak{D}^0 := \{((0,0,0,p),0,I) | p \in \mathfrak{P}\} \tag{11.1}$$

The full set of docking poses is then given by:

$$\mathfrak{D}^* := \mathfrak{D}^0 \cup \mathfrak{D} \tag{11.2}$$

For a complex comprising K monomers, a product of $k \in \{1, ..., K\}$ transformations associated with k sequentially chosen docking poses $d_1, ..., d_k \in \mathfrak{D}^*$ (each docking pose can be chosen multiple times) represents a so-called k-combination of transformations. Such a k-combination is considered a valid monomer placement M, if the following restriction are fulfilled:

1. The first pose d_1 used for a particular k-combination must be from the set of initial poses, i.e., $d_1 \in \mathfrak{D}^0$.
2. The protein type of the receptor of d_i, the i-th pose in the k-combination, must be equal to the protein type of the ligand of pose d_{i-1}, $i \in \{2, ..., k\}$, i.e. $R(d_i) = L(d_{i-1})$ (Sect. 9.2).
3. When considering distinct interfaces, the interfaces $I(d_{i-1}), I(d_i) \in \mathfrak{I}$ (Sect. 9.2) associated with two sequentially chosen poses $d_{i-1}, d_i, i \in \{2, ..., k\}$ for the k-combination must satisfy $I(d_i) \neq -I(d_{i-1})$ (Sect. 9.2). That is, no pose d_i may be used whose associated directed interface is equal to the reverse of the interface associated with d_{i-1}. By choosing pose d_{i-1}, the $i-1$-th monomer is placed, consequently the corresponding interface of this monomer, which is given by $-I(d_{i-1})$ is considered occupied. Because it is occupied, a pose d_i must be associated with a different interface, hence $I(d_i) \neq -I(d_{i-1})$.

The set of valid monomer placements w.r.t. $\mathfrak{D}^*$ is called $\mathfrak{M}$.

11.1.1.2 *Information Associated with Monomer Placements*

The definition of constraints for our IQP requires additional information on the properties of the placements contained in $\mathfrak{M}$. Hence, we assume that each monomer placement $M_i \in \mathfrak{M}, i \in \{1, ..., |\mathfrak{M}|\}$ has the following labels:

- $T(i) \in \mathbb{R}^{(4,4)}$, the transformation of M_i, i.e., the product of the transformations associated with the individual docking poses selected for k-combination M_i (cmp. Eq. 10.1).
- $Protein(i) \in \mathfrak{P}$, the protein type of the monomer which is placed by k-combination M_i.
- $Parent(i) \in \{0, ..., |\mathfrak{M}|\}$, the index of the $(k-1)$-combination which yields k-combination M_i as $T(M_i) = T(M_{Parent(i)}) \cdot \mathbf{T}$, where $\mathbf{T}$ denotes the transformation associated with a docking pose selected from $\mathfrak{D}^*$. The parent is 0 if and only if $k = 1$, i.e., if M_i corresponds to the placement of an initial monomer.
- $Interface(i) \in \mathfrak{I}$, the interface at M_i's parent $M_{Parent(i)}$ via which the most recent docking pose is attached (compare Section 9.2).

11.1.1.3 *Compatibility of Monomer Placements*

Not all pairs of monomer placements are necessarily compatible with each other: the proteins corresponding to these placements might overlap significantly, leading to severe steric clashes. Such pairs of placements will not lead to a plausible solution. We thus assume that the information on clashing pairs of monomer placements is represented by a matrix $\mathbf{C} \in \mathbb{N}^{(|\mathfrak{M}|,|\mathfrak{M}|)}$, where each entry $\mathbf{C}_{i,j}$ denotes the number of clashing atoms for the proteins $Protein(i)$ and $Protein(j)$ w.r.t. monomer placements $M_i, M_j \in \mathfrak{M}$, where $\mathfrak{M}$ denotes the set of valid monomer placements as described in Section 11.1.1.1.

Similarly, we require the transformation match score for any pair M_i, M_j with $Protein(i) = Protein(j)$. To this end, let $sim(\mathbf{T}_1, \mathbf{T}_2)$ denote one of the transformation match scores between two transformations $\mathbf{T}_1, \mathbf{T}_2$, as presented in the previous chapter, i.e., $S^{da}(\mathbf{T}_1, \mathbf{T}_2)$ (Eq. 10.4) or $S^{rmsd}(\mathbf{T}_1, \mathbf{T}_2)$ (Eq. 10.7). We assume that the pairwise transformation match scores for any two monomer placements $M_i, M_j \in \mathfrak{M}$ w.r.t. the transformations $T(i), T(j)$ are given by a matrix $\mathbf{S}$ as follows:

$$\mathbf{S}_{i,j} = \begin{cases} sim(T(i), T(j)) & \text{if } Protein(i) = Protein(j) \\ -1 & \text{otherwise} \end{cases} \tag{11.3}$$

Cases with $Protein(i) \neq Protein(j)$ are never considered in practice, because only the matching of transformations corresponding to the same protein type is reasonable, hence $\mathbf{S}_{i,j} = -1$ in such cases.

11.1.2 *Representation as an Integer Quadratic Program*

For our IQP, we require two different kinds of indicator variables. The first one, $b_i \in \{0,1\}, i \in \{1, ..., |\mathfrak{M}|\}$, indicates whether the monomer represented by $M_i \in \mathfrak{M}$ is built, i.e., has been selected to be considered part of the complex.

The number of instances per protein id $p \in \mathfrak{P}$ in the complex is given by its stoichiometry $S(p)$, thus we must guarantee that only the allowed number of instances per protein id is built:

$$\sum_{i=1}^{|\mathfrak{M}|} b_i \cdot \delta_{Protein(i),p} = S(p) \quad \forall p \in \mathfrak{P} \tag{11.4}$$

Moreover, all built monomers in the complex must be connected: for each monomer there must be a second monomer in the complex that acts as a parent for the first monomer. Only one built monomer acts as the initial one and has no parent:

$$\sum_{i=1}^{|\mathfrak{M}|} b_i \cdot \delta_{0,Parent(i)} = 1 \tag{11.5}$$

All other built monomers must have another built monomer as a parent:

$$\sum_{i=1}^{|\mathfrak{M}|} \sum_{j=1}^{|\mathfrak{M}|} b_i \cdot b_j \cdot \delta_{i,Parent(j)} = K - 1 \tag{11.6}$$

Furthermore we do not allow severe clashes between pairs of built monomers. Given an upper threshold N on the allowed number of clashing atoms, we obtain:

$$b_i \cdot b_j \cdot \mathbf{C}_{i,j} \leq N \quad \forall i,j \in \{1,...,|\mathfrak{M}|\} \tag{11.7}$$

Each set of b_i fulfilling the above conditions is considered a potential complex solution.

For any particular $M_i \in \mathfrak{M}$ in this candidate solution, we now want to find docking poses that, when applied w.r.t. the orientation of other monomers in the complex candidate, produce monomers M_i' that result in a similar transformation to that of M_i. The greater the similarity, the greater the support for the hypothesis that M_i is a good choice in the context of the surrounding monomers. Because $\mathfrak{M}$ contains all possible valid combinations of docking poses from $\mathfrak{D}^*$ for a complex of size K, M_i' must also be part of $\mathfrak{M}$.

To indicate whether we use a particular monomer M_j to match monomer $M_i, i, j \in \{1,...,|\mathfrak{M}|\}$, we introduce a second set of indicator variables $m_{i,j} \in \{0,1\}$.

Finding the maximal matching between monomers can then be formulated as a maximization problem of the following form:

$$\max_{b,m} \sum_{i=1}^{|\mathfrak{M}|} b_i \cdot \sum_{j=1}^{|\mathfrak{M}|} m_{i,j} \cdot \mathbf{S}_{i,j} \tag{11.8}$$

To guide the use of matching monomers, we have to impose some additional constraints on the matching monomers. In the remainder of this section, let h, i, j be $\in \{1,...,|\mathfrak{M}|\}$. A monomer i can be matched by at most K other monomers in a complex of size K:

$$\sum_{j=1}^{|\mathfrak{M}|} m_{i,j} \leq K \quad \forall i \in \{1,...,|\mathfrak{M}|\} \tag{11.9}$$

In particular, the monomer optimally matching a built monomer i w.r.t. its parent is monomer i itself, thus we have:

$$m_{i,i} = 1 \quad \forall i \in \{1,...,|\mathfrak{M}|\} \tag{11.10}$$

In all other cases, when a monomer j is built, it must not be used to match another built monomer $i \neq j$ (if $b_j = 1$, $m_{i,j}$ must equal zero, otherwise it can assume 0 or 1):

$$m_{i,j} \leq 1 - b_j \quad \forall i,j \in \{1,...,|\mathfrak{M}|\}, i \neq j \tag{11.11}$$

Each monomer j may be used at most once to match another monomer:

$$\sum_{i=1}^{|\mathfrak{M}|} m_{i,j} \leq 1 \quad \forall j \in \{1,...,|\mathfrak{M}|\} \tag{11.12}$$

If a monomer j is used to match a built monomer i, i and j have to be of the same kind of protein (trivial for the above equation).

$$m_{i,j} \cdot \left(1 - \delta_{Protein(i),Protein(j)}\right) = 0 \quad \forall i,j \in \{1,...,|\mathfrak{M}|\} \tag{11.13}$$

Furthermore, j must have a parent monomer h among the built monomers:

$$m_{i,j} \cdot \left(1 - \sum_{h=1}^{|\mathfrak{M}|} b_h \cdot \delta_{h,Parent(j)}\right) = 0 \quad \forall i,j \in \{1,...,|\mathfrak{M}|\} \tag{11.14}$$

Any two monomers i, j with $i \neq j$ used to match the same monomer h must originate from different parents:

$$m_{h,i} \cdot m_{h,j} \cdot \delta_{Parent(i),Parent(j)} = 0 \quad \forall h,i,j \in \{1,...,|\mathfrak{M}|\}, i \neq j \tag{11.15}$$

Furthermore, they must be obtained via assembly from different interfaces (in the distinct interface case):

$$m_{h,i} \cdot m_{h,j} \cdot \delta_{Interface(i),Interface(j)} = 0 \quad \forall h,i,j \in \{1,...,|\mathfrak{M}|\}, i \neq j \tag{11.16}$$

Summarizing, the transformation match score optimization problem can then be written as an Integer Quadratic Program of the form:

$$\begin{array}{lllll}
 & & & & \text{Constraints} \\
\max\limits_{b,m} & \sum\limits_{i=1}^{|\mathfrak{M}|} b_i \cdot \sum\limits_{j=1}^{|\mathfrak{M}|} m_{i,j} \cdot S_{i,j} & & & \\
\text{s.t} & \sum\limits_{i=1}^{|\mathfrak{M}|} b_i \cdot \delta_{Protein(i),p} = S(p) & \forall p \in \mathfrak{P} & |\mathfrak{P}| \\
 & \sum\limits_{i=1}^{|\mathfrak{M}|} b_i \cdot \delta_{0,Parent(i)} = 1 & & 1 \\
 & \sum\limits_{i=1}^{|\mathfrak{M}|} \sum\limits_{j=1}^{|\mathfrak{M}|} b_i \cdot b_j \cdot \delta_{i,Parent(j)} = K - 1 & & 1 \\
 & b_i \cdot b_j \cdot C_{i,j} \leq N & \forall i,j \in \{1,...,|\mathfrak{M}|\} & |\mathfrak{M}|^2 \\
 & \sum\limits_{j=1}^{|\mathfrak{M}|} m_{i,j} \leq K & \forall i \in \{1,...,|\mathfrak{M}|\} & |\mathfrak{M}| \\
 & m_{i,i} = 1 & \forall i \in \{1,...,|\mathfrak{M}|\} & |\mathfrak{M}| \\
 & m_{i,j} \leq 1 - b_j & \forall i,j \in \{1,...,|\mathfrak{M}|\}, i \neq j & |\mathfrak{M}|^2 - |\mathfrak{M}| \\
 & \sum\limits_{i=1}^{|\mathfrak{M}|} m_{i,j} \leq 1 & \forall j \in \{1,...,|\mathfrak{M}|\} & |\mathfrak{M}| \\
 & m_{i,j} \cdot \left(1 - \delta_{Protein(i),Protein(j)}\right) = 0 & \forall i,j \in \{1,...,|\mathfrak{M}|\} & |\mathfrak{M}|^2 \\
 & m_{i,j} \cdot \left(1 - \sum\limits_{h=1}^{|\mathfrak{M}|} b_h \cdot \delta_{h,Parent(j)}\right) = 0 & \forall i,j \in \{1,...,|\mathfrak{M}|\} & |\mathfrak{M}|^2 \\
 & m_{h,i} \cdot m_{h,j} \cdot \delta_{Parent(i),Parent(j)} = 0 & \forall h,i,j \in \{1,...,|\mathfrak{M}|\}, i \neq j & |\mathfrak{M}|^3 - |\mathfrak{M}|^2 \\
 & m_{h,i} \cdot m_{h,j} \cdot \delta_{Interface(i),Interface(j)} = 0 & \forall h,i,j \in \{1,...,|\mathfrak{M}|\}, i \neq j & |\mathfrak{M}|^3 - |\mathfrak{M}|^2
\end{array}$$

In the above formulation, the number of indicator variables amounts to $|\mathfrak{M}|^2 + |\mathfrak{M}|$ ($m_{i,j}$ with $i,j \in \{1,...,|\mathfrak{M}|\}$ and b_i with $i \in \{1,...,|\mathfrak{M}|\}$). Furthermore, we require a grand total of constraints of $2|\mathfrak{M}|^3 + 2|\mathfrak{M}|^2 + 2|\mathfrak{M}| + |\mathfrak{P}| + 2$. In the worst case, $|\mathfrak{M}|$ itself is exponential in the number K of monomers in the complex, i.e., $|\mathfrak{M}| = \sum_{k=1}^{K} |\mathfrak{D}|^{(k-1)}$ ($\mathfrak{M}$ contains all possible k-combinations of transformations). We thus see that an exact solution of the complex assembly problem using the above IQP is impracticable even for small K and $|\mathfrak{D}|$.

11.2 3D-MOSAIC: A HEURISTIC ALGORITHM TO SOLVE THE COMPLEX ASSEMBLY PROBLEM

In the previous section, we have presented a formal definition of our complex assembly problem in an integer quadratic program (IQP) representation. However, we have seen that solving the IQP is impracticable even for complexes with a small number of monomers and small sets of pairwise docking poses.

Algorithmically, an exhaustive search for the solution maximizing the complex match score, i.e., the sum of transformation match scores obtained for all monomers (Section 10.3), would require the construction of a tree that describes all possible monomer placements w.r.t. the used set of pairwise docking poses, because an estimation of the solution maximizing the complex match score from a sub-tree is impossible: for example, sub-complexes with a maximal score may become invalid when the last monomer to be attached produces severe steric clashes. For sub-complexes with a non-maximal score, attaching the last monomer may greatly increase the complex match score such that all previously maximal solutions are outperformed. Consequently, sub-solutions and their scores cannot be used to reliably estimate the globally optimal structure and the full combinatorial space must be explored to find the solution maximizing the complex match score. This task is computationally not tractable, even for small complex sizes and numbers of docking poses (cmp. Sect. 10.1).

As a practical, albeit inexact alternative, we employ a heuristic tree-based greedy strategy implementing the transformation match score presented in Section 10.3 to assemble oligomeric complexes from monomeric building blocks. The algorithm is called 3D-MOSAIC (**3**-**D**imensional **M**odeling of **O**ligomeric **S**tructural **A**ssemblies by efficient pairwise **I**nteraction **C**ombination).

Like the IQP, the only types of information the algorithm requires are high-resolution representative structures and the stoichiometry of the individual protein types involved as well as the pairwise dockings corresponding to the assumed native binding modes between the components of the complex. Prior symmetry information is not required and is inferred in the course of the algorithm.

In the first subsection, we present a short, intuitive summary of 3D-MOSAIC and present an exemplary assembly tree to illustrate the top-level algorithm. Susequently, we address the algorithm in more detail, especially the assembly, generation of candidate solutions through monomer attachment, the complex candidate scoring, as well as the post-processing including clustering and symmetry optimization. Several options have been implemented to adapt the algorithm to different use cases. They are shortly presented in Sect. B.13; here we describe the default workflow of the algorithm.

11.2.1 *Algorithm Outline*

3D-MOSAIC requires the 3D structures of all monomers of the complex and information on their stoichiometry. In addition, pairwise docking poses for each pair of interacting proteins, grouped w.r.t. the corresponding binding mode, must be provided. Using these poses, 3D-MOSAIC iteratively assembles protein complexes as follows.

Starting from a core monomer with a maximum number of interfaces, in each iteration, new child complex candidates, each extended by one monomer, are generated. Each monomer in an ancestor solution can act as a receptor in subsequent iterations,

as long as it still provides unoccupied interfaces. For potential ligand protein types that have not exceeded their stoichiometry in the complex, a new monomer can be attached to a receptor provided that i) a docking pose of the ligand with the receptor is available, ii) the two proteins interact across an interface identical to an unoccupied one in the complex, and iii) no significant steric clashes to other monomers in the parent complex candidate are found.

Subsequently, it is investigated whether additional interfaces to the ancestor monomers in the parent complex candidate are established: using the transformation match score, the ancestor interfaces are queried for a docking pose that best matches the placement of the new ligand monomer. The total score over all such poses (complex match score) provides the main ranking criterion for the new set of solutions, followed by a ranking w.r.t. complex energy for solutions with equal complex match score. A subsequent clustering of complex candidates ensures a diverse solution set for the next iteration. After the final iteration, a symmetry optimization is applied if possible.

An illustration of 3D-MOSAIC and an exemplary assembly can be found in Fig. 11.1.

11.2.2 *Preliminary Remarks*

In the following subsections, we will present details on individual parts of the algorithm that are of particular importance for this work. Details on algorithms of secondary importance (e.g., those used for clustering and clash checking) can be found in Appendix B. The notation used throughout the remainder of this chapter as well as in the presented pseudocode follows the definitions given in Chapter 9. Non-trivial capitalized functions are either tagged with a link to the corresponding pseudocode or, where no pseudocode is given, to the corresponding section explaining the respective algorithm.

The pseudocode presented in the following subsections does not necessarily comply with the real implementation in terms of the order of instructions (which can for example be different to achieve runtime speed-ups) nor is it complete w.r.t. the different possible use cases to which the algorithm can be applied. They rather represent a simplified description of the steps performed by 3D-MOSAIC to assemble macromolecular oligomeric complexes.

Finally, for the sake of simplicity, we also assume all essential data (options, protein and complex representations, etc.) to be globally accessible throughout the algorithm.

11.2.3 *Initialization*

Before the actual assembly can start, several initialization steps have to be performed. First, the algorithm options, protein structures $A(p)$ (A for atom set) corresponding to the individual protein types $p \in \mathfrak{P}$ of the complex, their stoichiometries S and the docking poses $\mathfrak{D}$ from which the complexes will be assembled must be retrieved. If a reference complex against which to evaluate the generated complexes and an alignment of the reference chains against the monomers used for the assembly are given, they must also be available.

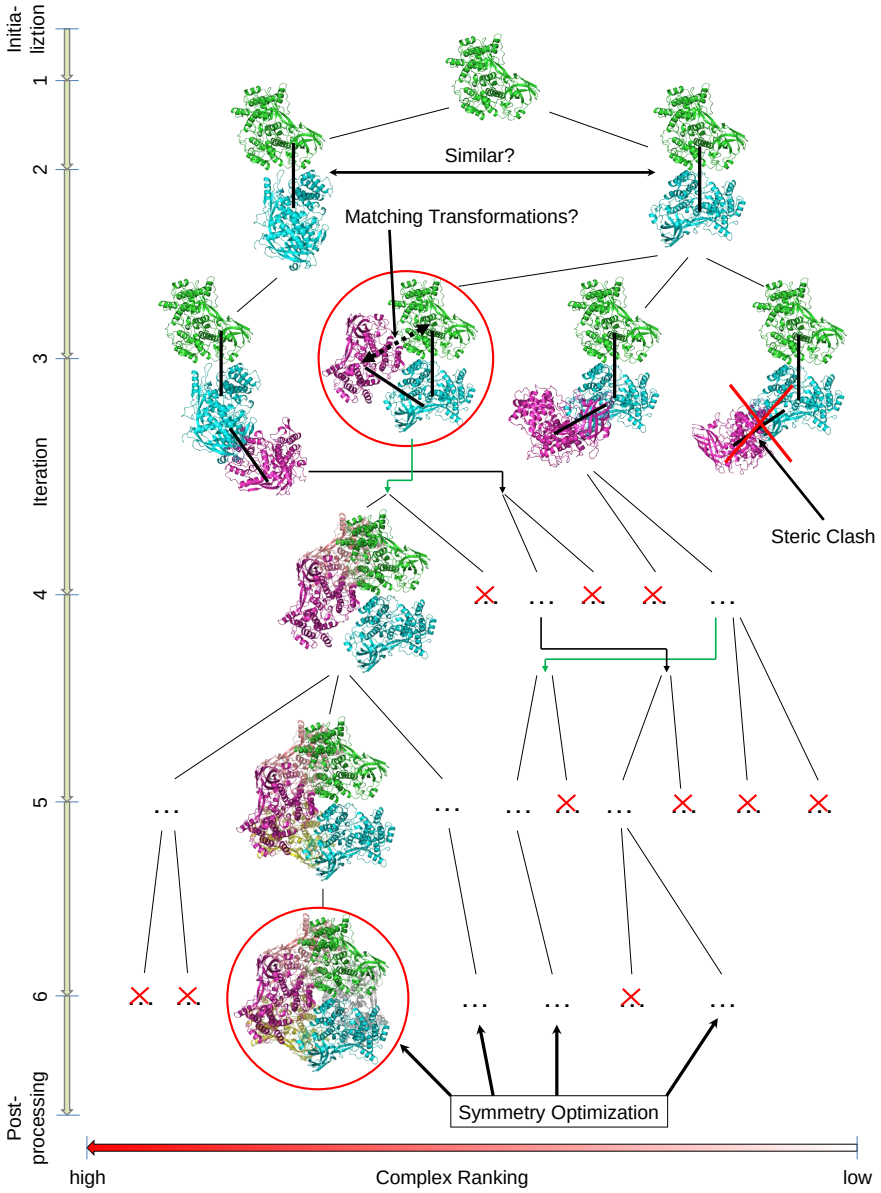


Figure 11.1: Illustration of 3D-MOSAIC on an exemplary assembly of the homo-hexameric hemocyanin from the California Spiny Lobster (*panulirus interruptus*, pdb code 1hcy). In each iteration, new monomers can be attached to all previously retained solutions. If a matching interface is found, the complex score might increase and the corresponding complex might be ranked further up in the list of solutions (green double-tilted arrows). Solutions similar to better-ranked ones or yielding severe steric clashes are discarded. After complex construction, a symmetry optimization can be performed.

The set of poses $\mathfrak{D}$ is then split into subsets w.r.t. the interfaces with which the poses are associated: we obtain a map D, with $D(i)$ providing the set of docking poses associated with interface $i \in \mathfrak{I}$.

Subsequently, D is used to detect interfaces representing symmetric binding modes. For each such mode, asymmetric poses are disabled and the corresponding two interfaces are implicitly treated as a single one (Section B.2). Furthermore $\mathfrak{I}$ is used to determine whether the underlying complex forms a mono-layered ring, because here, a transformation matching is only allowed when the last monomer is attached (Sec-

tion B.3 respectively). If a mono-layered ring is found, a boolean variable t is set to FALSE, indicating that transformation matching is not to be perfomed until the final iteration, otherwise to TRUE.

Protein representations for steric clash checking, complex matching, and, if requested, evaluation against a reference complex are generated (Sections B.4, B.8, B.11). In addition, the maps for the determination of best-matching poses (Sect. B.5) are filled with all enabled docking poses, one map for each interface $i \in \mathfrak{I}$ of each pair of interacting protein types.

11.2.4 *Iterative Assembly*

The iterative assembly process is described in Alg. 11.1. It serves as a wrapper for the repeated extension of complex candidates obtained from the previous iteration. The data it requires are the map D containing the docking poses grouped w.r.t. the corresponding interfaces $\mathfrak{I}$, the protein types $\mathfrak{P}$ with the map S of corresponding stoichiometries, as well as the boolean variable t indicating whether transformation matching is enabled.

First, the final complex size K must be calculated: it corresponds to the number of monomers in the complex and can be determined as the sum over the individual monomer stoichiometries $S(p), p \in \mathfrak{P}$. The level variable k corresponds to the k-th monomer the algorithm is about to attach, while counter *id* denotes the unique ID of the next complex candidate to be generated. Both are initialized with 1.

In addition, the following basic data structures are required: a complex candidate tree T (intially NIL) as well as an initially empty set C of complex candidates. The tree T will store information about the relationships between complex candidates as well as about the match scores for the given docking poses (Sect. B.6); furthermore, it keeps track of which poses are enabled or disabled and which interfaces of each complex candidate are locked for attachment.

In case the assembly is to be picked up at a certain checkpoint, i.e., a set of complex candidates representing sub-complexes produced by a previous 3D-MOSAIC run, a restart file F must be given (Section B.12): the tree T with corresponding relationships between complex candidates as well as the so-far determined best match scores for each docking pose is then retrieved from F. Furthermore, both the set of complex candidates C obtained in the most recent iteration before the file was written $(k-1)$ and the next unique *id* are loaded from F.

If no restart file is given, additional data must be initialized: the core protein type p^0 is determined as the receptor protein type with a maximum number of interfaces. From p^0, an initial monomer $m^0 \leftarrow (p^0, I)$ is created using the identity transformation I (which places a protein instance $A(p^0)$ at the origin, since all proteins $A(p), p \in \mathfrak{P}$ are centered).

m^0 in turn is used to initialize the root complex candidate $c^0 \leftarrow (0, m^0, 0, 0, 0, 0, 0, 0)$ which is then inserted into the initial complex candidate set $C \leftarrow \{c^0\}$. T is then initialized from this initial set and the initial match score of each docking pose is set to zero.

Now, the actual iterative assembly can start: as long as the full complex size K is not reached, i.e., level $k < K$, a new tree level is populated with complex candidates, yielding a set of child candidates C' (Sect. 11.2.5) from the parent generation C. The finalization phase of the current tree level (Sect. 11.2.8) entails the following two steps:

Algorithm 11.1 Iterative Assembly

function ASSEMBLECOMPLEXES($D, \mathfrak{I}, \mathfrak{P}, S, t$)
  ▷ Total and current complex size, candidate id and initial set of candidates
  $K \leftarrow \sum_{p \in \mathfrak{P}} S(p)$
  $k \leftarrow 1$
  $id \leftarrow 1$
  $C \leftarrow \emptyset$
  $T \leftarrow$ NIL

  ▷ Load data from restart file if present
  $F \leftarrow$ GETOPTION("RESTART_INFILE") ↪Sect B.12

  if $F \neq$ NIL **then**
    $(C, k, id, T) \leftarrow$ LOAD(F) ↪Sect. B.12
  else
    ▷ Find core protein type and initialize root monomer, candidate and tree
    $p^0 \leftarrow$ GETMAXINTERFACETYPE($\mathfrak{I}, \mathfrak{P}$)
    $m^0 \leftarrow (p^0, I)$
    $c^0 \leftarrow (0, m^0, 0, 0, 0, 0, 0, 0)$
    $C \leftarrow \{c^0\}$
    $T \leftarrow$ INITTREE(T, C, D)
  end if

  ▷ Iteratively assemble and post-process (e.g., placement interpolation and clus-
  ▷ tering) the solutions in each iteration until the full complex is constructed
  while $k < K$ **do**
    $(C', id) \leftarrow$ POPULATELEVEL($C, D, \mathfrak{P}, \mathfrak{I}, T, id, t$) ↪Alg. 11.2
    $C \leftarrow$ FINALIZELEVEL(C', T) ↪Alg. 11.5
    $k \leftarrow k + 1$
  end while

  ▷ Write and Evaluate complex candidates
  for all candidates $c \in C$ **do**
    WRITEANDEVALUATE(c) ↪Sect. B.12

    ▷ Try symmetry-optimization and re-evaluate if successful
    $c^s \leftarrow$ OPTIMIZESYMMETRY(c) ↪Sect. B.10

    if $c^s \neq$ NIL **then**
      WRITEANDEVALUATE(c^s)
    end if
  end for
end function

the placement of the most recently attached monomer of each complex candidate in C' can be optionally interpolated from all matching transformations (see Fig. 11.2 in Sect. 11.2.8), and a set C of diverse solutions which can serve as the starting point for the next iteration is obtained from C'.

After the complexes haven been fully assembled, i.e., a final set of complex candidates C has been obtained, each single complex candidate $c \in C$ is transformed into a PDB structure that is written to a file. In addition, if a reference complex is given, c can be evaluated (Sect. B.11) against that structure (this functionality is used for validating the algorithm on data sets with known complex structure). Subsequently, a symmetry optimization (Sect. B.10) of the current complex candidate is attempted, yielding c^s if successful. c^s can then be written and evaluated as well.

The algorithm then terminates after all final solutions have been processed.

11.2.5 *Level Population*

The level population, i.e., the generation of new complex candidates through attachment of new monomers to a set of parent solutions is presented in Alg. 11.2.

The algorithm requires the parent set C, the docking poses D used for attachment, as well as the protein types $\mathfrak{P}$ and interfaces $\mathfrak{I}$ present in the complex. Furthermore, the complex candidate tree T, the unique *id* of the next complex candidate to be generated and a variable t indicating whether transformation matching should be performed are given.

First, a new empty set $C' \leftarrow \emptyset$ storing the solutions generated in the current iteration is initialized. Additionally, a set D^s is required to store the matching docking poses found in the current level, along with their corresponding match score.

Then, the set of candidates C obtained in the previous iteration is sequentially processed, i.e., each complex candidate $c \in C$ is iteratively considered. First, the protein types $P(c) \subseteq \mathfrak{P}$ whose stoichiometries have not yet been exceeded in c are determined. Similarly, the unoccupied interfaces $I(m) \subseteq \mathfrak{I}$ of each monomer $m \in M(c)$ of the complex candidate are identified.

The algorithm then tries to extend each complex candidate c as follows: for each receptor monomer $r \in M(c)$ the algorithm iterates over all free interfaces $i \in I(r)$ whose ligand protein type $L(i)$ is present in the set of available protein types $P(c)$.

Each docking pose $d \in D(i)$ is then checked in turn: if d is enabled, it can be used to attempt the attachment of a new monomer l to the current complex candidate c at receptor monomer r (see Sect. 11.2.6). From the attempt, a new complex candidate c' with corresponding ligand monomer l is obtained.

If the attempt was successful, i.e., $c' \neq$ NIL, the monomer match score S^m of c', i.e., the sum of the transformation match scores (see Sect. 10.3) over all ancestor monomers $a \in M(c) \setminus \{r\}$ that provide a suitable docking pose matching l at any of their interfaces, is determined (see Sect. 11.2.6). In other words, S^m indicates how well the placement of l coincides with suitable interfaces at any previously present monomers, i.e., how well native binding modes with other monomers in the complex are resembled upon placement of l. Based on the assumptions made in Section 10.2, we can consider complexes with a large score to be energetically favorable.

The monomers and poses from which the monomer match score is obtained as well as their individual match scores are stored in D^m.

Algorithm 11.2 Level Population

```
1: function populateLevel(C, D, 𝔓, ℑ, T, id, t)
2:     u ← getOption("NUM_TMP_SOLUTIONS_TO_KEEP")
3:
4:     ▷ New complex candidate set C′ and set D^s for scored poses
5:     C′ ← ∅
6:     D^s ← ∅
7:
8:     ▷ Try to generate new solutions from each previously generated candidate
9:     for all candidates c ∈ C do
10:
11:        ▷ Determine all protein types whose stoichiometry is not exceeded in c
12:        P(c) ← availableProteinTypes(c, 𝔓)
13:
14:        ▷ Determine the unused interfaces of each monomer
15:        for all monomers m ∈ M(c) do
16:            I(m) ← availableInterfaces(m, ℑ, T)
17:        end for
18:
19:        ▷ Each free interface at each monomer of c can act as a site for attachment
20:        for all receptor monomers r ∈ M(c) do
21:            for all available interfaces i ∈ I(r) do
22:
23:                ▷ Examine only interfaces with available ligand protein type
24:                if ligand type L(i) ∈ P(c) then
25:
26:                    ▷ Consider only the enabled docking poses of interface i
27:                    for all docking poses d ∈ D(i) do
28:                        if enabled(d, T) then
29:
30:                            ▷ Try to attach a new ligand without severe clashes
31:                            (c′, l) ← attemptAttachment(d, r, c, i, id, T)   ↪Alg. 11.3
32:
33:                            if c′ ≠ NIL then
34:                                (S^m, D^m) ← monomerMatchScore(c′, r, l, i, I, D, t, T)
35:                                                                              ↪Alg. 11.4
36:                                S(c′) ← S(c) + 1 + S^m
37:                                id ← id + 1
38:                                C′ ← insert(C′, c′, u)
39:                                D^s ← updateScores(D^s, D^m)
40:                            end if
41:                        end if
42:                    end for
43:                end if
44:            end for
45:        end for
46:    end for
47:
48:    T ← updatePoseScores(D^s, T)
49:
50:    return C′, id
51: end function
```

The overall *complex match score* $S(c')$ is then calculated as follows:

$$S(c') := 1 + S(c) + S^m \tag{11.17}$$

In other words, the complex match score of c' is the sum of three terms. First, the score for the actual attachment of a new monomer l to r: obviously, w.r.t. receptor r, d is already the docking pose yielding an optimal, in fact perfect, transformation match score. Hence, the contribution of the pose used for the attachment itself to the overall score is 1. The second term corresponds to the already calculated $S(c)$ for the parent complex candidate c and the third term to the above explained monomer match score S^m.

By construction, the complex match score of the monomeric complex c^0 (cmp. Sect. 11.2.4) is $S(c^0) = 0$, because no attachment or matching can be performed with a single monomer. Consequentially, the score for any dimer $c^\diamond$ is $S(c^\diamond) = 1 + S(c^0) = 1$, because a transformation matching to other monomers can only take place if the complex candidate has at least three monomers. Hence, while in subsequent iterations ranking is performed based on the transformation matching score, in the first iteration the algorithm must rely on the ranking w.r.t. computed complex energy which is the sum of all docking scores used for assembly and transformation matching (which is updated in line 24 of Alg. 11.3 and line 43 of Alg. 11.4).

Fortunately, in the first iteration, near-native solutions are likely to be found among the first several hundreds or thousands docking poses, a still tractable number. But unless these poses are ranked very well, the combinatorial explosion will likely lead to a severe down-ranking of viable solutions when relying on docking scores in subsequent iterations. Hence, finding the near-native solutions using docking scores alone will become an infeasible task.

In contrast, the *transformation match score* which estimates how well the placement of l conforms with native interactions with other complex monomers does not rely on the docking scores but favors complex candidates where as many binding modes as possible are simultaneously satisfied; docking scores are only used as a ranking criterion for complex candidates $c_1, c_2 \in C', c_1 \neq c_2$ when $S(c_1) = S(c_2)$.

Along with the complex match score, the following updates are performed: c' is added to the set of new complex candidates C', if $|C'| < u$ (the maximum number of temporary solutions to keep) or if its complex match score is greater than the smallest one in C'. In the latter case, the smallest one is discarded, hence $|C'|$ never exceeds u.

Furthermore, the unique *id* is increased and D^s, containing the matching poses with the obtained match scores, is updated by D^m, the corresponding matching poses and scores obtained for c'.

Once all complex candidates $c \in C$ have been processed, the match scores of all matching poses in D^s are updated, i.e., if for a particular pose d from D^s the new match score is better than the one already stored in the tree T, T is updated w.r.t. the new score of d.

Finally, the new set of complex candidates C' along with the unique *id* counter is returned.

11.2.6 *Monomer Attachment*

Alg. 11.3 represents the process of monomer attachment. It requires a docking pose d as well as the parent complex candidate c, its receptor monomer r and the interface i,

Algorithm 11.3 Monomer Attachment

function ATTEMPTATTACHMENT(d, r, c, i, id, T)
  $t_c \leftarrow$ GETOPTION("MAX_CLASHES")

  ▷ Initialize transformation and type for a potential new monomer m
  ▷ to be attached, as well as overall clash count
  $\mathbf{T}^{r,d} \leftarrow T(r) \cdot T(d)$
  $l \leftarrow L(d)$
  $m \leftarrow (l, \mathbf{T}^{r,d})$
  $n_c \leftarrow 0$

  ▷ Check for clashes of m with all ancestor monomers
  for all monomers $a \in M(c) \setminus \{r\}$ **do**
    $n_a \leftarrow$ COUNTCLASHES(m, a) ↪Sect. B.4

    if $n_a > t_c$ **then**
      return (NIL,NIL)
    end if

    $n_c \leftarrow n_c + n_a$
  end for

  ▷ If no significant clashes have been found, determine total energy,
  ▷ total number of clashes, and generate a new complex candidate
  $E \leftarrow E(c) + E(d)$
  $N_c \leftarrow C(c) + n_c$

  $c' \leftarrow (id, m, c, r, 0, E, N_c, 0)$

  ▷ Mark the interfaces i of monomer r and $-i$ of m used for attachment
  ▷ in c' as locked for later levels of tree T
  $T \leftarrow$ LOCKINTERFACES($r, i, m, -i, c', T$) ↪Sect. B.1

  return (c', m)
end function

where the attachment is to be attempted. In addition, the unique id of a potential new complex candidate, as well as the underlying complex candidate tree T are needed. A threshold t_c for the maximum allowed number of steric clashes is obtained from the options.

First, the transformation $\mathbf{T}^{r,d} \leftarrow T(r) \cdot T(d)$ placing a potential new monomer is determined from the transformations $T(r)$ and $T(d)$ of receptor r and docking pose d. A new monomer m is then generated from $\mathbf{T}^{r,d}$ and ligand type $L(d)$. Likewise, a counter n_c for the overall number of clashes of m to other monomers in c is initialized.

Each of these monomers $a \in M(c) \setminus \{r\}$ is then subject to a clash check with monomer m (see Sect. B.4): if the obtained number of clashes n_a exceeds the threshold t_c, the attachment fails and the algorithm terminates by returning NIL for both the potential new complex candidate and the potential new monomer.

If t_c is not exceeded, the overall clash count n_c is increased by the number of clashes n_a between monomers a and m. If m produces no significant steric clashes with any of the other monomers, it is a valid placement.

In that case, first the overall energy $E \leftarrow E(c) + E(d)$ and the total number of clashes $N_c \leftarrow C(c) + n_c$ for the new complex candidate are determined. A new complex candidate $c' \leftarrow (id, m, c, r, 0, E, N_c, 0)$ is then initialized from the unique id, the monomers m and r, the parent complex candidate c, as well as energy E and clash count N_c.

Subsequently, the interface i of monomers r and the reverse interface $-i$ of m (Section 9.2) of c' are locked in tree T for attachment during subsequent iterations.

Finally, the new complex candidate c' and the new monomer m are returned.

11.2.7 *Monomer Match Scoring*

The monomer match scoring is a procedure that tries to determine whether an attached ligand monomer satisfies additional interfaces of other monomers than the actual receptor and is described in Alg. 11.4. The required input data are: the complex candidate c', the receptor and ligand monomers r and l, the interface i over which the attachment occurred as well as the map D of docking poses partitioned w.r.t. the corresponding interfaces, the available interfaces I, a variable t indicating whether a transformation matching is to be performed and the underlying complex candidate tree T.

First of all, the monomer match score S^m is set to zero. In addition, an empty set D^m storing the best-matching docking poses of each monomer and interface, including the corresponding match score, is initialized.

If no transformation matching is to be performed, e.g., a mono-layered ring was detected during initialization and the attachment of the most recent monomer cannot have led to a ring, S^m and D^m are returned and the algorithm terminates, otherwise the algorithm proceeds as follows.

Each ancestor monomer $a \in M(c)$, except the receptor r, is considered in turn. First, the relative transformation $\mathbf{T}^{l,a}$ of the placement $T(l)$ of ligand l w.r.t. to the transformation $T(a)$ of the ancestor is determined (cmp. Eq. 10.2):

$$\mathbf{T}^{l,a} \leftarrow T(a)^{-1} \cdot T(l) \tag{11.18}$$

Then, each interface $i_a \in I(a)$, except interface i at which the ligand is attached, is processed to determine potential matching transformations: the poses $D(i_a)$ corresponding to interface i_a are searched for a pose d that is most similar to transformation

Algorithm 11.4 Monomer Match Scoring

function MONOMERMATCHSCORE(c', r, l, i, D, I, t, T)
  ▷ Match score, set of best-matching poses with scores, and match list
  $S^m \leftarrow 0$
  $D^m \leftarrow \emptyset$
  $matches \leftarrow []$

  ▷ Do not match if we have a mono-layered ring and it is not yet closed
  if $t =$ FALSE **and not** ISLASTRINGMONOMER(c', l, G) **then** ↪Sect. B.3
    return (S^m, D^m)
  end if

  ▷ Find for each interface of each ancestor monomer the best-matching pose
  for all monomers $a \in M(c') \setminus \{r, l\}$ **do**
    ▷ Determine relative orientation of ligand monomer w.r.t. ancestor
    $\mathbf{T}^{l,a} \leftarrow T(a)^{-1} \cdot T(l)$

    for all interfaces $i_a \in I(a) \setminus \{i\}$ **do**
      $(d, s) \leftarrow$ FINDANDSCOREBESTMATCHINGPOSE($\mathbf{T}^{l,a}, D(i_a)$) ↪Sect. B.5

      if $d \neq$ NIL **then**
        $matches \leftarrow matches + [(s, d, i_a, m)]$
      end if
    end for
  end for

  $matches \leftarrow$SORTBESTTOWORST($matches$)
  $I_{match} \leftarrow \emptyset$
  $M_{match} \leftarrow \emptyset$

  ▷ Determine overall complex score w.r.t. a set of matching docking poses
  ▷ where no interface or monomer occurs twice
  for all $p \in matches$ **do**
    $(s, d, i, m) \leftarrow p$

    ▷ Only consider the current match if the respective monomer or
    ▷ interface has not already been used in a match elsewhere
    if $i \notin I_{match}$ **and** $m \notin M_{match}$ **then**
      $I_{match} \leftarrow I_{match} \cup \{i\}$
      $M_{match} \leftarrow M_{match} \cup \{m\}$

      $S^m \leftarrow S^m + s$
      $D^m \leftarrow D^m \cup \{(s, d, m)\}$
      $E(c') \leftarrow E(c') + E(d)$

      ▷ Lock interfaces i and $-i$ connecting monomers m and l for c' in T
      $T \leftarrow$ LOCKINTERFACES($r, i, l, -i, c', T$) ↪Sect. B.1
    end if
  end for

  return (S^m, D^m)
end function

$\mathbf{T}^{l,a}$. d is obtained as given in Eq. 10.3, the actual score s can then be obtained using either the heuristic displacement-/angle-based transformation match score S^{da} (Eq. 10.4) or the RMSD-based score S^{rmsd} (Eq. 10.7).

If a matching docking pose d with a score greater than zero has been found, the list of matches is extended by another tuple containing the score s obtained for the pose d at interface i_a of monomer m.

After all ancestors and corresponding interfaces have been investigated, the list of obtained potential matches is sorted, from best to worst score. In addition, two empty sets M_{match} and I_{match} are initialized, which are used to keep track of the interfaces and monomers that have already been considered, and are updated in the following procedure.

Because the ligand monomer l can only establish an interaction with one other monomer over a particular interface, and can only use one interface to interact with a particular monomer, we now must determine a set of docking poses where each interface and monomer are only considered once. Assuming that better-matching docking poses provide better support for a potential interaction between two monomers, we investigate the list of matches from the best match down to the worst as follows:

For each potential match p from the list of *matches*, first the corresponding information s, d, i, and m for the score, the used docking pose and interface as well as the matching monomer is obtained. If i and m are present in the corresponding sets I_{match} and M_{match}, they have already been used for another matching docking pose.

Otherwise, they are added to the corresponding sets, S^m is increased by s, the set of best-matching poses D^m is updated by d, meaning that d at interface i of monomer m matches the placement of ligand monomer l, and the complex energy is updated. Consequently, i at m and the corresponding reverse interface $-i$ at l is locked in c' w.r.t. to tree T.

After the full set D^m of best matching poses has been obtained, it is returned along with the monomer match score S^m, before the algorithm terminates.

11.2.8 *Level Finalization*

Once the set of potential complex candidates for the current level has been obtained, the final subset must be selected. This task is performed by the algorithm presented in Alg. 11.5. The data it requires consist of the set C' of complex candidates generated in the current iteration as well as the underlying complex candidate tree T.

First, the number k of solutions to be retained for the next iteration as well as an indicator variable *interpolate* are obtained from the options. Subsequently, an empty set C of final solutions is initialized. The determination of the final solution set C is performed as follows:

As long as the size of C is smaller than the given number of solutions k, the next-best solution $c' \in C'$ is considered. If no more solution is available, i.e., $c' = \text{NIL}$, the loop terminates.

If *interpolate* is set to TRUE, an interpolation is attempted (Sect. B.7): the final placement of the most recently attached monomer is averaged using the original placement and all transformations matching that placement (given by D^m, line 42 in Alg. 11.4); the original candidate c' is returned if the attempt failed due to steric clashes of the interpolated ligand placement. If the attempt was successful, c' is instead updated by the new ligand placement as well as the number of clashes of the new ligand place-

Algorithm 11.5 Level Finalization

```
1: function finalizeLevel(C′, T)
2:     k ← getOption("NUM_SOLUTIONS_TO_KEEP")
3:     interpolate ← getOption("INTERPOLATE_LIGAND")
4:
5:     ▷ The final, diverse set of complex candidates
6:     C ← ∅
7:
8:     ▷ Iteratively process all generated solutions, from best to worst, until
9:     ▷ the k diverse solutions to retain for the next level are found
10:    while |C| < k do
11:        c′ ← next(C′)
12:
13:        ▷ Stop if no more solutions are available
14:        if c′ = NIL then
15:            break
16:        end if
17:
18:        ▷ Try to interpolate the most recently attached monomer if requested
19:        if interpolate then
20:            c′ ← attemptInterpolation(c′)                    ↪Sect. B.7
21:        end if
22:
23:        ▷ Check whether c′ can be clustered to a representative from the
24:        ▷ diverse set C
25:        representative ← cluster(c′,C)                       ↪Sect. B.9
26:
27:        ▷ If no representative could be found, add c′ to C
28:        if representative = NIL then
29:            C ← C ∪ {c′}
30:        end if
31:    end while
32:
33:    updateTree(T, C)
34:
35:    ▷ Remove any ancestor candidate without descendant in C
36:    pruneAndDelete(T)
37:
38:    ▷ Write data to restart file if present
39:    F ← getOption("RESTART_OUTFILE")                         ↪Sect B.12
40:
41:    if F ≠ NIL then
42:        write(F, C, T)                                       ↪Sect. B.12
43:    end if
44:
45:    return C
46: end function
```

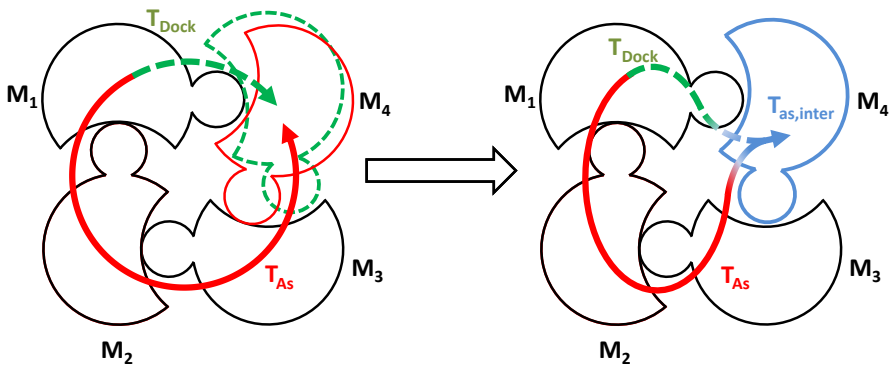


Figure 11.2: Illustration of the interpolation procedure, in analogy to the illustration scheme used in Figures 10.4 and 10.5: $\mathbf{T}_{As}$ corresponds to the transformation used for attaching a new monomer during assembly, $\mathbf{T}_{Dock}$ to a transformation of a matching docking pose found during monomer match scoring (Sect. 11.2.7) w.r.t. monomer M_1. Instead of using $\mathbf{T}_{As}$ to place the new monomer, an interpolated transformation $\mathbf{T}_{As,inter}$ that describes a mean placement of the monomer w.r.t. $\mathbf{T}_{As}$ and $\mathbf{T}_{Dock}$ can be applied.

ment w.r.t. all other monomers in c' (an exemplary illustration of the interpolation procedure is given in Fig. 11.2).

Then, a clustering procedure is applied to reduce the size of the solution space by removal of the most similar solutions w.r.t. to a given threshold (see Sect. B.9): c' is tested for similarity to all previously retained solutions in the final set C. If no representative candidate could be found in C, i.e., c' is not similar to any solution $c \in C$, c' is added to C.

After the set of diversified solutions C has been obtained, the tree T is updated accordingly, i.e., the relationships between the complex candidates from the current and previous iteration are established.

Furthermore, because only k solutions are retained for the next iteration, T is investigated for parent complex candidates from the previous iteration that have no child complex candidate in the current iteration and is pruned accordingly. The removal of a childless complex candidate may produce another childless parent farther up in the tree, hence this pruning procedure must be recursively repeated bottom-up whenever a parent complex candidate without children is encountered.

Finally, when all updates have been performed, a restart file containing all the information about tree structure, retained complex candidates and match scores for docking poses can be written upon request.

Before terminating, the algorithm returns the diverse solution set C and the updated tree T.

11.2.9 *Topology-RMSD Based Evaluation*

All solutions $c \in C$ retained in the final iteration of the algorithm can be matched and superimposed onto a reference complex R, if requested. The process of matching a particular c onto R is described in detail in Sect. B.11, together with the standard measures that have been implemented to compare the quality of model c with reference R.

However, such standard measures as for example C_α RMSD and fraction of native-contacts can underestimate the quality of the obtained structure, in particular, when

the monomers used to assemble the complex and corresponding ones present in the reference exhibit conformational differences. For example alternative loop conformations can influence the obtained RMSD and thus the estimated quality of the model. Consequently, the RMSD could indicate a wrong solution even though the overall topology, i.e., the placements of the monomers and connectivities between them is correct.

Based on the interaction RMSD (iRMSD) proposed by *Aloy et al.* [220] (described in Subsect. 3.6.2) which can be used to compare structures of protein dimers and is robust against conformational differences between the corresponding monomers in the compared dimers, we propose a novel measure called topology-RMSD (tRMSD) to compare the overall topology of the assembled complex with the reference. It is applicable to complexes with at least 3 monomers, while the original iRMSD is intended for protein dimers. The performance evaluation of 3D-MOSAIC mainly relies on this measure.

We provide two versions of tRMSDs which are applied to multimeric complexes as follows: the global gtRMSD represents the iRMSD between all reduced monomer representations (each monomer is represented by seven points as described in Subsect. 3.6.2) in c and the corresponding ones in the reference R after an optimal rigid superimposition of the whole complex. In contrast, the local tRMSD, which we mainly refer to in this thesis, represents the average over all iRMSDs of dimers interacting in the reference R after optimal rigid superimposition to the corresponding matching dimers in c.

From our experience by visual inspection, solutions with a local tRMSD ≤ 2.5Å can generally be considered near-native reconstructions with an overall correct topology and monomer orientation; smaller tRMSDs correspond to a higher structural similarity to the reference.

11.2.10 *Runtime Complexity*

To determine the runtime complexity of 3D-MOSAIC, we use the following variables: $i = |\mathfrak{I}|$ the number of overall interfaces types of a complex, $t = |\mathfrak{P}|$ the number of protein types, m the size of the fully assembled complex in terms of the number of monomers (the complexity of the iterative assembly will be dominated by the last iteration), n the total number of atoms of the fully assembled complex, $d = |\mathfrak{D}|$ the total number of docking poses, k the solutions to retain per iteration, u the number of temporary solutions to store.

The runtime complexities of the individual sub-algorithms are given in Chapter B, here, we present the overall complexity.

LEVEL POPULATION The level population (Alg. 11.2) first iterates over at most k solutions retained from the previous iteration, where the following is done for each solution: i) the available protein types and ii) interfaces are determined, the former being in $\mathcal{O}(m)$, the latter in $\mathcal{O}(mi)$, because here, the open interfaces for each monomer have to be checked. Now, for each of the monomers, the attachment via all open interfaces and docking poses is attempted, i.e., $\mathcal{O}(d)$ elements have to be considered (the d docking poses are distributed over all interfaces), yielding iii) $\mathcal{O}(md)$ in total. For each potential attachment, iv) the clashes to all other monomers in the current solution have to be checked, leading to $\mathcal{O}(mn^2)$ operations (Alg. 11.3). For allowed monomer

attachments where no significant clashes are found, we v) try to find matching transformations to all other monomers (Alg. 11.4). This step is in $\mathcal{O}(md \cdot \log(md))$ because, in the worst case, all the docking poses of all interfaces at all monomers have to be investigated and sorted w.r.t. the match score (the calculation of the match score between two transformations is constant). The insertion into the set of child solutions requires vi) $\mathcal{O}(\log(u))$, because only u potential child candidates with the best complex scores are retained for level finalization. Furthermore, the update of the match scores of each the d docking poses is in vii) $\mathcal{O}(m)$ (we can have at most m matching poses per attached monomer). In total, the generation of child candidates from one particular parent thus yields the following complexity (we can assume $\mathcal{O}(i) \in \mathcal{O}(d)$ because the d poses are distributed over i interfaces):

$$\begin{aligned}
&\underbrace{\mathcal{O}(m)}_{\text{i)}} + \underbrace{\mathcal{O}(mi)}_{\text{ii)}} + \underbrace{\mathcal{O}(md)}_{\text{iii)}} \cdot \left(\underbrace{\mathcal{O}(mn^2)}_{\text{iv)}} + \underbrace{\mathcal{O}(md \cdot \log(md))}_{\text{v)}} + \underbrace{\mathcal{O}(\log(u))}_{\text{vi)}} + \underbrace{\mathcal{O}(m)}_{\text{vii)}} \right) \\
&= \mathcal{O}(md \cdot (mn^2 + md \cdot \log(md) + \log(u) + m)) \\
&= \mathcal{O}(m^2n^2d + m^2d^2 \cdot \log(md) + md \cdot \log(u))
\end{aligned} \tag{11.19}$$

The above is done for all k solutions obtained from the previous iteration. Afterwards, the match scores of all docking poses are updated, which is in $\mathcal{O}(d \log(d))$, leading to a total complexity for one iteration of the level population of $\mathcal{O}(m^2n^2dk + m^2d^2k \cdot \log(md) + mdk \cdot \log(u))$.

LEVEL FINALIZATION From the level population, we can have obtained at most u new candidates from which now k have to be selected in the level finalization step (Alg. 11.5).

For each of the u solutions the following complexity is obtained: the ligand interpolation ($\mathcal{O}(m^2n^2)$ if enabled, see Section B.7) and the clustering have to be performed ($\mathcal{O}(m^5nku)$ over the whole set of u solutions, see Section B.9). In total, the complexity of diversifying the data set is:

$$\begin{aligned}
&\mathcal{O}(u) \cdot \mathcal{O}(m^2n^2) + \mathcal{O}(m^5nku) \\
&= \mathcal{O}(u \cdot (m^2n^2 + m^5nk)) \\
&= \mathcal{O}(m^5nku + m^2n^2u)
\end{aligned} \tag{11.20}$$

The tree update is in $\mathcal{O}(k)$, i.e., k retained solutions are inserted into the tree, the tree pruning in $O(km)$, because each previous level can only contain k solutions.

In total, the finalization of a particular level thus takes $\mathcal{O}(m^5nku + m^2n^2u + km + k)$. Because $k \in \mathcal{O}(u)$ we obtain $\mathcal{O}(m^5nku + m^2n^2u)$.

ITERATIVE ASSEMBLY The iterative assembly performs the previous two steps m times, hence we obtain:

$$\begin{aligned}
&\mathcal{O}(m) \cdot (\mathcal{O}(km^2n^2d + km^2d^2 \cdot \log(md) + mdk \cdot \log(u)) + \mathcal{O}(m^5nku + m^2n^2u)) \\
&= \mathcal{O}(km^3n^2d + km^3d^2 \cdot \log(md) + m^2dk \cdot \log(u) + m^6nku + m^3n^2u) \\
&= \mathcal{O}(m^6nku + m^3n^2dk + m^3n^2u + m^3d^2k \cdot \log(md) + m^2dk \cdot \log(u))
\end{aligned} \tag{11.21}$$

Because $k \in \mathcal{O}(u)$, we can further simplify:

$$\begin{aligned} &\mathcal{O}(m^6nku + m^3n^2dk + m^3n^2u + m^3d^2k \cdot \log(md) + m^2dk \cdot \log(u)) \\ &= \mathcal{O}(m^6nu^2 + m^3n^2du + m^3n^2u + m^3d^2k \cdot \log(md) + m^2du \cdot \log(u)) \\ &= \mathcal{O}(m^6nu^2 + m^3n^2du + m^3d^2k \cdot \log(md) + m^2du \cdot \log(u)) \end{aligned} \tag{11.22}$$

TOTAL COMPLEXITY In total, the initialization including reading and sorting of the docking poses ($\mathcal{O}(d\log(d))$), their registration in the transformation hash maps ($\mathcal{O}(d)$, the reading of the proteins ($\mathcal{O}(tn)$), the ring detection ($\mathcal{O}(i^m) + \mathcal{O}(t)$), the clash-tree construction ($\mathcal{O}(tn^4)$) and the generation of the required protein representations ($\mathcal{O}(n^3)$), as well as reading a potential restart file ($\mathcal{O}(d\log(d)) + \mathcal{O}(km)$) (see Chapter B), has the following complexity:

$$\mathcal{O}(d\log(d) + i^m + tn^4 + km) \tag{11.23}$$

The evaluation and symmetry optimization of one solution are in $\mathcal{O}(m^5n + n^2)$ and $\mathcal{O}(m^7n + m^5n^3)$, which, for k final solutions gives a complexity of:

$$\mathcal{O}(m^7nk + m^5n^3k) \tag{11.24}$$

In total, summarizing over initialization, iterative assembly and post-processing, we get the following complexity:

$$\begin{aligned} &\mathcal{O}(d\log(d) + i^m + tn^4 + km) \\ &+ \mathcal{O}(m^6nu^2 + m^3n^2du + m^3d^2k \cdot \log(md) + m^2du \cdot \log(u)) \\ &+ \mathcal{O}(m^7nk + m^5n^3k) \\ &= \mathcal{O}(m^7nk + m^6nu^2 + m^5n^3k + m^3n^2du + m^3d^2k \cdot \log(md) + m^2du \cdot \log(u) + d\log(d) \\ &+ i^m + tn^4 + km) \\ &= \mathcal{O}(m^7nk + m^6nu^2 + m^5n^3k + m^3n^2du + m^3d^2k \cdot \log(md) + m^2du \cdot \log(u) + i^m + tn^4) \end{aligned} \tag{11.25}$$

We thus see that the overall performance is strongly dominated by the number of monomers m the complex contains. The particularly complex part hereby is the matching of the monomers of two complexes, especially during symmetry optimization, where in addition to the matching, the interpolation of the transformations for the obtained symmetry mappings has to be performed.

However, this detailed analysis also shows the contribution of the different parts of the algorithm to the complexity, in particular that the population of a individual level is comparatively cheap ($\mathcal{O}(m^2n^2dk + m^2d^2k \cdot \log(md) + mdk \cdot \log(u))$ and that the complexity during assembly is dominated by the level finalization, in particular the clustering.

However, the actual performance of the algorithm strongly depends on the chosen parameters, and how the above determined runtime complexities translate to practical running times will be addressed in Subsection 13.1.10 of the results chapter.

12 BENCHMARK DATA SET AND EXPERIMENTAL DESIGN

To thoroughly assess the performance of 3D-MOSAIC, we require a diverse benchmark data set that is representative of the protein complexes present in the PDB. In addition, a careful experimental design testing the algorithm in a broad range of application scenarios is required.

In the following sections, we will present the derivation of such a benchmark data set, the preparation for the use with 3D-MOSAIC, and how the experiments have been designed.

12.1 BENCHMARK DATA SET

The benchmark set on which 3D-MOSAIC is evaluated is based on the protein complexes present in the PDB as of Oct. 23, 2012. Based on this set, we want to demonstrate the strengths and weaknesses of the algorithm in dependence of the size of the macromolecular assemblies, the number of distinct proteins involved, their stoichiometry, and their topology. Second, we want to gain insights into how the assembly performance changes in dependence of the source of the monomers used for the assembly, i.e., whether the monomers are taken from an *unbound* structure, a *dimer*, a *foreign* or the *same* complex.

We queried the Protein Data Bank for three different data sets of X-ray structures: monomeric proteins (M), dimers (D), and oligomers (O) with at least three chains and without modified polymeric residues, DNA, RNA, or antibodies (to avoid the inclusion of designed antibodies). The query result contained 4601 ambiguous PDB entries for which multiple biological assemblies with different copy numbers of the involved proteins or the complex have been observed or are believed to be likely by the authors. Additionally, hypothetical assembly variants may have been generated by the PISA [353] or PQS [354] software, sometimes confirmed by the authors, but often not. And furthermore, the biological assembly annotated by the authors in the PDB is not always the same as the one assumed to be the correct one in the corresponding publication. We automatically removed the PDB entries with multiple biological assemblies from the monomeric and dimeric sets, leaving 22906 monomers, 16602 dimers, and 12604 complexes. The remaining structures were ranked according to their structural quality as defined by the PDB.

We then assigned the SCOP (version 1.75) [355] (cmp. Sect. 3.2.1) superfamily signatures, i.e., the combination of all superfamilies of all chains in the assembly. Structures having chains without SCOP assignment were discarded. For each chain of the remaining complexes we then determined equivalent chains in the three data sets using the 100% sequence identity clusters from the PDB.

We derived four source data sets for the complexes to be assembled, called *unbound*, *dimer*, *foreign*, and *same* as follows: for each protein type of each complex we first determined all sequence-identical chains in the set of monomers (M), dimers (D), or oligomers (O) described above. Each such sequence-identical chain was checked for structural issues and discrepancies in the sequence provided by the RESSEQ entry in

the PDB file (PDB format specification [356]) and the amino-acid sequence present in the actual structure: chains with missing internal loops, mutations or non-standard residues were removed. If for a protein multiple sequence-identical chains were found, the one with the highest structural quality as defined by the PDB was retained. Subsequently, the complex categories *unbound*, *dimer*, *foreign*, and *same* were established as follows: if all proteins of a particular complex have a sequence-identical chain in M (D), this complex together with the corresponding chains from M (D) is added to *unbound* (*dimer*). In the case of *foreign*, and *same*, all proteins of the complex must have sequence-identical chains in O, in *foreign* these chains originate from a different complex than the one under consideration, in *same* they are taken from the complex itself. The four initial source data sets comprise 480, 842, 6003, and 9882 assemblies and their corresponding sequence-identical chains, respectively.

All complexes were checked for structural problems. Some assemblies were found to be split into several MODEL entries (PDB format specification [356]); here a unification into one single structure, chain renaming and clash checking were performed. Structures with more than 10 steric clashes per pair of monomers (heavy-atom distance 1.5Å), multiple connected components (heavy-atom contact range 6.5Å), or those exceeding the limitations of the PDB format (more than 100,000 atoms or 62 chains) were discarded. When aiming for assembling even larger complexes, other file formats must be used, however, the above restrictions apply to only 83 of the 12,604 complexes obtained from the PDB. Furthermore, by allowing for complexes with up to 62 chains, our benchmark set already significantly extends the limits and diversity of benchmark sets used in other studies [184, 185, 193] and is thus well-suited to demonstrate the capabilities of 3D-MOSAIC.

Many of the retained structures contain hetero groups such as ligands or solvent molecules. In such cases, the question often arises whether such compounds are required in docking applications. Cofactors for example can be considered an essential part of the protein and should thus be explicitly dealt with in docking experiments. In contrast, solvent molecules can often be ignored because they are often expected to have no significant influence on the protein conformation or the interactions at protein-protein interfaces and are mainly used to buffer the solution during crystallization. Despite the sheer amount of hetero groups present in the PDB, to the best of our knowledge, an extensive database with annotated functions of the hetero groups present in the PDB is not available and can only be found for special subsets of compounds of great interest, e.g., the EBI cofactor database [357].

We thus manually inspected all hetero groups present in at least 50 of the structures of our data sets (186 compounds) and divided them into two sets, one containing cofactors, ligands, and ions considered to be essential, the other one molecules which can be ignored (mostly solvent). We also added the cofactors reported in the EBI cofactor database to the former set. Ignored hetero groups were removed from the structures, entries with compounds not present in one of the two lists were discarded. We further discarded all structures with hetero groups not accountable for by the atom parameter sets provided by the employed docking algorithm (RosettaDock, see Sect. 12.6), leaving 285, 491, 3229, and 5043 entries for the *unbound*, *dimer*, *foreign*, and *same* data sets, respectively.

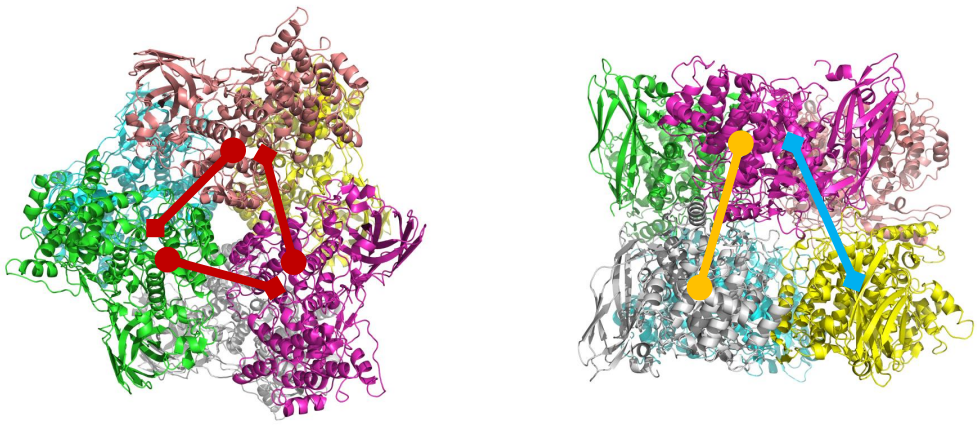
Yielding a data set with a sufficient number of protein complexes of six or more monomers, we decided to remove all complexes with fewer components. For each remaining signature in the four data sets, we selected the assembly with the highest

structural quality [358] as the representative of that signature. These 576 structures were then manually verified against the corresponding literature and related PDB entries. When no evidence could be found that the assembly under consideration is correctly annotated, it was discarded from the data set and replaced (if available) by the next-ranked valid structure of the corresponding signature. Complexes automatically resolved by the Protein Structure Initiative (PSI) [359] were kept in the data set.

The final source data sets sum up to 350 complexes in total, with *unbound*, *dimer*, *foreign*, and *same* being comprised of 9, 10, 122, and 209 complexes, respectively.

12.2 BINDING MODE DETECTION

A general property of most complexes is that of recurring binding modes (modes of protein-protein interactions). For example, the hexameric haemocyanin from panulirus interruptus - our initial test example during the development of 3D-MOSAIC - consists of two trimers stacked on top of each other. Each monomer establishes interactions to both other monomers in the same trimer (horizontal interactions, Fig. 12.1a), and, due to a slight rotation of the trimers against each other, also two interfaces to two monomers of the other trimer (vertical interactions, Fig. 12.1b). All horizontal binding modes are identical, with our monomer being once on each side of the protein-protein interface. The vertical binding modes of a particular monomer are distinct, however, each of the monomers again establishes both of the interfaces. Furthermore, the vertical binding modes are symmetric w.r.t. to the participating monomers. In total, we thus have three distinct binding modes: the horizontal one with a redundancy of six, the other two with a redundancy of three each.



(a) Horizontal, asymmetric binding modes of 1HCY

(b) Vertical, symmetric binding modes of 1HCY

Figure 12.1: The unique binding modes of 1HCY, the hexameric haemocyanin from panulirus interruptus

Because it is sufficient to perform one binary docking run per unique binding mode, we first determined the respective unique binding modes for each of the 350 complexes as follows: for each monomer in each of the complexes, we aligned the cor-

responding highest-quality sequence-identical chain (reference monomer) from the corresponding source data set using the *align* command in PyMol [360]. The contacts between pairs of aligned monomers were determined as those residues in both chains whose C_α atoms are closer than 10.0Å from any residue in the other chain. The resulting dimers, i.e., pairs of contacting chains, were then clustered with a C_α cluster RMSD of 5.0Å. From each cluster, the one with the least severe clashes (heavy-atom distance < 1.5Å) was kept as the representative of a unique binding mode.

12.3 SINGLE RESIDUE-PAIR INTERACTION CONSTRAINTS (SRPIC)

In the previous section, we have determined the approximate binding modes of all proteins in a complex. While such information is often available, for example from low-resolution electron density maps [361] or protein-protein interactions in homologous proteins [362], this is not always the case.

But even in data-scarce situations, assumptions about potential binding modes between pairs of proteins can be made: for example studies on correlated mutations [322, 323, 197, 199] of individual residues or cross-linking studies [326] can provide information on whether a specific pair of residues from two proteins might interact. We thus want to investigate if knowing one single interacting residue pair per native binding mode is sufficient to successfully reconstruct the full complex.

To simulate such experiments, we determined for each unique binding mode in a subset of complexes those pairs of surface residues (one from each monomer participating in the binding mode) whose C_α atom distance is at most 10.0Å. For each unique binding mode, one such residue pair was randomly selected and a docking start dimer (required by RosettaDock) was generated as follows: The protein centroids were determined and the monomers were rotated so that the C_α atoms and the centroids were placed on a straight line such that the C_α atoms are placed between the centroids at a C_α-C_α distance of 10.0ÅİIn addition, we derived distance constraints for the C_α atoms of the interacting residues, so-called single residue-pair interaction constraints (SRPIC), which were applied during docking as explained in the following section.

Analogously, we generated 10 false-positive dimers obtained from residue-pairs where at least one of the residues was not involved in any binding mode, mimicking a situation where noisy or wrong experimental data is available or additional dimer binding modes are known or plausible. A manual optimization of the starting dimer was intentionally not carried out to simulate the case where only very little knowledge is available.

12.4 DIMER PREPARATION AND DOCKING EXPERIMENTS

All starting dimers were prepared for docking with RosettaDock's prepack protocol to pack side chains into low-energy conformations. For the tests in this work, we employed two different docking scenarios. The docking poses generated for the validation on the benchmark set and the evaluation on Comeau's data set [193] were generated using RosettaDock's standard parameters (*-dock_pert 3 8*, *-spin*, *-ex1* and *-ex2aro*) for a local docking protocol in low-resolution mode (side chains are represented by centroid atoms). The local refinement in the high-resolution stage was skipped as it

can be expected that optimal side-chain and rigid-body orientations of the dimers will differ from those in the complex because of additional energy contributions of the other surrounding components. For each interface 10,000 decoys were generated.

For the SRPIC experiments (Section 12.3), we performed constraint dockings penalizing any pose whose constrained residues' C_α atom distance exceeds 10.0Å by applying a bounded harmonic penalty score (slope 1) in dependence on the deviation from that threshold. Here, because the side-chain and rigid-body orientation optimization can have critical influence on that penalty, we applied the full docking protocol including local refinement. In addition, due to the straightforward generation of starting dimers, we increased the perturbation parameters significantly to *-dock_pert 20 30* to achieve an extensive, wide-range sampling of the whole orientational space satisfying the interaction constraints. Again, we generated 10,000 docking poses. Due to the wide-range sampling, poses obtained from the dockings w.r.t. one start dimer might better fulfill a constraint that is different from the one the original start dimer was created from. In such a case, the pose was reassigned to the binding mode for which the lowest constraint penalty was achieved. We then repeated the following clustering procedure until a stable clustering was achieved: Using the constant-time RMSD-based clustering approach we proposed in [351] and a Ward-distance of 5.0, all poses assigned to one interface were clustered and singleton clusters were removed. Once no more singleton clusters were found, the clustering was considered stable and the iteration stopped.

For each reference monomer used during binding mode detection, a randomly rotated copy centered at the origin was created and used for assembly. The docking transformations were determined w.r.t. to these centered monomers and grouped into binding modes and interfaces as described in Sect. 9.2.

12.5 RESCORING DOCKING POSES

To provide an estimate of the overall interaction energy of the assembled complexes, redundant score contributions of the same monomer to the total complex energy must be avoided (compare Sect. 9.2). To this end, we rescored all poses using RosettaDock's *cen_std* weights and determined the interaction energy (Eq. 9.3) of each pose by subtracting the scores of the individual isolated monomers from the total score of the docking pose.

12.6 ASSEMBLY EXPERIMENTS

To evaluate our algorithm, we use three different scenarios of increasing difficulty. The first scenario corresponds to cases where the information about the approximate location of the binding modes is available. The second scenario simulates a situation where less detailed information is available: only one pair of interacting residues per binding mode is known, furthermore additional false positive interactions corresponding to binding modes that are plausible but not natively present in the complex are introduced. In the third scenario, we use global dockings, without any information on protein-protein interactions.

12.6.1 *Benchmark Experiments*

Due to the high diversity in our benchmark set, we ran our algorithm with different parameter configurations. In the benchmark validation, we used different parameters for pre-/intra-/post-clustering C_α RMSD (1.0Å/2.0Å/3.0Å and 1.0Å/3.0Å/5.0Å), clash checking (10, 25, 50, 150 allowed clashes [363] between any two monomers) and dis-/enabled interpolation between matching docking transformations. We also investigated the effect of the selected transformation similarity measure. For S^{da} (Eq. 10.4) the following combinations for l_{max}/a_{max} were used: 1.0Å/5.0°, 1.5Å/10.0°, and 2.5Å/15.0°. Analogously, runs for S^{rmsd} (Eq. 10.6) used configurations for displacement-based prefiltering (l_{max}) and $rmsd_{max}$ of 1.0Å/3.0Å, 1.5Å/4.5Å, and 2.5Å/7.5Å. In total, this yielded 96 parameter sets. The RMSD threshold for detection of symmetric binding modes was set to 0.5Å, the fraction of poses with an RMSD below that threshold to 1%. All runs were performed using a solution reduction scheme, considering 2000 solutions in the first iteration, reduced by 50% in each subsequent iteration until a threshold of 100 solutions is reached. For performance comparison, we also performed runs with smaller clustering values, i.e., 1.0Å/0.0Å/1.0Å and 1.0Å/1.0Å/2.0Å for pre-/intra-/post-clustering (96 parameter sets).

Analogously, we performed a baseline study demonstrating the effectiveness of the transformation match score. Similar configurations were used, but completely disabling the influence of other factors on the assembly process. Symmetric binding mode determination was turned off and pre-/intra-/post-clustering was set to 0.0Å/0.0Å/0.0Å and 1.0Å/0.0Å/0.0Å respectively (96 parameter sets in total). In doing so, we can investigate how well the transformation match score is able to rank near-native (sub-)complexes such that they survive subsequent iterations when no clustering (or only pre-clustering to diversify the docking poses) is applied. For comparison, we performed runs completely disabling the transformation match score. In this case, the parameters configuring S^{da} and S^{rmsd} have no effect, hence only eight configurations combining the cluster and clash parameters are generated.

In all of the above experiments, we changed the parameters for large complexes with more than 20 or 40 monomers as follows to reduce the required computational time: after adding the 20th monomer, the cluster parameters were reduced by a factor of 5. In addition, the number of solutions per level was reduced to 50. After attachment of the 40th monomer, cluster parameters were again reduced by a factor of 2 and the number of solutions per level was set to 25. In addition, in the first 20 levels, all docking poses were considered; after 20 (40) levels, only the 500 (250) poses of each interface yielding the highest transformation matching score were used.

12.6.2 *SRPIC Experiments*

In the SRPIC experiments (single residue-pair interaction constraint experiments, Section 12.3), the complexes can be expected to be more diverse due to the wide-range sampling, hence we only used a C_α RMSD threshold of 1.0Å/3.0Å/5.0Å for pre-/intra-/post-clustering. Clash checking values of 50 and 150 were used. For runs involving S^{da} (Eq. 10.4) the following combinations for l_{max}/a_{max} were used: 2.5Å/15.0° and 3.0Å/20.0°. For S^{rmsd} (Eq. 10.6) we used the following parameters for (l_{max}) and d_{max} 2.5Å/7.5Å and 3.0Å/9.0Å. Symmetric binding mode detection was turned off, because the very diverse set of obtained poses can not be expected to produce any reli-

able information on the symmetry of protein-protein interfaces. However, we tested an additional feature in half of the runs: the pre-ranking of the poses obtained in a level by the number of detected symmetry mappings (Def. 9.5) with a subsequent ranking of transformation match score for (sub-)complexes with the same number of mappings. The number of solutions again followed a solution reduction scheme with 2000 solutions in the first level, 50% reduction rate in subsequent levels, towards a threshold of 250 solutions. 32 parameter sets were obtained in total.

Again some changes in parameter configuration were introduced to reduce the computational cost for larger complexes: after 6 (16) levels, only the 500 (250) poses yielding the highest transformation matching score were considered. In addition, the number of solutions per level was reduced to 100 solutions after the attachment of the 16th monomer. Due to the diversity of the docking poses, the clustering parameters were kept unchanged.

Assembly experiments were performed with dockings corresponding to the following binding modes: i) native ones only, ii) native ones + three false positives, iii) native ones + six false positives, and iv) native ones + all (ten) false positives.

12.6.3 *Comparison with CombDock*

To compare our algorithm to CombDock, we ran CombDock on all benchmark complexes with the standard parameter configuration provided by CombDock. From the docking poses, we obtained the corresponding transformations as described in Sect. 12.5 and Fig. 9.1. Because CombDock performs an all-vs-all pairwise docking and only considers the best 100 solutions of each docking, we generated the following four assembly data sets: i) all pairs of dockings, with 100 solutions per pair, ii) all pairs of dockings, with all available solutions per pair, iii) dockings corresponding to natively interacting protein types, with 100 solutions per pair, ii) dockings corresponding to natively interacting protein types, with all available solutions per pair. The latter two cases only apply to hetero-oligomers, because in the homo-oligomeric case, only one protein type is present.

The comparison with CombDock employs parameters that are a hybrid of the benchmark and SRPIC runs: because CombDock employs global dockings, the solutions can be expected to be even more diverse than in the SRPIC experiments, however, we refrained from increasing the transformation match score parameters even more, because we do not want to risk assigning transformation match scores to false-positive matchings. Solution and clustering reduction scheme were applied as in the benchmark data set because CombDock typically produces less solutions per protein-protein-docking than RosettaDock. All other parameters were the same as in the SRIPC experiments, except the following: because of the global docking, we do not have distinct interfaces, and hence, the flag prohibiting attachment of multiple proteins to the same interface was turned off. In addition, the rejection of all non-matching solutions once at least one complex candidate with matching docking poses was found in a particular level was turned off to make the algorithm robust to chance matchings. 32 parameter sets were generated in total.

12.6.4 *Computational Resources*

All docking and assembly experiments were performed on the high-performance cluster MOGON at the Johannes Gutenberg University in Mainz, Germany. MOGON consists of 535 nodes, each with 4 CPUs and 16 cores per CPU, each clocked with 2.1 Ghz. The maximum allowed running time of a job was limited to 5 days by the queuing system.

13

EVALUATION OF 3D-MOSAIC IN DIFFERENT APPLICATION SCENARIOS

In this chapter, we present the results obtained with 3D-MOSAIC in three different application scenarios of increasing difficulty. In the first section, we focus on the performance of 3D-MOSAIC on the benchmark data set that we established.

The second section comprises the results for the assembly of complexes when only a single residue-residue interaction per binding mode is known.

The concluding section then summarizes how well 3D-MOSAIC can assemble macromolecular complexes when no interaction information is given and global dockings have to be applied.

13.1 BENCHMARK DATA SET

In this section, we present the results of 3D-MOSAIC for the assembly of complexes from our benchmark data set. The first section, however, deals with the results obtained from the RosettaDock docking runs and the selection of proper binding modes. We present the docking results obtained for all determined binding modes and address the selection of those binding modes that are considered irrelevant for the assembly process and are thus not considered during assembly.

We then present the overall performance of 3D-MOSAIC on our benchmark data set using 96 different parameter combinations for clustering, clash-checking, interpolation and transformation matching. From the 96 parameter sets used for the validation of 3D-MOSAIC on our benchmark data set we then determine well-performing parameter sets that are assessed using 1000 times ten-fold cross-validation and an evaluation on a previously unseen set of complexes.

Subsequently, we address the individual effects of some of 3D-MOSAIC's modules: we demonstrate the importance of the transformation match score by dissecting the contributions of the most relevant components of 3D-MOSAIC (clustering and transformation matching) to the overall assembly performance in several baseline runs. Thereafter, we present the symmetry-based post-ranking scheme that we use to assess the performance in all assembly experiments. Throughout this chapter, we will always refer to the symmetry-based ranks.

We conclude this chapter by addressing the limitations of our algorithm, present a selection of examples which could be assembled with 3D-MOSAIC but are beyond the capabilities of other algorithms and demonstrate the running time behavior of 3D-MOSAIC in dependence of different parameter combinations.

13.1.1 *Docking Results and Native Binding Mode Determination*

By design, 3D-MOSAIC can be expected to favor complex topologies with a high interconnectivity of the individual components, because here, a docking pose might match interfaces from many other surrounding monomers. To avoid an overestimation of the

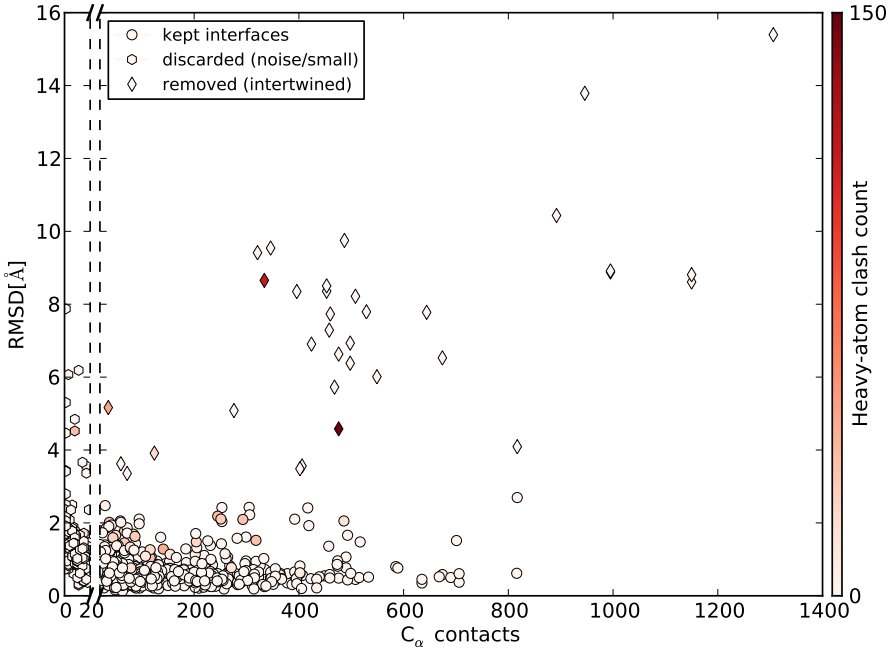


Figure 13.1: Best docking dimer C_α-RMSDs per binding mode vs. C_α contact count (radius 10Å). The results for the range from 0-20 C_α contacts are scaled by a factor of 2.5 compared to the range from 20-1,400. The heat map indicates the number of heavy-atom clashes (atom distance $<$ 1.5Å) occurring in the docking start dimer which is generated using monomers from the respective source data set (*unbound*, *dimer*, *foreign*, *same*).

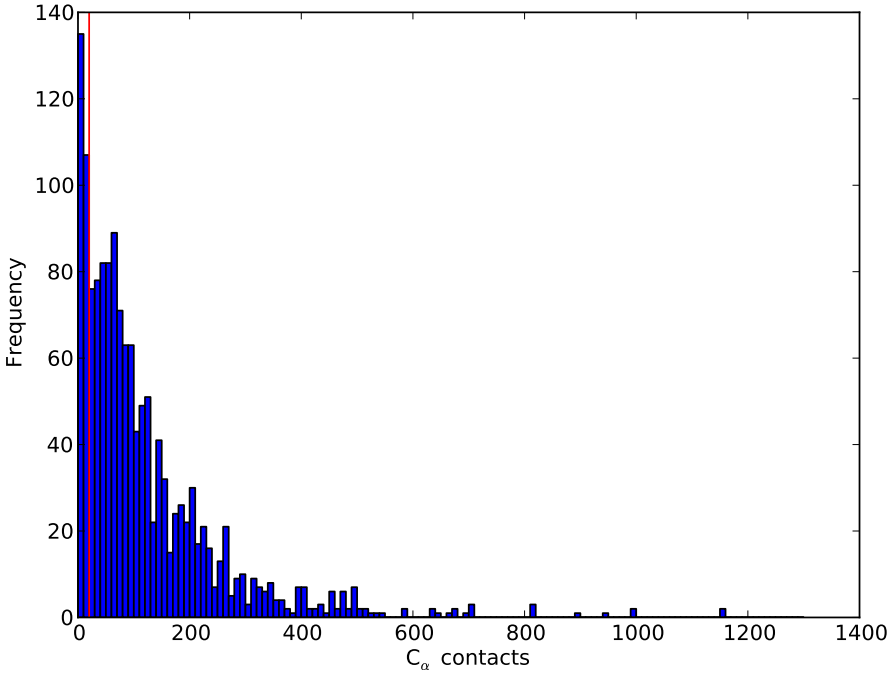


Figure 13.2: C_α contact distribution. The red line indicates the chosen cutoff of 20 residues.

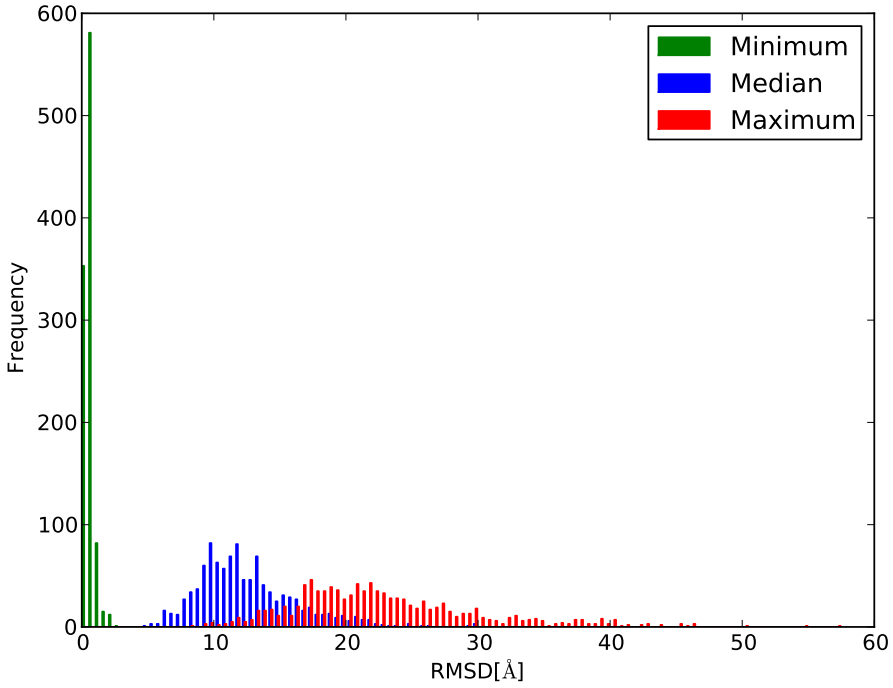


Figure 13.3: Distribution of minimum, median and maximum dimer C_α RMSD over all 10,000 dockings per binding mode.

performance of our algorithm, we first removed all binding modes from the initial set of 1,461 modes that are unlikely to contribute significantly to the complex stability. To this end, we determined the number of contacts (i.e., the number of residues in a protein-protein interface with a C_α atom distance of at most 10.0Å to a residue in the respective binding partner) as well as the minimum RMSD of all poses obtained from the dockings w.r.t. a particular binding mode.

An investigation of the minimum RMSD over all docking poses per binding mode w.r.t. the number of contacts between the two monomers in the respective binding mode (Fig. 13.1) reveals that RosettaDock is often unable to find a near-native binding conformation when the number of residues in the interface of the binding mode is less than 20. This indicates that these binding modes are likely to be incidental contacts that do not contribute significantly to the association of the complex. These observations are in good concordance with the C_α distribution of encountered contacts between residues of both dimer monomers (Fig. 13.2): there is a distinct overrepresentation of binding modes with less than 20 contacts (first two bars, bin size 10).

Thus, we discarded all 230 binding modes that form less than 20 C_α contacts. 10 complexes (and additional 14 binding modes) had to be excluded because they decomposed into several connected components when all binding modes with less than 20 residues in the protein-protein interface were removed. In such cases, smaller oligomers often assemble independently before being able to provide sufficiently large interfaces for the final complex formation [364], as for example in the case of 1IXR (RuvA-RuvB complex), where two tetrameric sub-complexes of RuvA assemble into an octameric shell to which satellite RuvB units become attached. Even more remarkably, in the case of 2A3X (serum amyloid P component), the dimerization of

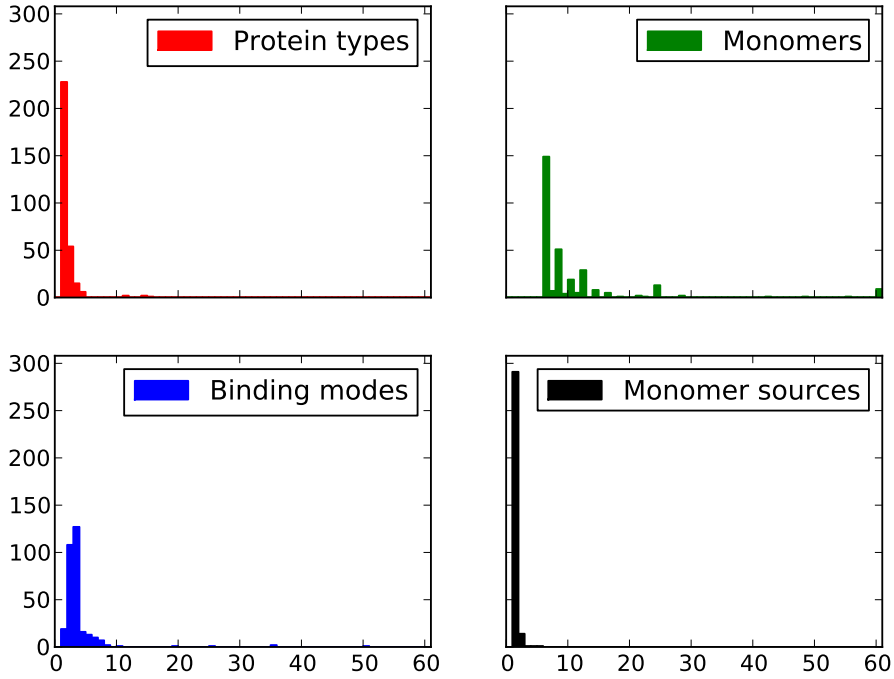


Figure 13.4: Distributions of the number of distinct protein types, the number of monomers, the number of distinct binding modes and the number of monomer sources of all complexes.

two pentameric rings into a decamer is the result of a ligand-mediated non-covalent linkage of both pentamers [365].

In the 44 cases where the lowest C_α RMSD between the generated pairwise docking poses and the native dimer was greater than 3.0Å (diamond-shaped data points, see Fig. 13.1), the docking algorithm was not able to reproduce the near-native binding mode despite the interfaces being sufficiently large. In all these cases, the two binding partners significantly protrude into one another, some of them forming inter-chain β-sheets that leave only a very small docking funnel. Since this can be considered an artifact of the docking procedure, we discarded the corresponding 32 complexes with 173 binding modes from the data set, leaving 308 complexes (1,044 binding modes) in total, with 9, 8, 108, and 183 complexes in the *unbound*, *dimer*, *foreign*, *same* sets, respectively.

An assessment of the minimum, median and maximum dimer C_α RMSD over all 10,000 poses per binding mode (Fig. 13.3) reveals that RosettaDock was able to find a near-native dimer for 1,031 of the remaining binding modes, with a mean (standard deviation) RMSD for the minimum-RMSD distribution of 0.654Å (0.326Å). However, the distributions of median and maximum RMSDs per binding mode with mean values (standard deviations) of 12.457Å (3.718Å) and 22.756Å (7.046Å), respectively, show that our docking protocol yielded an interface sampling with a sufficient amount of decoys to provide a reasonable test scenario for our algorithm.

The 308 complexes of the benchmark set are diverse in nature of the number of monomers (6-60), different protein types (1-15), binding modes (1-50), and pdb source structures used for the assembly of the respective complex (1-5). Distributions of the respective properties are given in Fig. 13.4. A complete overview of all benchmark complexes including the above properties can be found in Tab. C.1.

13.1.2 *Benchmark Performance*

3D-MOSAIC combines various different components, most notably the transformation match score, the clustering procedure, and the symmetry optimization, to allow for a successful reconstruction of a diverse range of protein complexes. In later sections, we will discuss the individual effects of these components; in particular, we will show in section 13.1.5 that the transformation match score we proposed in this work is the main driving force for a successful assembly, even when other modules such as clustering are disabled.

In this section, we will first present the overall performance of the 3D-MOSAIC as a whole, i.e., when all of these components are enabled. To this end, we established 48 parameter sets for each of the two transformation match scores, S^{da} and S^{rmsd}, with different thresholds for clustering, clash checking, and transformation matching. Apart from the *tms*-specific parameters, the parameter sets are identical for both transformation match scores. A list of the 48 parameter sets employed for each of the transformation match scores can be found in Tabs. C.7 and C.9.

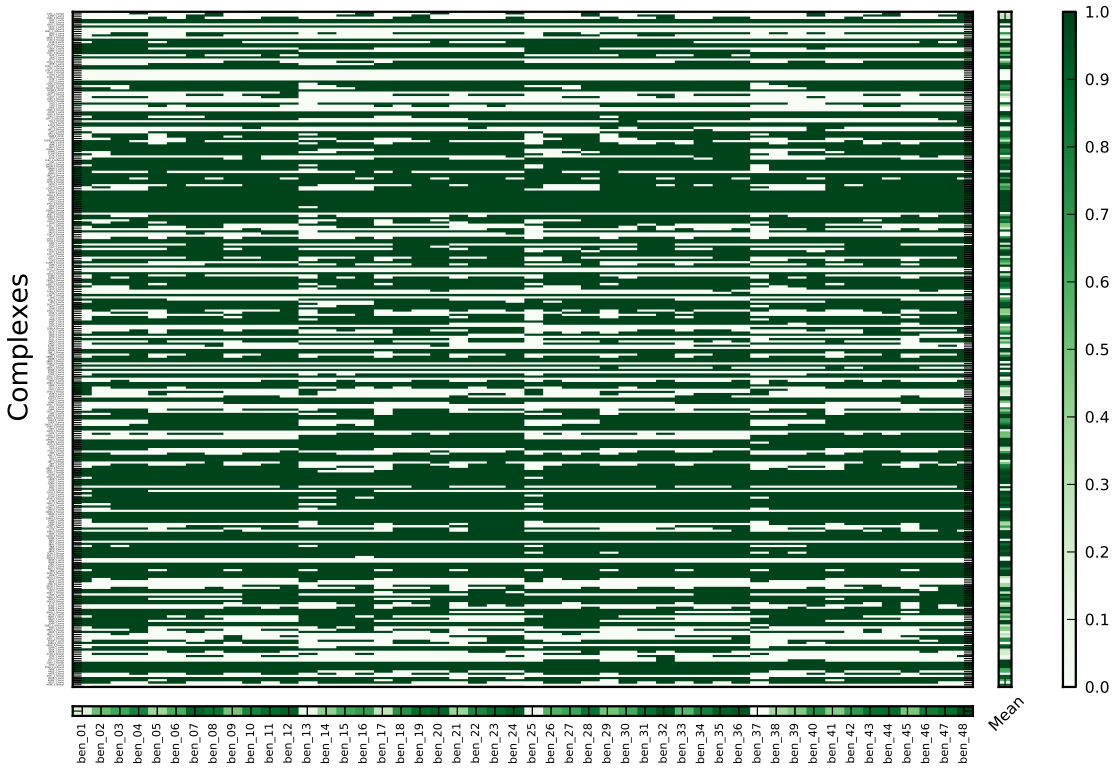
An overview of the successful benchmark runs for each benchmark complex, differentiated w.r.t. S^{da} and S^{rmsd} can be found in Figs. 13.5a and 13.5b, respectively (a detailed view on the performances of each individual parameter set is presented in Tabs. C.8 and C.10, respectively). It can be clearly seen that, regardless of which transformation match score is used, the overall number of successful assemblies per parameter set is good: the average number of complexes per parameter set for which a near-native solution could be obtained (tRMSD $\leq$ 2.5Å), is 198.96 for S^{da} and 202.50 for S^{rmsd}. The corresponding distributions are presented in Fig. 13.6, their main characteristics are given in Tab. 13.1

A Wilcoxon signed-rank test comparing the two distributions of the number of near-native complexes per parameter set w.r.t. S^{da} and S^{rmsd} yielded a p-value of 0.0052, showing that the distributions are significantly different w.r.t to a significance level $\alpha = 0.05$. Hence, we observe that S^{rmsd} on average ($\Delta_{S^{rmsd},S^{da}}$ mean 3.54) yields a near-native solution for more complexes per parameter set.

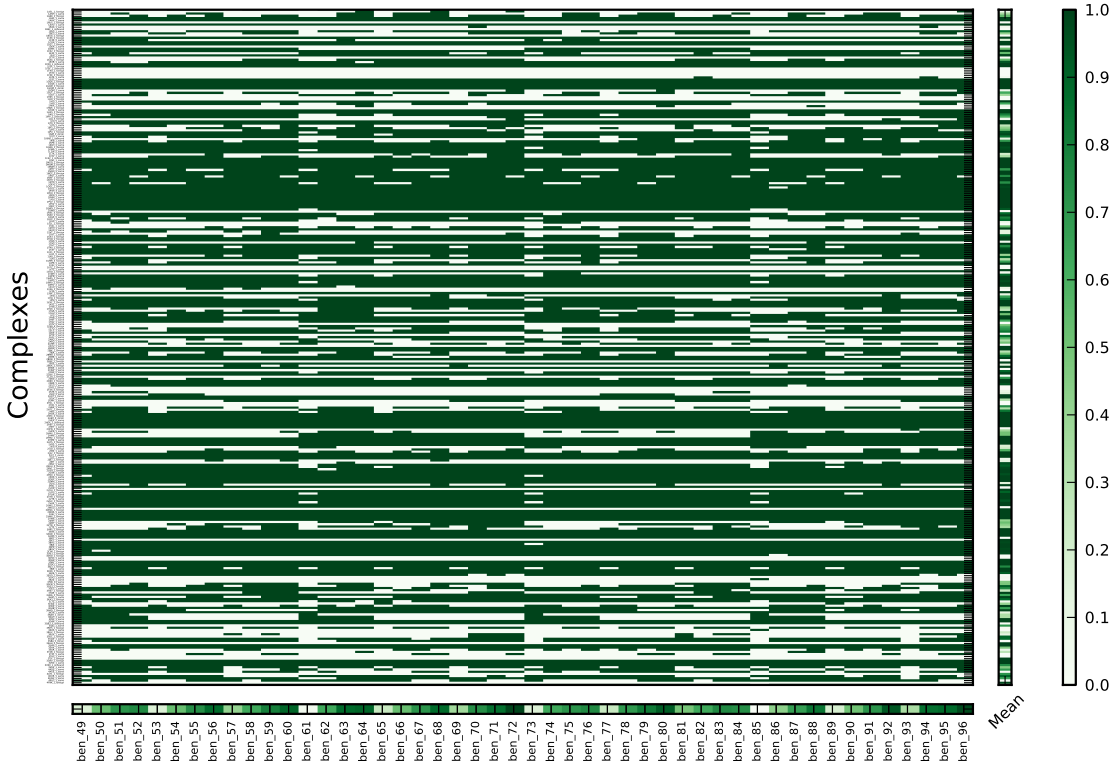
On the other hand, while using S^{rmsd} yields more complexes with near-native solutions on average, S^{da} yields a greater total number of such complexes over all parameter sets. In total, with S^{rmsd}, 267 (86.7%) complexes could be reconstructed, whereas S^{da} yields 272 (88.3%) complexes. In total, over all benchmark runs, 278 (90.3%) of the complexes could be reconstructed.

We did not observe any significant difference w.r.t. the underlying data set, i.e., whether the monomers used for the assembly originate from an *unbound* structure, a *dimer*, a *foreign* or the *same* complex. Unfortunately, the number of complexes to be assembled from *unbound* or *dimer* sources is too small (9 respectively 8 complexes) to yield any reliable statement on performance differences to corresponding complexes from the other data sets, yet we observe that all but one of the 17 structures (3SBA, *dimer* set) could be successfully reconstructed. In addition, we performed a Wilcoxon signed-rank test over the success rates (fraction of parameter sets with correct reconstruction) of the 64 corresponding structures from the *foreign* and *same* data sets, which yielded a p-value of 0.29, showing that there is no significant difference in performance.

We also observe that some parameter sets are more promising than others: in particular, for example, we notice the following in Fig. 13.5: parameter sets with a



(a) Parameter sets using S^{da}



(b) Parameter sets using S^{rmsd}

Figure 13.5: Matrix of successes of 3D-MOSAIC w.r.t. the benchmark runs over all complexes for the two transformation match scores: each column corresponds to one parameter set, each line to one complex; green matrix entries denote a successful reconstruction of the corresponding complex using the respective parameter set, white ones a failure. The small bars below and to the right of the main plot represent the corresponding average success per parameter set and per complex, respectively.

low number of allowed clashes per pair of monomers (every fourth parameter set $ben\_i, i = 4j + 1, j \in \{0, ..., 23\}$) perform worse on average than the subsequent three sets $ben\_i + 2, ..., ben\_i + 4$. This is especially important, because typically, applying as many as 96 different parameter sets is not doable in a real application scenario. In this case, such a parameter set can possibly considered to be a bad choice.

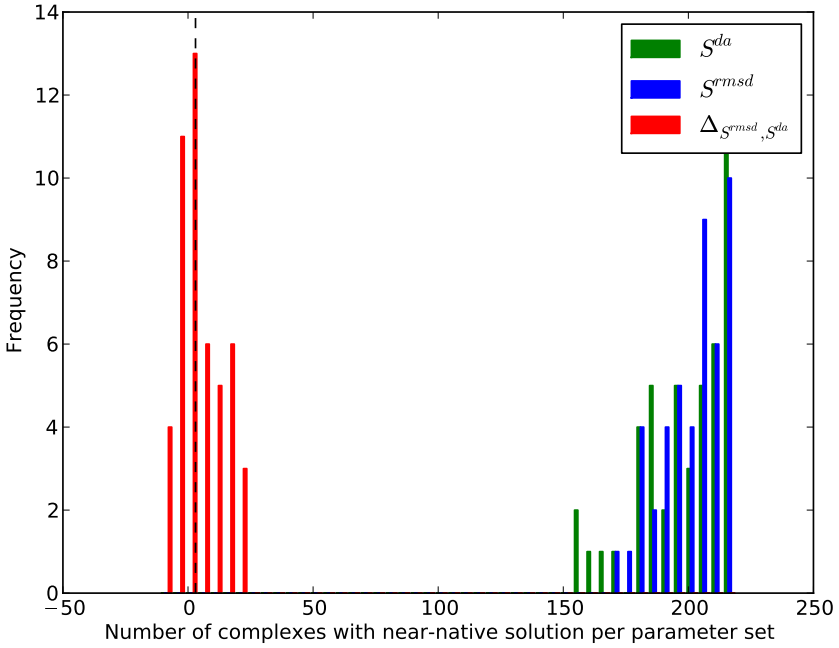


Figure 13.6: Distributions for the number of complexes with near-native solutions over all parameter sets for the two transformation match scores S^{da} and S^{rmsd} and their difference $\Delta_{S^{rmsd},S^{da}}$.

	Number of complexes with near-native solution	
	Mean	σ
S^{da}	198.96	8.43
S^{rmsd}	202.50	12.93
$\Delta_{S^{rmsd},S^{da}}$	3.54	8.43

Table 13.1: Main characteristics of the distributions for the number of complexes with near-native solutions over all parameter sets for the two transformation match scores S^{da} and S^{rmsd} and their difference $\Delta_{S^{rmsd},S^{da}}$.

When only a small number of parameter sets can be used, it is thus important to determine that parameter set (or a combination thereof) for which the expected rate of successful reconstructions of previously unknown complexes is maximal. The process of selecting and assessing promising parameter sets and their evaluation on previously unseen complexes is described in the the following two sections.

In addition, Fig. 13.5 indicates that there are some complexes for which none of the benchmark parameter leads to a correct assembly. Likewise, some complexes can be assembled with a large variety of different parameter sets. The determination of the common properties of such complexes is presented in Subsect. 13.1.8.

	S^{da}		S^{rmsd}	
Parameter set	Coverage	cov_{cv}	Coverage	cov_{cv}
Best single	223 (72.4%)	70.8%	221 (71.8%)	69.1%
Best pair	250 (81.2%)	78.7%	245 (79.5%)	76.7%
Best triple	258 (83.8%)	80.4%	252 (81.8%)	78.1%

Table 13.2: Best-covering single, pair and triple parameter sets for both transformation match scores S^{da} and S^{rmsd} and corresponding cross-validation coverage rates. A benchmark complex is considered correctly reconstructed if 3D-MOSAIC could generate a solution with tRMSD leq2.5Åto the reference.

13.1.3 *Selection of Parameter Sets and Cross-Validation Coverage*

In Subsection 13.1.2, we assessed the general capability of 3D-MOSAIC to assemble oligomeric complexes under a wide variety of different parameter sets, including different values for clash checking, clustering, as well as transformation matching. However, in a real application scenario, the number of parameter sets that can be tried is typically small, for example because the computational resources are limited or because the total solution space to be visually inspected becomes too large when using too many different parameter sets.

Hence, we are now interested in selecting a small number of parameter sets that are able to cover a wide range of different complexes, i.e., the number of benchmark complexes that can be reconstructed with the selected parameter sets is maximal. In Subsect. 13.1.1, we have already seen that the benchmark set that we determined is diverse w.r.t. the number of monomers, the different protein types, binding modes, and pdb source structures used for the assembly of the respective complex. Thus, we can not expect to reconstruct all complexes using a single parameter set; an assumption that is confirmed by the findings presented in Subsect. 13.1.2.

Consequently, we consider three different cases for the parameter set selection: i) the single parameter set, ii) the pair of parameter sets, and iii) the triple of parameter sets which yields the highest number (coverage) of reconstructed complexes, respectively. Due to the diversity of the benchmark data set and because we aim at providing parameter sets that are generally applicable, we did not investigate parameter sets that are tuned to complexes with specific properties. Like in Subsect. 13.1.2, a reconstruction is considered successful if $tRMSD \leq 2.5$Å.

In addition to the actual selection of the best-covering single (and pair and triple) parameter set we also perform a cross-validation (see Subsect. 3.7.1) to estimate how well such a selection generalizes to previously unseen complexes. To this end, we employ a 1000 times 10-fold cross-validation, i.e., the following is repeated for 1000 iterations: the set of 308 benchmark complexes is randomly divided into 10 folds, where each of the ten folds in turn is once used for reconstruction (validation) while the best-covering parameter set (combination) is selected from the other 9 training folds. We use a binary loss function L which returns 1 if the complex is correctly reconstructed and 0 otherwise. The cross-validation coverage cov_{cv} is then the mean percentage of successfully reconstructed complexes over all 10 validation folds, averaged over 1000 cross-validation runs.

The results are presented in Tab. 13.2. We hereby differentiate between the two transformation match scores S^{da} and S^{rmsd} to investigate potential differences in their performance. As can be seen, the coverage of S^{da} is marginally better than that of S^{rmsd} over all parameter sets and combinations thereof with 223, 250, and 258 vs. 221, 245, and 252 correctly reconstructed complexes for the best single, pair and triple of parameter sets using S^{da} and S^{rmsd}, respectively. The corresponding parameter sets for S^{da} and S^{rmsd} are highlighted in Tables C.7 and C.9. Moreover, the difference in coverage increases slightly from 0.6% over 1.6% to 1.9% when considering the best pair or triple of parameter sets.

This observation is confirmed by the corresponding cross-validation coverage cov_{cv}: first, the comparatively small deviation between the estimated cov_{cv} and the observed coverage for the respective best combination of parameter sets (maximum respective deviation 3.7%, found for S^{rmsd}, best triple set) demonstrates that none of the chosen parameter sets (or a combination thereof) is overly well adapted (overtrained) to the underlying set of complexes and that the parameter sets can be expected to perform similarly well on unseen data. Second, a comparison of the respective cross-validation coverages for the two transformation match scores underlines the initial observation that S^{da} performs slightly better than S^{rmsd}: the difference in cross-validation coverages ranges from 1.7% (best single) over 2.0% (best pair) to 2.3% (best triple) and is even a bit higher than the respective differences in the observed coverages.

However, scoring with S^{da} requires two parameters, i.e., a maximum allowed displacement and angular deviation between two transformations, whereas S^{rmsd} only requires a maximum RMSD deviation. Because this additional parameter can account for some of the variance in the underlying data set, S^{da} can be expected to better cover the underlying set of complexes than S^{rmsd}. Hence, using the S^{rmsd} instead of S^{da} can be considered an equally valid choice, especially given the fact that it relies on an exact computation of the difference between two docking poses.

13.1.4 *Evaluation on Comeau's Data Set*

In the previous section, we have assessed the quality of our selected parameter sets and shown that they are not prone to any over-fitting w.r.t. the respective data set they have been trained on. Thus, they represent valid choices to assemble previously unseen complexes.

To the best of our knowledge, there are no data sets that exhibit a diversity comparable to the one we established. To nevertheless evaluate our approach on previously unseen complexes, we use a data set of homo-hexamers used to benchmark the ClusPro Multi-Docking algorithm [193]. The respective data set, given in Tab. C.2 comprises 17 hexameric complexes containing one-layered rings as well as dimers of trimers and trimers of dimers. In that context, it is worth mentioning that two of these 17 complexes (1I40, 1NSF) are also contained in our benchmark data set, however using *foreign* monomers for the assembly. In the following, we will thus give all performance results once on all 17 and once on the non-redundant set of 15 complexes excluding the aforementioned two. The results are presented in Tab. 13.3.

The evaluation was performed using the best-performing single, pair and triple parameter sets from the benchmark runs, for both S^{rmsd} and S^{da}. When comparing these results to the performance of the respective parameter set combinations in the benchmark scenario (Tab. 13.2), we observe the following: first, with exception of

	S^{da}		S^{rmsd}	
Parameter set	All	Non-redundant	All	Non-redundant
Best single	12 (70.6%)	11 (73.3%)	13 (76.5%)	12 (80.0%)
Best pair	14 (82.4%)	13 (86.7%)	14 (82.4%)	13 (86.7%)
Best triple	15 (88.2%)	14 (93.3%)	12 (70.6%)	11 (73.3%)

Table 13.3: Performance of the selected best-covering single, pair and triple parameter sets for both transformation match scores S^{da} and S^{rmsd} obtained from the benchmark data set on Comeau's data set. Columns denoted by "All" take all 17 complexes into account, those denoted by "Non-redundant" disregard the two complexes 1I40 and 1NSF for which related entries are present in the benchmark data set.

two outliers (the best single parameter set for S^{da} and the best triple for S^{rmsd}) the observed performances are slightly better than in the benchmark case. Compared to the respective cross-validation coverage rates, none of the observed coverages can be considered significantly worse.

Besides the lower diversity of the test set, the major reason is the comparatively small size of the data set (17 in total and 15 when disregarding complexes with related entries in the benchmark data set). Hence, an additional complex that can be successfully reconstructed has a larger impact on the overall observed coverage rate. This is also the reason why the two outliers mentioned above have occurred: in the case of the best single parameter set for S^{da}, one additional successfully reconstructed complex would have yielded a better coverage rate than that obtained from the respective benchmark run. Similarly, this is true for the best triple for S^{rmsd}, though in this case two additional complexes would have to be successfully reconstructed.

This observation is also underlined by the coverage rates for the non-redundant set: here only 15 complexes are considered. Whereas 1NSF could consistently not be reconstructed among all (combinations of) parameter sets, the opposite is true for 1I40. Hence, the overall data set size is decreased by two complexes, from which one could not be reconstructed. As a consequence, the obtained coverage rates are slightly higher.

Summarizing, the overall results, even though obtained on a small and less diverse data set, are comparable to those of the corresponding cross-validation coverage rates determined in the previous subsection and none of the presented results is significantly worse than the corresponding cross-validation coverage cov_{cv} presented in Tab. 13.2. This demonstrates that our selected parameter sets are indeed good choices and are not overtrained on the benchmark data set.

13.1.5 *The Importance of the Transformation Match Score*

The core component of our algorithm is the transformation match score (*tms*) that we developed in Ch. 10. To differentiate its influence on the performance from that of the clustering procedure, we disabled the intra- and post-clustering and compared how the algorithm performs on the benchmark set when the transformation match score is enabled or disabled.

Over all 29,568 baseline runs with enabled transformation match score (S^{da} and S^{rmsd}), only 16 cases (0.05%) did not produce any solution compared to 8 failed runs

Parameter set	With *tms*, no clustering	No *tms*, no clustering	No *tms*, but clustering
Worst single	155	44	62
Best single	184	60	110
Best pair	208	69	125
Best triple	216	73	128
Total	243	76	135

Table 13.4: Comparison of baseline runs, demonstrating the superiority of 3D-MOSAIC runs with enabled transformation match score.

(0.3%) for the 2,464 runs without *tms*. Among the 96 parameter sets with *tms*, the worst single parameter set could still reconstruct 155 complexes as compared to the best parameter set without *tms* with 60 correct solutions (improvement factor i_f of 2.58), demonstrating that there is a large performance difference in both scenarios. The worst single parameter set without *tms* can generate even fewer correct solutions (44), the best one with enabled *tms* 184.

When choosing the best-performing triple (pair) combination of parameter sets without transformation match score, we could correctly reconstruct 73 (69) complexes (with *tms* 216 (208), $i_f = 2.96$ (3.01)). The average ranking of the first correct solution per assembly was also significantly improved when *tms* was enabled: 3.23 compared to 8.92 for the best performing parameter set and 7.02 (5.70) vs. 29.16 (22.07) for the best-performing triple (pair) combination thereof. The average ranking for the best single parameters set is 2.17 for the *tms*-enabled one and 8.63 without *tms*.

In total, without clustering and *tms* enabled, 243 complexes could be reconstructed (76 without *tms*).

We can also compare the results of runs with enabled *tms* but disabled intra- and post-clustering to runs where *tms* is disabled, but clustering is enabled. In doing so, we can gain insight into whether the clustering procedure can compensate the effects of the transformation match score.

For the 16 parameter sets (4,928 assembly runs) performed in the latter case, we obtain the following values: 16 runs (0.3%) failed without producing any solution. In total, 135 complexes could be correctly reconstructed. The best single parameter configuration yielded 110 correct complexes ($i_f = 1.67$). With the best pair and triple of parameter sets 125 ($i_f = 1.66$) and 128 ($i_f = 1.69$) complexes could be correctly reconstructed.

The average ranking for the first correct solution obtained with best single, pair, and triple parameter set is 6.76, 12.98 and 12.35, respectively.

A summary of the comparison of the baseline runs, demonstrating the power of the transformation match score, is given in Tab. 13.4. Detailed information for the runs with enabled *tms*, split into those employing S^{da} and S^{rmsd}, can be found in Tables C.16 and C.18, respectively. Full accounts on those runs with disabled *tms* and with clustering en-/disabled are given in Tables C.21 and C.22.

13.1.6 *Symmetry Optimization and Ranking of Assembled Complexes*

As already stated in Sect. 9.5, many protein complexes exhibit at least partial symmetries. However, when assembling unknown complexes, the respective type of symme-

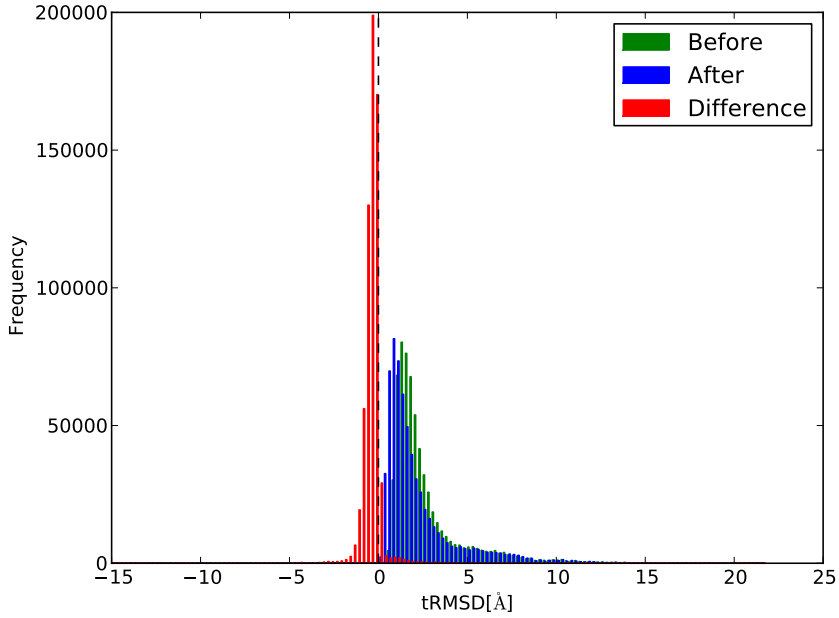


Figure 13.7: The effect of symmetry optimization on the tRMSD to the reference complex. The distributions of tRMSD after and before symmetry optimization as well as their difference are shown.

try is also often unknown. Hence, 3D-MOSAIC assembles complexes without prior assumptions about the complex symmetry. Nevertheless, after completed assembly of the complex, the symmetry can often be guessed and optimized (see Section B.10).

The symmetry-optimized assembled complexes can be expected to be more similar to the native complex than the corresponding solution before symmetry optimization. To investigate whether this assumption is valid, we determined all symmetry-optimized complexes present in the benchmark assemblies (see Subsect. 13.1.2) and calculated the respective difference between the tRMSD after and before the symmetry optimization.

The results are presented in Fig. 13.7. In total, over all 29,568 runs, N=633,922 symmetry-optimized complexes were found. The corresponding distributions of tRMSD before and after the symmetry optimization look similar at first sight, especially when considering the corresponding means and standard deviations presented in Tab. 13.5a). However, they are not: since our data can be understood as pairs of corresponding tRMSDs, an appropriate test statistic is given by the Wilcoxon signed-rank test [366]. This method tests whether the null hypothesis that corresponding pairs of values represent samples from the same distribution is true. The obtained p-value was found to be below the floating point accuracy of MATLAB, hence p-value $< 5e - 324$, demonstrating that both distributions are significantly different.

From Tab. 13.5a, we can also see that on average, the symmetry optimization yields a tRMSD improvement of 0.4Å and thus has a large impact on the structural quality of the resulting complexes.

Consequently, the question whether a complex is symmetry-optimized, or rather how many symmetry mappings (cmp. Def. 9.7) were obtained for a particular complex, can serve as a ranking criterion, because the number of symmetry mappings that can be obtained for a complex candidate (m at most for a complex with m monomers) w.r.t. a given (reasonable) RMSD threshold can be considered an indicator of the qual-

	tRMSD	
	Mean	σ
Before	2.66Å	2.06Å
After	2.26Å	2.20Å
Δ	-0.40Å	0.55Å

(a) Main characteristics of the distributions of tRMSDs for complexes after and before symmetry optimization and their respective difference.

	Rank	
	Mean	σ
Before	4.14	11.65
After	3.19	9.73
Δ	-0.95	7.44

(b) Main characteristics of the rank distributions after and before symmetry-based re-ranking and their respective difference.

	Rank	
	≤ 10	≤ 25
Not re-ranked	92.8% (1.8%)	95.7% (1.2%)
Re-ranked	94.9% (2.4%)	97.1% (1.6%)

(c) Average percentage of complexes per benchmark run for which a near-native solution was found among the top 10 and 25 ranks

Table 13.5: Effects of symmetry optimization and symmetry-based re-ranking.

ity of the solution: given two solutions for which we have determined the number of symmetry mappings, the one with the greater number of symmetry mappings typically exhibits a more regular topology (otherwise, less symmetry mappings would have been found). Hence, the number of symmetry mappings also provides information on the quality of the solutions.

By default, 3D-MOSAIC ranks all complexes by tRMSD, however, the number of obtained symmetry mappings of each symmetry-optimized complex are also returned. With this knowledge, we can re-rank the solutions of a particular 3D-MOSAIC run according to the following criteria, in decreasing priority: i) by the number m of symmetry mappings, ii) for complexes with equal m by tRMSD, and iii) complexes equal in i) and ii) by accumulated interaction score of all monomers. W.r.t. this ranking, we then determined the rank of the first near-native structure with an tRMSD $\leq$ 2.5Å.

The characteristics of the obtained distributions are presented in Tab. 13.5b. Again, we applied a Wilcoxon signed-rank test for pairs of ranks before and after symmetry optimization and obtained a p-value of $7.3e-242$, showing that the two distributions are significantly different. On average, the ranking of the first near-native structure was improved by 1, with an standard deviation of 7.4.

In absolute numbers, using the symmetry-based re-ranking scheme, we could improve the rank of the first near-native solution in 2,149 cases, a worsening was observed in 830 cases. In Tab. 13.5c, we present the rank improvement in dependence of the number of benchmark runs for which a near-native assembly was obtained at all, i.e., at any rank. The results are first averaged over all successful assemblies per parameter set, subsequently over all parameter sets.

The first remarkable result is that, even without re-ranking, in 92.8% (95.7%) of the benchmark runs, a near-native complex with a tRMSD $\leq$ 2.5Å can be found among

F_3	MF	N	Signature
1.00	0.93	5	All beta proteins | Small proteins
1.00	0.90	7	Small proteins
1.00	0.73	6	Multi-domain proteins (alpha and beta)
0.86	0.81	7	All alpha proteins | Alpha and beta proteins (a+b)
0.87	0.78	15	All beta proteins | Alpha and beta proteins (a/b)
0.85	0.77	13	Alpha and beta proteins (a+b) | Alpha and beta proteins (a/b)
0.78	0.68	32	All alpha proteins
0.77	0.68	82	Alpha and beta proteins (a/b)
0.66	0.59	58	Alpha and beta proteins (a+b)
0.62	0.57	39	All beta proteins
0.61	0.55	44	Others

Table 13.6: The performance of 3D-MOSAIC w.r.t. to the SCOP class signatures. F_3 denotes the fraction of structures in the respective signature for which a successful reconstruction could be obtained in at least one third of the parameter sets, MF the fraction of parameter sets for which a successful reconstruction of the respective assembly could be obtained, averaged over all structures in the respective signature, and N the number of assemblies with corresponding signature.

the top 10 (25) ranks. These outstanding values are even improved upon symmetry-based re-ranking: here, in 94.9% (97.1%) of the cases where a correct assembly could be obtained, the first near-native solution is located among the top 10 (25) solutions, yielding an improvement of 2.1% (1.4%)

Hence, from these observations, we can not only summarize that a symmetry-based re-ranking can improve ranking performance, but, more importantly, that if 3D-MOSAIC could generate a near-native solution, it is almost always located among the top 25 solutions.

13.1.7 *Performance w.r.t. SCOP Class Signature*

To assess how the performance of 3D-MOSAIC depends on the general structural properties, i.e., the secondary and supersecondary structure elements the complexes are composed of, we determined the SCOP class signature of each complex (in analogy to the SCOP superfamily signature determination described in Section 12.1). The results are presented in Table 13.6; all signatures containing less than five members were combined into one signature termed "Others".

The general observation is that 3D-MOSAIC performs particularly well for signatures comprised of more than one SCOP class (as well as multi-domain proteins): F_3, the average number of structures for which a correct reconstruction could be obtained in at least one third of the parameter sets (which corresponds to applying 3D-MOSAIC with three randomly selected parameter sets) ranges from 100% to 85%. Likewise, MF, the fraction of parameter sets average over all complexes in that signature for which a correct reconstruction could be obtained (corresponding to randomly selecting one parameter set for use with 3D-MOSAIC) ranges from 93% to 68%. One reason for the good performance in such classes may be that complexes comprising different secondary and supersecondary structure elements may provide more spe-

cific interfaces for the individual binding modes. Furthermore, signatures including small proteins comprise the top two ranks: because we generate 10,000 docking poses per binding mode regardless of the size of the involved proteins, the interfaces of small proteins may be sampled more densely. In addition, the interfaces in small proteins are closer to each other, which may additionally restrict the space of compatible solutions.

A comparatively bad performance is observed for complexes comprising monomers that consist only of β-sheets or exhibit β-sheets which are separated from (and not intermixed with) the α-helices in the respective domains. These structures often form additional β-sheets with other monomers, however the non-specific interface in terms of surface complementarity and the comparatively low tolerance in the orientation of the two monomers to each other for the interaction to be established represents a challenge to many docking algorithms in general and particularly RosettaDock in our case. Yet, we still observe a successful reconstruction in at least one third of the parameter sets for 62% of the complexes (F_3), and on average for 57% of the complexes when using only one parameter set (MF).

13.1.8 *Limitations and Hard Cases*

Due to the diversity of our data set, a general rule on limitations and hard cases is difficult to obtain. While for example graph-based measures like average shortest path length between monomers (with interacting monomers connected by an edge) in a complex can give at least a weak indication of difficult cases (the longer the shortest path the less likely a transformation matching), they cannot be used to predict the performance beforehand. The reason is that in an application scenario, the real complex topology is unknown and can hardly be predicted from binary docking poses ahead of the actual assembly process.

However, a comparatively simple measure deduced from the stoichiometry of the protein types and their interfaces can provide some information on how well a complex can be assembled. This is the minimum number of interfaces over all protein types (degree), deg_{min} divided by the number of monomers in the complex with the same degree n_{min}:

$$r_{min} := \frac{deg_{min}}{n_{min}} \tag{13.1}$$

The rationale behind this measure is as follows: the smaller the minimum degree and the more monomers share that degree, the more difficult can we expect the assembly process to be. In particular, the degree contains some information about the immediate neighborhood of the respective protein type: if the degree is small, many of the docking poses lead to a valid attachment, because in a sparse neighborhood, only few potential attachments lead to clashes.

We compare this measure to the average ROC AUC (cmp. Subsections 3.7.2 and 3.7.3) for each complex over all benchmark parameter sets. We thus incorporate the information on the ranking and the distribution of correct solutions among the set of all generated solutions. An $AUC > 0$ denotes that a solution has been found, with smaller values corresponding to cases where the first correct solutions tend to be found among the last ranks. A value of 0.5 denotes an equal true and false positive

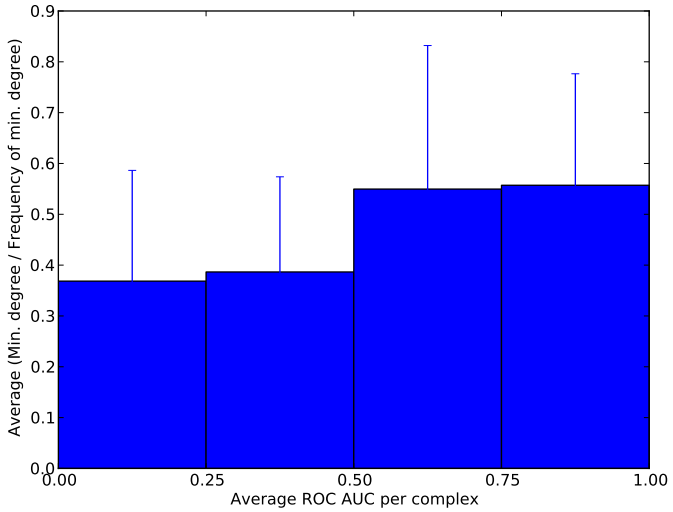


Figure 13.8: The mean AUCs of the four difficulty classes including the respective standard deviations.

rate among all ranks, and values close to 1 represent cases where correct solutions outbalance false ones. Fig. 13.9 shows the ROC AUCs over all benchmark runs presented in Subsect. 13.1.2.

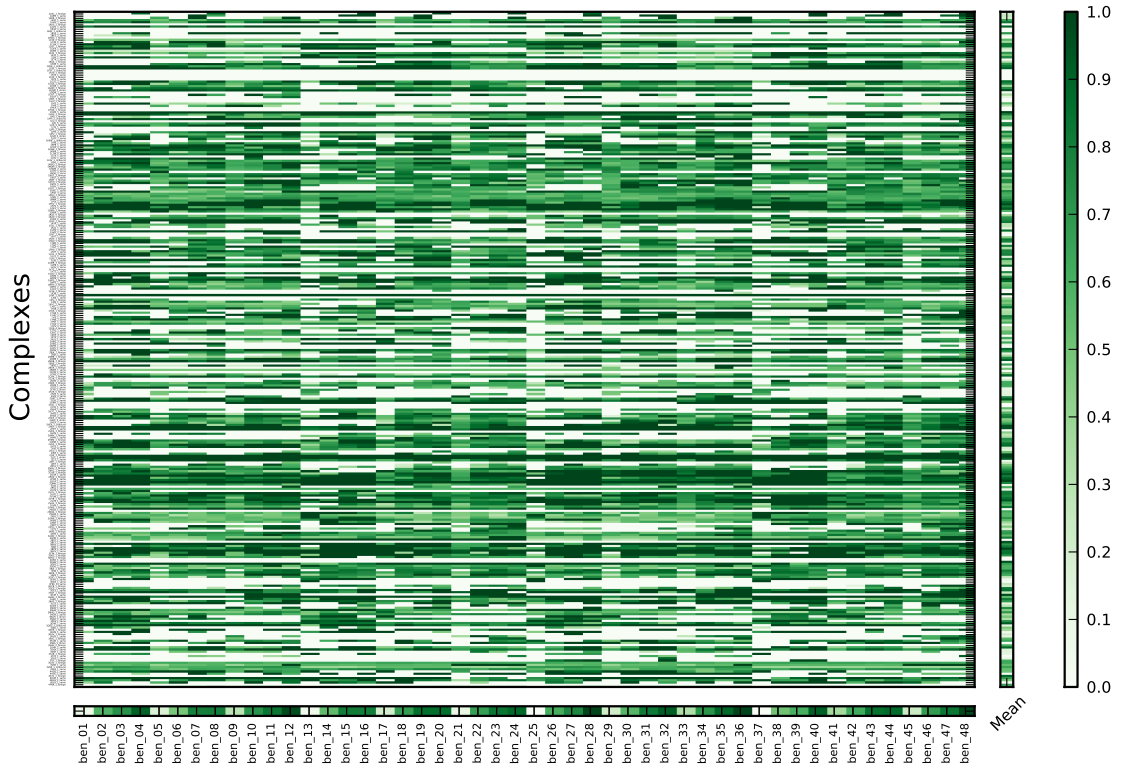
The mean AUCs (small vertical bars) show a wide distribution, with some complexes performing especially well and others where correct solutions are scarce. We classify the complexes into four groups of difficulty w.r.t. the mean AUCs: 1) complexes with mean AUC ≤ 0.25, 2) > 0.25 and ≤ 0.5, 3) > 0.5 and ≤ 0.75, 4) > 0.75 and ≤ 1. The average AUCs of each of these groups are presented in Fig. 13.8.

We see that these four classes fall into two categories, those with a mean AUC less or equal to and those above 0.5. A Wilcoxon rank-sum test [366] over all pairs of classes confirms this observation: all p-values except those between class 1 and 2 (p-value 0.52) as well as 3 and 4 (p-value 0.33) are below statistical significance w.r.t. significance level $\alpha = 0.05$ (max. p-value 2.29e-05).

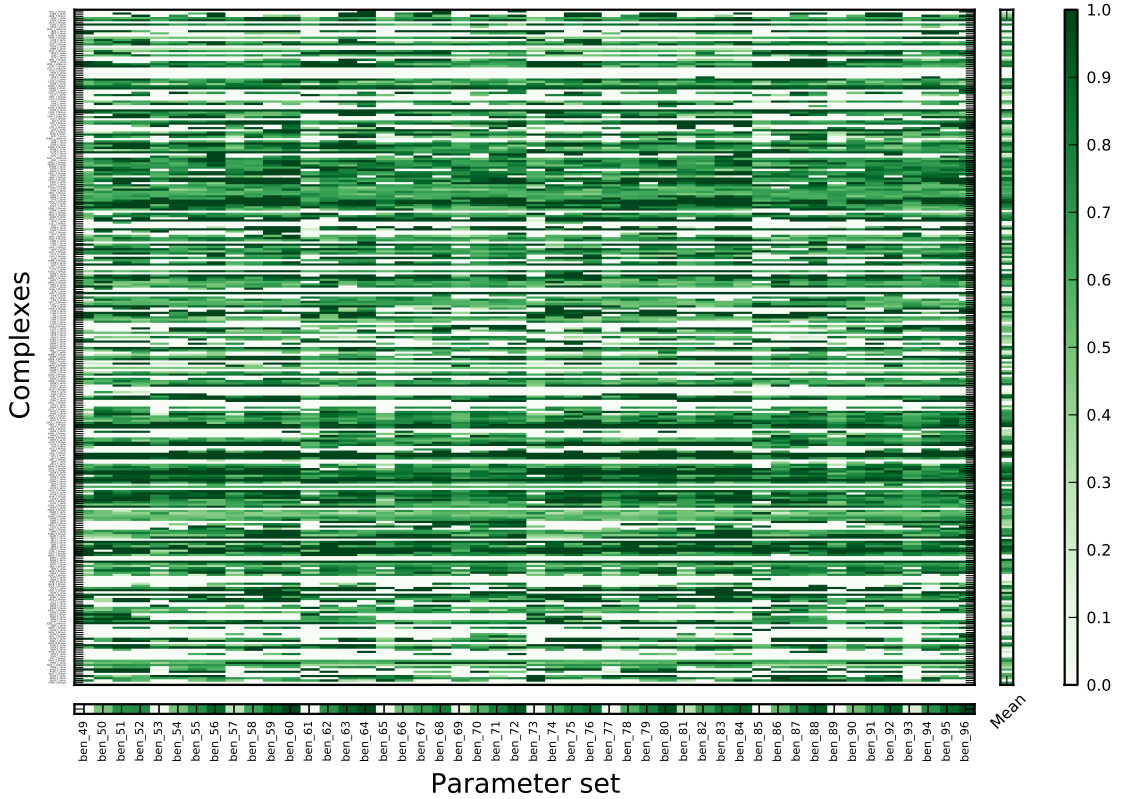
Consequently, we can state that our measure r_{min} provides at least some information on how well a particular complex can be assembled. Especially for cases where r_{min} is below approximately 0.25, we can expect that the correct solutions are hard to obtain.

This is especially true for two complex topology classes: one-layered rings and cage-like complexes. An example of a one-layered ring, where many poses lead to a valid attachment of monomers, and the corresponding topology are shown in Figs. 13.10a and 13.10b, respectively. Similarly, cage-like complexes mainly comprise assemblies that consist of patches of small sub-complexes, for example trimers, with sparse connections between these sub-complexes. They are related to ring-like structures, because here, rings are formed across sub-complex patches. An example, the pyruvate dehydrogenase complex E2 core (cmp. Subsect. 2.5.1) and its topology are shown in Figs. 13.10c and 13.10d, respectively.

In addition to these topology classes, the we have also observed a poor performance for complexes with monomers that are mostly helical or that are heavily intertwined with other monomers. In the former case, the reason is that the monomers show virtually no surface complementarity, hence, a large variety of different poses are nearly equally likely and near-native poses are hard to discriminate from decoys (see Figs. 13.10e and 13.10f). In the latter case, the intertwining greatly reduces the number of

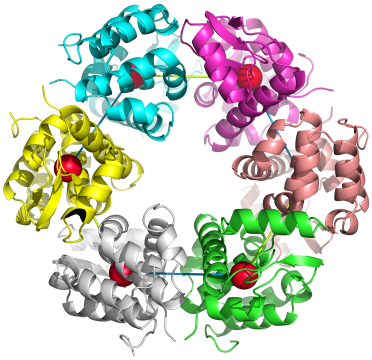


(a) Parameter sets using S^{da}

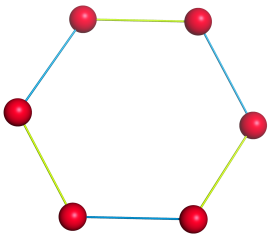


(b) Parameter sets using S^{rmsd}

Figure 13.9: ROC AUCs of all benchmark runs: each column corresponds to one parameter set, each line to one complex; matrix entries colored with darker green shades denote larger ROC AUC values, as specified by the color bar on the right. The small bars below and to the right of the main plot represent the corresponding average ROC AUCs per parameter set and per complex, respectively.

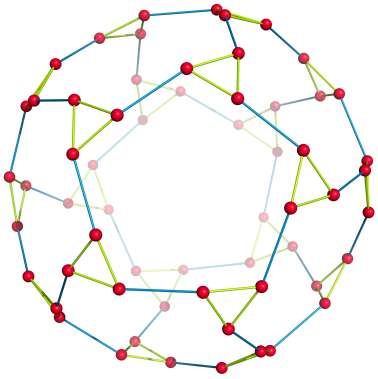


(a) T4 lysozyme hexamer (3SBA)

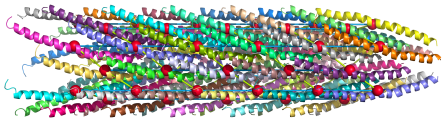


(b) Ring-like topology of 3SBA

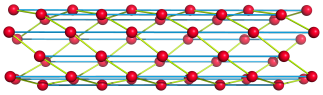
(c) Pyruvate dehydrogenase E2 60-mer core complex (1B5S)



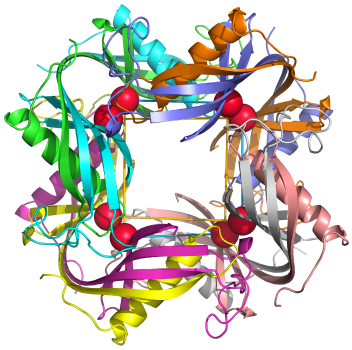
(d) Cage-like topology of 1B5S



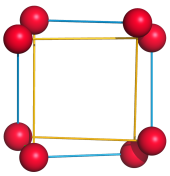
(e) Helical monomers of inovirus coat protein filament (2CoW)



(f) Topology of 2CoW



(g) Interchain β-sheet of human cystatin C (1R4C)



(h) Topology of 1R4C

Figure 13.10: Examples of complexes and corresponding topology graphs for hard cases. Node colors correspond to protein types, edge colors to binding modes.

docking poses that are compatible and most of them will likely to lead to steric clashes. Such an example, where the intertwining leads to a formation of interchain β-sheets, and the corresponding topology are shown in Figs. 13.10g and 13.10h, respectively. Yet, typically, in such cases, the dimeric structure interacts so strongly that the dimer instead of the monomers can be used for the assembly.

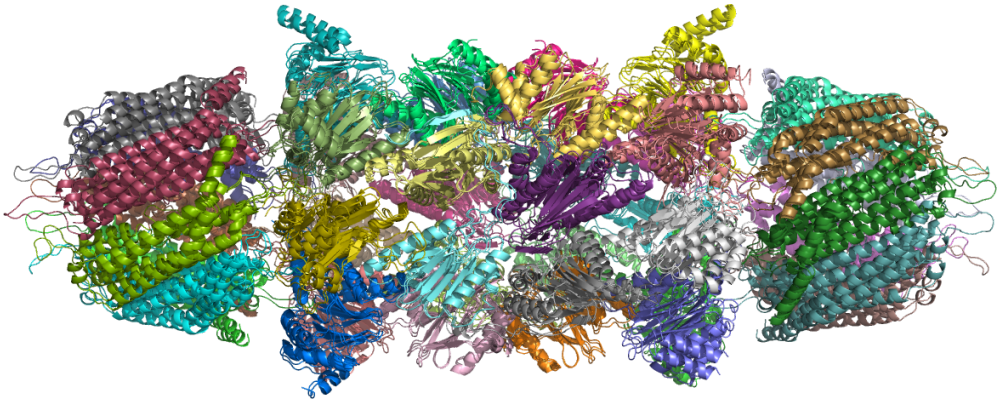
13.1.9 *Examples of Successful Assemblies*

In the previous section, we have presented the most prominent factors that can hamper the assembly process and the generation of near-native solutions. Yet, many of such difficult cases could also be reconstructed across different parameter sets, despite the complex size or the number of different protein types. In particular, in addition to the 278 correctly reconstructed complexes in the benchmark scenario, near-native solutions for further 8 complexes were observed in the baseline runs used to demonstrate the effectiveness of the transformation match score (Subsect. 13.1.5), summing up to 286 out of 308 (93%) complexes with a near-native solution. In the following, we shortly discuss some examples whose assembly is beyond the capabilities of current methods (Ch. 8) in terms of complex size, monomer sources and number of protein types, but could be achieved by 3D-MOSAIC.

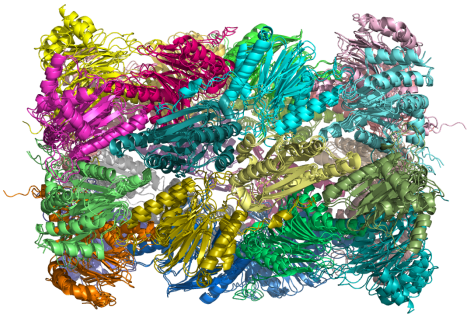
One of the most prominent examples that we already introduced in Subsect. 2.5.2 is the proteasome. Our benchmark set comprises three proteasome versions: the 20s proteasome from yeast complexed with proteasome activator PA26 (1Z7Q, 15 protein types, 42 monomers) in the *same* data set, as well as the stand-alone 20s yeast proteasome (1RYP, 14 protein types, 28 monomers) in the *same* and *foreign* sets. In the latter case, five different monomer sources are used (1Z7Q, 1FNT, 3L5Q, 3UN4, 1VSY), which is the maximum number over all structures in the *foreign* set. All of these complexes exhibit a two-fold symmetry, because the minimum stoichiometry over all protein types is 2, hence the effect of the symmetry optimization on the structural quality of the final model is comparatively small, making the generation of good-quality models during the iterative assembly stage especially important. The best tRMSDs for the three complexes are 0.78Å (Fig. 13.11a), 0.68Å and 0.67Å (Fig. 13.11b). The latter result is insofar remarkable as an assembly of equal quality could be obtained regardless of the source(s) from which the complex monomers originate.

Another interesting example, composed of 11 different protein types, each with a stoichiometry of 2, is the cytochrome BC 1 complex (pdb code 1BE3, tRMSD 0.91Å) which is located in the membrane of the mitochondria, and is part of the final stages of energy recovery in the electron transport chain [367]: it serves as a proton pump which ultimately sets the ATP synthase motor [368, 369] into rotation. This motor can then synthesize ATP, the universal energy storage molecule. 1BE3 is of special importance, because it represents a trans-membrane protein complex, a class of proteins which are typically hard to determine structurally because a removal of the membrane upon crystallization often leads to a denaturation of the protein complex and thus causes the crystallization to fail in many cases.

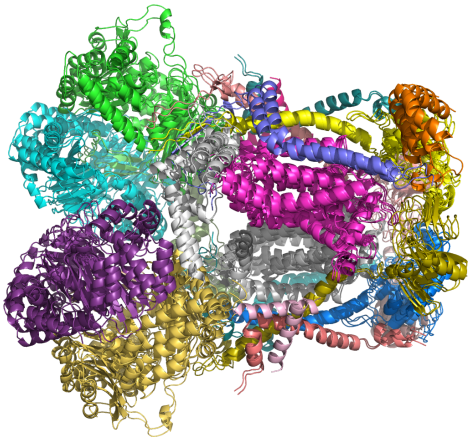
The next example is sTALL-1 (pdb code 1JH5), a member of the tumor necrosis factors family (TNF) which are, amongst others relevant for tumor regression. The structure we present here (Fig. 13.11d) has been experimentally determined to form a homo-60-mer with a virus-like appearance [370]. 1JH5 is especially remarkable, because it was found to be the complex for which an assembly with the lowest tRMSD



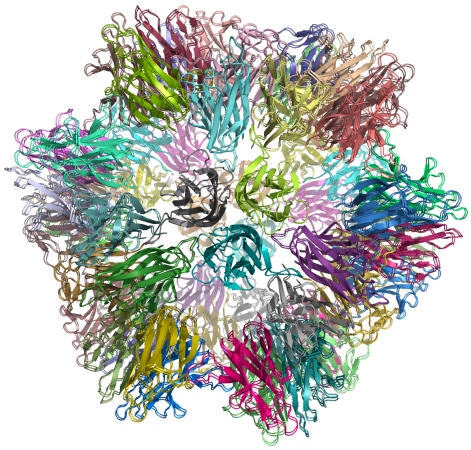
(a) 1Z7Q, *same*, tRMSD 0.78Å



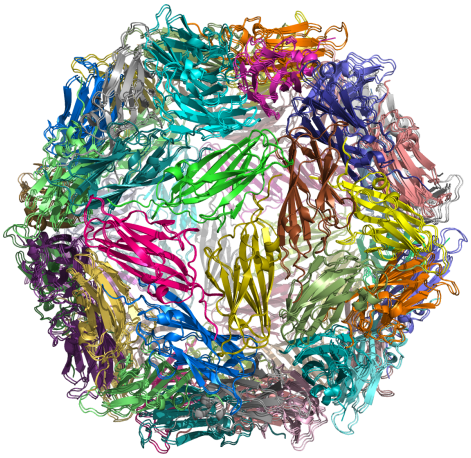
(b) 1RYP, *foreign*, tRMSD 0.67Å



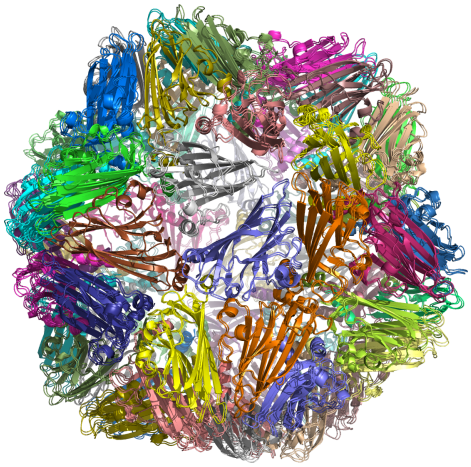
(c) 1BE3, *same*, tRMSD 0.91Å



(d) 1JH5, *same*, tRMSD 0.06Å



(e) 1STM, *same*, tRMSD 0.23Å



(f) 2BUK, *foreign*, tRMSD 0.31Å

Figure 13.11: Examples of successful assemblies obtained with 3D-MOSAIC superimposed onto the corresponding reference complex

over all complexes and parameter set was generated (0.06Å). despite the fact that it is one of the largest complexes in our data set, both in terms of the number of monomers and the diameter ($\approx$ 200Å).

Finally, the structural determination of viral capsids is an important field, because these capsids represent potential target sites for anti-viral drug treatment (see Subsect. 2.5.3). Figures 13.11e and 13.11f show two examples: the capsids of the satellite panicum mosaic virus (pdb code 1STM) and satellite tobacco necrosis virus (2BUK), with respective tRMSDs of 0.23Å and 0.31Å. Both of them are composed of 60 monomers with two-, three- and five-fold rotational symmetries and show three different binding modes.

13.1.10 *Running Times and Memory Consumption*

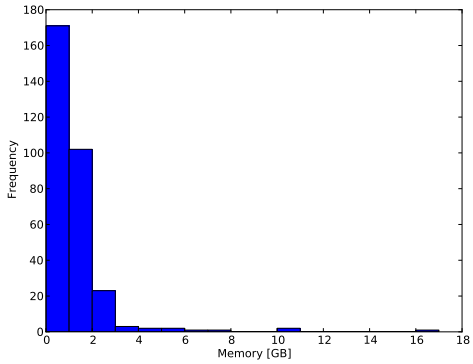
In the previous subsections, we have demonstrated the success of 3D-MOSAIC in assembling macromolecular oligomeric complexes from pairwise docking data. In this subsection we will discuss the running times and memory requirements of the algorithm. The corresponding results are summarized in Fig. 13.12, all experiments were carried out on the high-performance cluster MOGON (see Sect. 12.6.4 for details).

The distribution of maximum memory requirements per complex is presented in Fig. 13.12a. As can be observed, the largest fraction of complexes (273, 88.6%) requires less than 2GB for the assembly. The required memory is largely dependent on the input data, i.e., the number of docking poses. In the benchmark scenario, we use a constant number of 10,000 docking poses per interface, consequently, complexes with many interfaces require more memory. In particular the hetero-complexes 1Z7Q (15 protein types, 50 binding modes), 1RYP (14 protein types, 35 binding modes) and 1BE3 (11 protein types, 25 binding modes) require 16.22, 10.1 and 7.15 GB. Similarly, the preparation times (Fig. 13.12b) are dominated by the time required to insert the docking poses into the hash maps which also grows w.r.t. the number of interfaces.

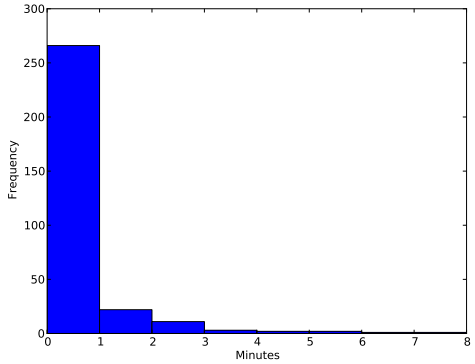
The population of a level, i.e., the attachment of new monomers to all solutions retained from the previous level (Subsect. 11.2.5), mainly depends on the number of allowed clashes per pair of monomers as well as the the parameters for the transformation match scores. Figs. 13.12c and 13.12d show the mean population times per level and the respective accumulated population time up to a particular level. Each data line represents the average over all complexes w.r.t. a particular combination of number of allowed clashes and the allowed max. displacement l_{max}. Corresponding data lines of S^{da} and S^{rmsd} using the same l_{max} value were combined, because no significant difference in running time was observed.

The dashed vertical lines indicate the following: the area between dashed lines at 1 and 6 (area 1), the initial number of 2000 retained solutions is reduced by a factor of 2 after each level until the lower bound of 100 is reached (area 2). The dashed lines at 20 and 40 (areas 3 and 4) correspond to a reduction of the clustering parameters as well as the number of solutions and best-matching poses that are considered for further assembly (see Subsect. 12.6.1).

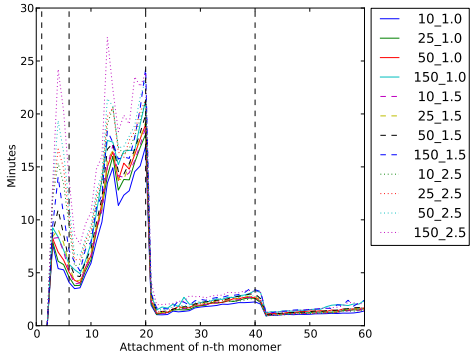
In general, we observe that the running time per level strongly depends on the max. allowed displacement. In particular, the data lines with $l_{max} = 2.5$ dominate the other lines. As a second criterion the number of clashes is relevant, with a smaller running time for smaller clash values. The solution reduction scheme strongly affects the overall running time per level: the attachment of the first monomer to the initial



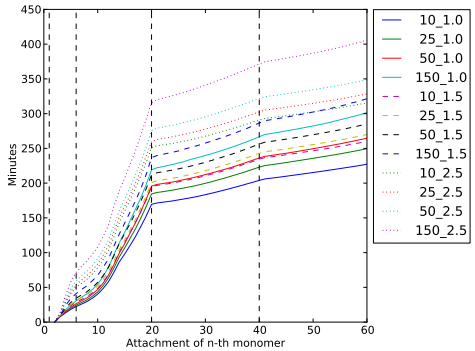
(a) Distribution of maximum memory requirements over all complexes.



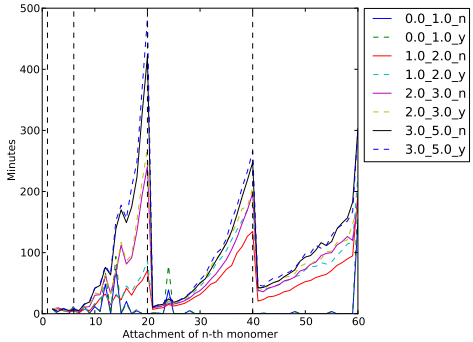
(b) Distribution of mean preparation times over all complexes.



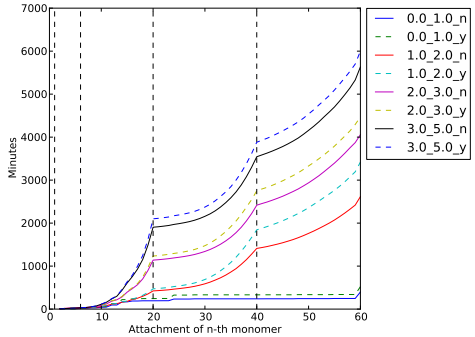
(c) Mean population times per level w.r.t. max number of allowed clashes and displacement l_{max}.



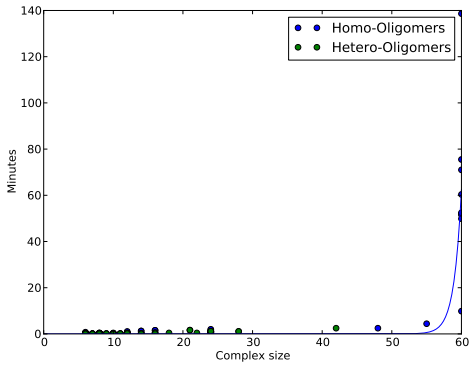
(d) Mean accumulated population times w.r.t. max number of allowed clashes and displacement l_{max}.



(e) Mean finalization times per level w.r.t. intra- and post-clustering C_α RMSD and ligand interpolation (y/n).



(f) Mean accumulated finalization times w.r.t. intra- and post-clustering C_α RMSD and ligand interpolation (y/n).



(g) Mean evaluation times per generated solution w.r.t. complex size.

Figure 13.12: Memory requirements and running times (for used hardware see Sect. 12.6.4). In the area between dashed lines at 1 and 6, the number of solutions is reduced by a factor of 2 in each iteration, starting at 2000, ending at 100. The vertical dashed lines at 20 and 40 correspond to a reduction of the clustering parameters and the number of matching poses used for further assembly.

receptor considers all docking poses and thus can require up to 25 minutes on average. Subsequently, the reduction in the number of retained solution leads to a decrease of running times, even though the number of potential interfaces that can be investigated as well as the pairs of monomers that have to be checked typically grows with the number of monomers in the sub-complex.

In area 2, the number of solutions per level is constantly kept at 100. Here, the running times increase again due to the increasing number of interfaces and clash checks to be performed. The drop after the attachment of the 14th monomer results from a completion of 10% of the remaining complexes. Typically, population times in the last stages before completion take longer because here the number of clash checks to be performed with neighboring monomers is commonly the highest.

After monomer 20, the number of solutions is set to 50 and only the 500 best-matching poses from initially 10,000 per interface are considered. The effects on the running time are two-fold: i) a 95% smaller solution space per interface, consisting only of the most reasonable docking poses, has to be investigated and ii) these poses typically fit well and thus less clash checks are required.

After monomer 40, both of the above values are reduced again to 25 and 250, again leading to a drop. However this drop only occurs stepwise until monomer 42. The reason here is, similar to the peak at monomer 14, that 1Z7Q is about to be finished, again leading to higher number of clash checks. After monomer 42, the drop is complete and the curves slightly increase again.

The finalization times show a similar behavior (Fig. 13.12e and 13.12f). Here, the main parameters are the values for intra- and post-clustering as well as the choice whether ligand interpolation should be enabled. To obtain more detailed information, we applied additional runs with smaller clustering parameters than those used in the benchmark scenario. In particular, the RMSD thresholds for intra-/post-clustering were set to 0.0Å/1.0Å and 1.0Å/2.0Å. The former is a special case insofar as a clustering is only applied when a complex has been completely assembled, leading to distinct peaks in the corresponding data lines at the respective locations in the plot.

In general, the higher the clustering parameters, the higher the running times. Enabling ligand interpolation leads to higher running times, because the interpolation leads to the generation of solutions that are more similar than in the runs using no interpolation. In area 1, the mean running times remain constant, because after each iteration less solutions have to be clustered to determine the set to be retained for the next iteration. In area 2, the running times then increase because the number of complex similarity mappings rapidly grows with the complex size. A drop is again observed at the beginning of area 3 and 4, because here, the clustering parameters are reduced by a factor of 5 and then again by 2. The reason for this reduction is, that the overall RMSD between two complexes is less and less affected by the attachment of an additional monomer, the more monomers the complex already contains. Yet, the running time required for clustering is soon again dominated by the actual complex size and the number of potential mappings between two complexes that can be obtained.

Consequently, also the running times for evaluation and symmetry optimization (Fig. 13.12g) depend on these mappings, even three-fold. First, an evaluation against the reference complex is performed (Sect. B.11), subsequently a symmetry optimization is attempted (Sect. B.10) and if successful, an evaluation against the reference is performed again. Because the symmetry optimization performs up to twenty iterations by default, trying partial symmetry optimizations until either a clash-free com-

plete set of symmetry mappings is obtained or the iteration threshold is reached, this heavily affects the running time for large complexes. However, the use of restart files allows for a heavily parallel evaluation of all solutions obtained in the final iteration of complex assembly.

The running times required to generate the docking poses are not considered: they strongly depend on the employed docking algorithm, its parameterization, the number of atoms of the monomers to be docked and the number of poses to be generated. In our case, we employed RosettaDock in low-resolution mode and generated 10,000 dockings. Depending on the proteins to be docked, we observed running times from several hours to two days when applying serial docking. However, in a real case study where the poses have to be generated for only one instead of 308 complexes, the docking with RosettaDock can be parallelized, and thus the required running time can be drastically reduced.

Summarizing, the assembly process is heavily dominated by the finalization stage, in particular the clustering. For complexes of size 60, the running time in the finalization stage is 100 times greater than that of the population stage, when using the parameters that perform worst in the respective part of the algorithm. Consequently, in the average worst case, the accumulated finalization time of about 6,000 minutes (4.2 days) for a complex of size 60 is approximately 15 times greater than that of the population stage.

However, all 60-mers but 1HQK could be reconstructed even in the additional runs using small clustering parameters. Hence, several additional scenarios not tested in this work are thinkable: for example, clustering is only applied in the cheap early stages of the algorithm to diversify the solution space, and later stages do not apply clustering at all. Alternatively, clustering could only be applied in every k-th iteration to decrease the overall running times. The latter is especially important in cases where the solutions obtained from the population are very similar. In particular, we encountered a few cases (less than 15 in total for complexes 1KIB, 2F1D and 3KA3), where the finalization took unusually long when employing ligand interpolation. Here, using the restart feature, one would typically reduce the clustering parameter or the number of solutions to be retained; however we refrained from introducing such changes to keep the parametrization consistent over all benchmark runs.

13.2 SINGLE-RESIDUE PAIR INTERACTION CONSTRAINTS

In this section, we concentrate on a scenario where only little information about potential binding modes is available. In particular, we assume that we only know one pair of interacting residues per native binding mode in the complex as described in Section 12.3, for example from correlated mutation studies [322, 323, 197, 199]. To further increase the difficulty in this scenario, we also included additional false-positive (non-native) binding modes. Such binding modes can for example have been observed in homologous proteins, obtained by protein-protein interaction prediction tools such as PRISM [371], or correspond to native binding modes which are, however, not present in the complex (Sect. 12.3).

In this section, we thus want to assess how well 3D-MOSAIC performs on docking poses obtained using wide-range sampling parameters under consideration of constraints derived for residues assumed to interact, so-called single residue-pair interaction constraints (SRPIC), as explained in Section 12.4.

Structure	Binding Mode A	B	C
1HI9	0.165 (124)	0.567 (68)	0.839 (68)
1KW6	1.138 (62)	0.351 (162)	0.792 (188)
1PVV	1.938 (38)	1.251 (56)	2.299 (6)
1QK1	1.053 (100)	0.670 (74)	0.491 (30)
1X1O	- (0)	2.280 (8)	0.615 (6)
1YNB	2.343 (4)	0.347 (96)	0.653 (58)
2BJK	- (0)	- (0)	0.202 (50)
2F1D	1.026 (104)	0.546 (104)	1.580 (108)
2UYU	- (0)	0.723 (34)	1.034 (274)
3Q46	0.167 (346)	0.461 (122)	0.195 (108)

Table 13.7: Docking results for the respective three native binding modes of the ten complexes selected for the SRPIC experiments. Per structure and binding mode, the C_α RMSD of the best pose (and the number of poses) with a C_αRMSD $\leq$ 3.0Å is given.

Due to the increased computational effort required for both the docking pose generation and clustering as well as the subsequent assembly, we randomly selected ten homo-oligomers, each with three native binding modes, which proved to be successful in the benchmark experiments. The following complexes (with number of monomers) where selected: 1HI9 (10), 1KW6 (8), 1PVV (12), 1QK1 (8), 1X1O (6), 1YNB (6), 2BJK (6), 2F1D (24), 2UYU (8), 3Q46 (6).

In the following, we first discuss the results obtained from the dockings for the individual binding modes. Subsequently, we investigate the overall performance of 3D-MOSAIC using these docking poses. In particular, we address how the performance of 3D-MOSAIC changes when only the docking poses corresponding to the native binding modes are used or when additional docking poses corresponding to false-positive (non-native) binding modes are included. Finally, we present the results of the cross-validation we performed on the employed parameter sets, once when using only docking poses that correspond to native binding modes, and once when additionally using docking poses corresponding to false binding modes.

13.2.1 *Pairwise Docking Results*

In this subsection, we present the results obtained from the docking of the start dimers corresponding to the individual single residue-pair interaction constraints. The random selection of pairs of interacting residues is described in Sect. 12.3, the obtained constraints are listed in Tab. C.3. The generation of start dimers from these constraints and the corresponding docking and clustering procedures are described in Sect. 12.4.

The results of the dockings w.r.t. the three native binding modes (Sect. 12.2) of each of the ten structures is listed in Tab. 13.7: for each of the binding modes the C_α RMSD of the best pose with a C_αRMSD $\leq$ 3.0Å and the corresponding number of poses fulfilling this criterion are presented.

First of all, we see that for three structures, at least one of the binding modes does not possess a corresponding pose with a C_α RMSD $\leq$ 3.0Å. In the case of 1X1O and

2UYU, one of the binding modes misses such a pose (binding mode A in both cases), in the case of 2BJK, two native binding modes (A and B) could not be found among the dockings. In addition, several binding modes have a low number of low-RMSD poses: in particular 1PVV C, 1X1O B and C, and 1YNB A have less than ten poses with a C_α RMSD ≤ 3.0Å to the respective binding mode. In addition, in three of these cases (all but 1X1O C) the lowest-RMSD pose yields a quite poor C_α RMSD $\in \left[2.0\text{Å}, 3.0\text{Å}\right]$.

In the other cases, we find a better coverage of the corresponding binding modes, with values between 30 and 346, and very good best-pose C_α RMSDs ≤ 1.0Å. Only for 1KW6 A, 1PVV A and B, 1QK1 A, 2F1D A and C, as well as 2UYU C, the best pose yielded a C_α RMSD $\in \left[1.0\text{Å}, 2.0\text{Å}\right]$ to the respective binding mode.

When additionally including false-positive constraints not corresponding to the native binding modes present in the complex, the above values are slightly different. This is due to the fact that each pose is assigned to the binding mode corresponding to the constraint for which the lowest penalty score was obtained. Hence, the subsequent iterative procedure of clustering and singleton removal (see Sect. 12.4) might yield slightly different results. The corresponding results are presented in Tab. C.4 (differences are underlined), demonstrating that the employed procedure can be considered robust against the inclusion of additional constraints.

13.2.2 *Assembly Performance*

To assess the performance of 3D-MOSAIC when only one pair of interacting residues per assumed complex binding mode is known (see Section 12.3), we performed 32 runs with different parameters to assess whether 3D-MOSAIC is able to find near-native complexes in such a scenario. The parameters used in these runs differ from those used in the benchmark scenario, in particular, greater values for transformation matching are used. In addition, based on the observation that many complexes exhibit symmetries, in 16 of these runs, we try to pre-rank the obtained sub-complexes by the number of symmetry mappings that have been found for the corresponding solution. A full list of the used parameters is given in Tab. C.25.

We now discuss the results for the assembly of our ten exemplary complexes with 3D-MOSAIC, before we investigate how this performance changes when additional false-positive binding modes are introduced. In this subsection, we will address the overall performance and will provide a more detailed look on the performance in dependence of the number of false-positive binding modes used in the next subsection.

In total, we could reconstruct a native assembly (tRMSD to the reference complex ≤ 2.5Å) for seven of ten complexes when using only docking poses corresponding to native binding modes. The complexes for which such an assembly could be obtained, are 1HI9, 1KW6, 1QK1, 1X1O, 1YNB, 2F1D and 3Q46. An overview of the overall performance is given in Tab. 13.8. A comparison with the docking results already presented in Tab. 13.7 shows that these complexes coincide well with those complexes for which good docking poses have been obtained for all native binding modes.

In particular, 2BJK and 2UYU could be considered a hard case, because here, two respectively one binding modes were not covered by a sufficiently good docking pose. The same is true for binding mode A of 1X1O, the other two are also poorly covered: only 8 and 6 good poses with respective best-pose RMSDs of 2.280Å and 0.615Å were

Complex	1HI9	1KW6	1PVV	1QK1	1X1O	1YNB	2BJK	2F1D	2UYU	3Q46
Monomers	10	8	12	8	6	6	6	24	8	6
Correct	24	12	0	31	4	10	0	1	0	32
Mean Rank	7.17	1.17	-	5.25	85	29.8	-	70	-	1.03

Table 13.8: Overview of the reconstruction performance of the SRPIC experiments using only the docking poses corresponding to native binding modes. "Correct" denotes the number of parameter sets for which a near-native complex was found, "Mean rank" the average rank of the first pose with a tRMSD $\leq$ 2.5Å over all successful parameter sets.

found for binding modes B and C respectively. Yet, 1X1O could be reconstructed in 4 of 32 models (12.5%).

1X1O is with six monomers a comparatively small dimer of tightly bound trimers, which increases the likelihood of being reconstructable. Here, not the transformation match score is responsible for the correct reconstruction but a tolerant clash checking (150 clashes per pair in all four models) and the final symmetry optimization. For example, for several solutions generated by 3D-MOSAIC using parameter set *srpic_06*, a full set of six symmetry mappings could be determined during symmetry optimization, yielding an overall good topology. In particular, the top-ranked solution after re-ranking by the number of symmetry mappings that could be performed during symmetry optimization, originates from a solution that was originally ranked at 41 and yielded a tRMSD of 3.41Å. After symmetry optimization, the tRMSD improved to 1.883Å.

In contrast, 1PVV had at least mediocre poses for each of the three binding modes, even though one (C) showed a poor coverage. Yet, it was not reconstructable. There are two main reasons for that. First, 1PVV's topology resembles that of a hollow sphere. Consequently, many different poses lead to a valid attachment of a new monomer, because the number of direct neighbors is comparatively low and thus, less clashes can occur. In comparison, the monomers of a tight complex like 1X1O must fit better into their neighborhood, even if no matching poses from other monomers can be found. The second reason is the size: 1PVV is built from twelve monomers, which vastly increases the number of potential solutions in comparison to 1X1O. Hence, if the transformation match score can not determine well-matching poses (which can be assumed from the docking results), a native solution (if possible at all) will be further down-ranked in any subsequent iteration and soon be dropped.

Given these observations it comes a bit as a surprise that the 24-mer 2F1D could be reconstructed. And indeed, there was only one parameter set (*srpic_16*) with which 3D-MOSAIC could reconstruct 2F1D sufficiently well. The corresponding tRMSD of the best pose is 2.21Å, the rank for this pose 94. The first pose with an tRMSD $\leq$ 2.5Å was found at rank 70 (tRMSD 2.49Å). Here, no symmetry optimization could be performed, but the overall complex match scores of 38.867 and 38.824 for the first and best pose with tRMSD $\leq$ 2.5Å can be considered reasonable (by experience, as a rule of thumb, we found that complexes with a complex match score $\geq$ 1.5 times the number of monomers can be considered to show at least the general properties of the original complex topology). While a near-native assembly was obtained for only one parameter set, at least one pose with a tRMSD $\leq$ 4.0Å was found for ten different parameter sets.

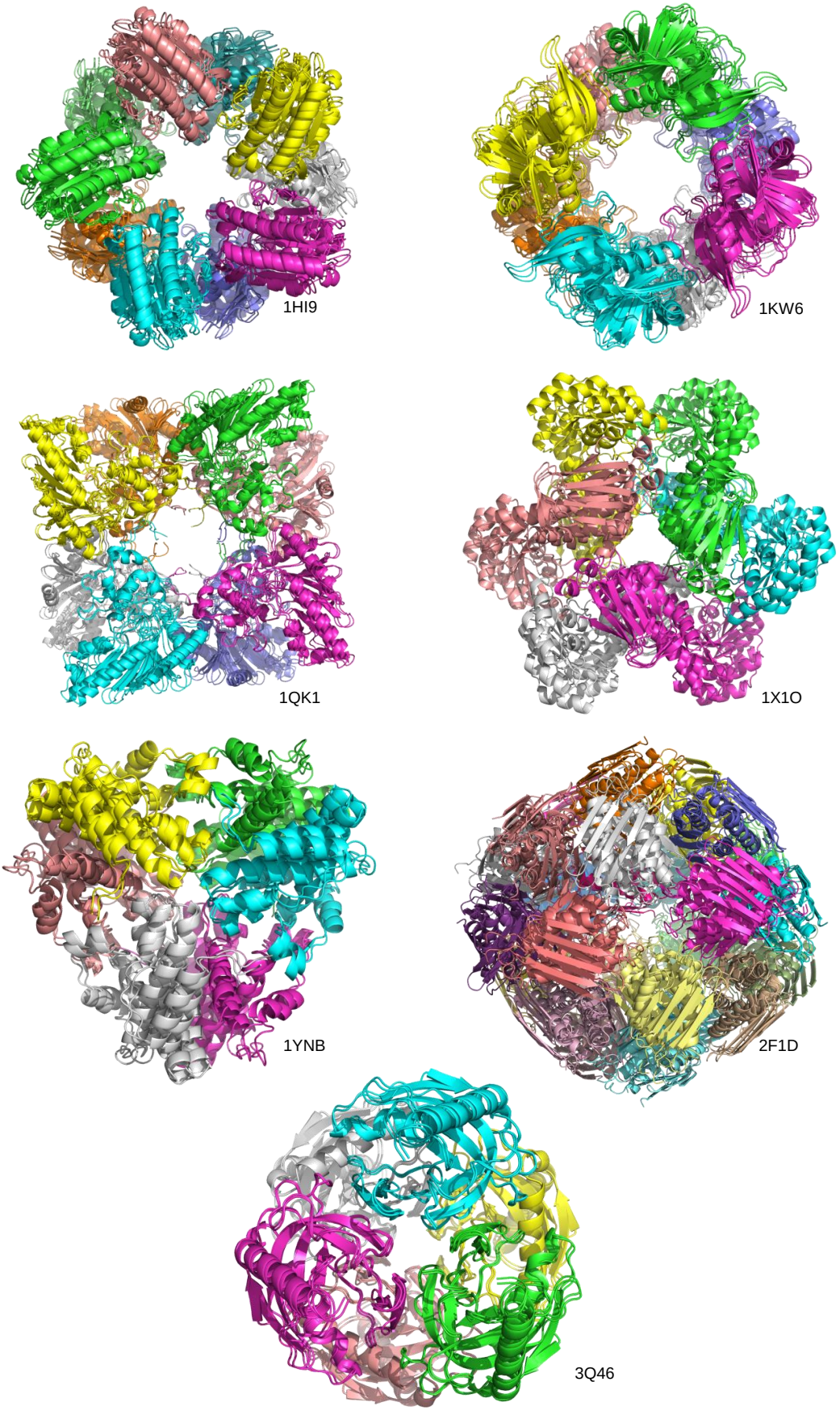


Figure 13.13: The seven complexes that could be reconstructed using only docking poses that correspond to native binding modes in the reference complex. Each assembled complex is superimposed onto the respective reference.

We do not consider such complexes to be near-native, however, they can potentially help to derive more detailed information on the individual near-native binding modes. This information could then be used to re-dock the respective binding modes at a more local scale and obtain a higher sampling density for the respective interfaces. This in turn could lead to an improved quality of the assembly. In addition, it has to be kept in mind that we intentionally performed a very wide-range sampling for each of the constraints (Sect. 12.4), neglecting the fact that additional expert knowledge might be available. Such knowledge could already be used to further limit the space that has to be sampled for each binding mode, and thus to obtain a higher sampling rate in the vicinity of the native binding modes (or constraints).

To summarize the above presented findings, we can say that the assembly succeeded in five out of ten cases (1HI9, 1KW6, 1QK1, 1YNB, and 3Q46) with a sufficient number of parameter sets ($\geq$ 10 ($\approx$ 30%)) and a reasonable average rank. Two additional complexes (1X1O, 2F1D) could be assembled in a small number of parameter sets and three complexes (1PVV, 2BJK, 2UYU) could not be assembled at all. Here, the quality of the docking poses and to some extent also the complex topology are the crucial factors for a successful assembly. An overview of the seven reconstructed complexes is given in Fig. 13.13.

13.2.3 *Performance When Introducing Non-Native Binding Modes*

In the previous subsection, we presented the results of assemblies using only the docking poses corresponding to the constraints derived from the native binding modes (called scenario i)). However, we can also investigate how the performance of 3D-MOSAIC changes when additional docking poses corresponding to false-positive constraints are included.

In addition to scenario i) which uses only docking poses from the native binding modes, the following assembly data sets were compiled for each of the complexes: ii) native and three false-positive binding modes, iii) native and six false-positive binding modes, and iv) native and all (ten) false-positive binding modes. In total, we thus yield 40 assembly data sets, four per complex.

False positives	1HI9	1KW6	1PVV	1QK1	1X1O	1YNB	2BJK	2F1D	2UYU	3Q46	Total
0	24	12	0	31	4	10	0	1	0	32	7
3	0	10	0	24	0	0	0	0	0	32	3
6	0	8	0	12	0	0	0	0	0	12	3
10	0	2	0	9	0	0	0	0	0	0	2

Table 13.9: Overview of the reconstruction performance of 3D-MOSAIC in the SRPIC experiments using docking poses corresponding to different numbers of false-positive binding modes in addition to those of the native binding modes. The last column denotes the number of complexes that could be reconstructed using the respective number of false-positive positive binding modes.

From the seven complexes that could be reconstructed using only the docking poses that correspond to the native binding modes, three could also be reconstructed when including up to six false-positive constraints. The three complexes are 1KW6, 1QK1, and 3Q46, three complexes for which good docking poses similar to each of the native binding modes could be obtained at a high coverage ($\geq$ 50 poses, cmp. Tab. 13.7), as

		Number of parameter sets		
Experiment	N	single	pair	triple
CV1	10	0.244	0.489	0.578
CV2	40	0.279	0.298	0.304

Table 13.10: Mean cross-validation coverages when a) considering only complexes assembled from docking poses corresponding to native binding modes, and b) including docking poses of up to ten additional false-positive binding modes. N corresponds to the number of assembly data sets, as described in Subsect. 13.2.2

compared to the remaining 4 complexes (1HI9, 1X1O, 1YNB, and 2F1D) which have (with the exception of 1HI9) a low coverage for at least one of their native binding modes or exhibit a more difficult topology (2F1D).

This indicates that if a sufficiently high coverage can be achieved such that highly compatible poses yielding a large match score are available, the noise introduced by additional decoy poses does not severely affect the assembly process. In fact, two of the complexes (1KW6 and 1QK1) could also be reconstructed when including all docking poses corresponding to all ten false-positive binding modes.

Summarizing over all scenarios, a near-native complex could be generated for 15 out of 40 assembly data sets (37.5%).

13.2.4 *Cross-Validation*

In the previous sections, we have seen that 3D-MOSAIC is, in principle, able to correctly reconstruct oligomeric assemblies in an information-scarce situation. We are now interested in how well the above parameter sets can be expected to perform in the general case, i.e., on unseen data for which only information on one interacting pair of residues per assumed complex binding mode is available. To this end, we performed two cross-validation experiments over the above presented parameters as follows: in the first cross-validation experiment (CV1), we consider only those ten assemblies from scenario i), i.e., we use for each complex only those docking poses which correspond to native binding modes. The second experiment (CV2) then comprises all forty complexes from scenarios i)-iv).

In both cross-validation experiments, we use a 5-fold cross-validation, because the data sets are comparatively small (ten and 40 complexes, respectively). In CV1, rather than providing an estimate, the exact average CV coverage can be easily computed because only $\binom{10}{8} = 45$ (data set size 10, training set size 8) different training folds can be generated. In CV2, we average over 1000 5-fold cross-validation runs.

As in the cross-validation experiments from Subsect. 13.1.3, we again perform the cross-validation in the following three setups: the single best-performing parameter set on the training data is used for prediction, and analogously, the combination of two (and three) parameter sets that yield a maximum number of correctly reconstructed benchmark complexes in the training set. The results are summarized in Tab. 13.10.

As can be seen, there is a clear difference between both experiments. The cross-validation coverage for experiment CV1 is better than that of CV2, at least when more than one parameter set is selected in the training stage. In CV1, on average, 5

respectively 6 out of ten complexes can be reconstructed when using the best pair or triple of parameter sets. Given the fact that only 7 complexes in total could be reconstructed, we can assume that three different parameter sets suffice to reconstruct the majority of complexes, at least if the complexes are similar to those used in our SRPIC experiments.

In experiment CV2, the obtained coverages are much lower, however, this results from the decreasing performance of 3D-MOSAIC when additional docking poses for false-positive binding modes are included. As described in Subsect. 13.2.3, a successful reconstruction was observed only for 15 of 40 assembly data sets (37.5%). Hence, the assembly data sets for which a successful reconstruction can be achieved are well-covered already with two models. Yet, these values have to be taken with care, due to the low performance of 3D-MOSAIC when docking poses corresponding to non-native binding modes are included in the assembly.

13.3 GLOBAL DOCKINGS AND COMPARISON TO COMBDOCK

CombDock is an algorithm that, similar to 3D-MOSAIC, uses docking poses that are generated independently of the actual assembly process. In this section, we will thus compare the assembly performance of both approaches on the global docking poses generated by CombDock's docking protocol.

First, we will investigate the overall performance of the docking protocol and the quality of the corresponding docking protocol. Subsequently, we will discuss and compare the results of the assembly stage employed in both methods.

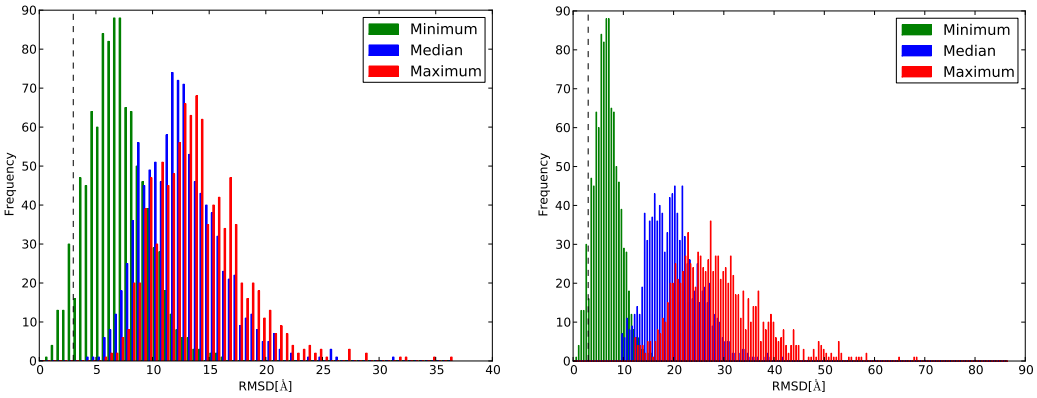
13.3.1 *Docking Results*

The CombDock algorithm entails, besides the actual assembly, also the generation of the docking poses between all pairs of monomers in the complex. The poses are obtained from a rigid, global docking protocol, i.e., no interaction or binding mode information is required (Subsect. 3.3.5). The employed algorithm uses a local feature mapping to generate poses which are then subsequently scored by shape complementarity, neglecting any biochemical properties. The obtained docking poses are not constraint to specific interfaces, but rather the mobile protein can be globally docked at any surface patch of the binding partner that yields a promising interaction score.

Unlike our previous experiments, in this case, a discrimination into distinct interfaces is not given, due to the global nature of the docking algorithm. A clustering of the docking poses could yield a rough discrimination into groups of similar poses; clusters with a high coverage could be interpreted as potential native binding modes. However, performing such a clustering would give us an unfair advantage over CombDock which uses the global dockings without further knowledge of potential binding modes.

In addition, CombDock performs an all-vs-all docking. For hetero-oligomers, we thus also obtain docking poses between pairs of protein types that do not natively interact in the resulting complex. We refrained from excluding such poses during the assembly with 3D-MOSAIC to keep the comparison to CombDock fair.

Nevertheless, we can assess how well the docking stage of the CombDock algorithm was able to find near-native poses for pairs of natively interacting protein types. To



(a) Dimer C_α RMSDs of the first 100 poses per pair of docked monomers

(b) Dimer C_α RMSDs of all poses per pair of docked monomers

Figure 13.14: Distributions of C_α RMSDs obtained from the CombDock-generated docking poses for all native binding modes.

this end, we calculated for each of the obtained poses the C_α RMSD to any of the dimers representing native binding modes as determined in Sect. 13.1.1.

By default, CombDock uses the first 100 poses per pair of monomers to assemble the complex. The corresponding distributions of minimum, median and maximum RMSD are given in Fig. 13.14a. The distributions over all values is given in Fig. 13.14b. In can be clearly seen that CombDock is not able to find a near-native pose for most of the binding modes.

In fact, only in 61 cases (6%), at least one pose with a C_α RMSD $\leq$ 3.0Å was obtained (the corresponding complexes and binding modes are given in Tab. C.5). Among them, only 2 complexes have such a pose for each of their native binding modes: 1XXC_1_foreign, a dimer of trimers and the cage-like 3M4B_3_same (12 monomers); each of them contains two native binding modes. In 18 of these cases (2%), a pose with a C_α RMSD $\leq$ 2.0Å and only in one case (0.1%) at least one below 1Å was found. These findings do not change drastically when investigating all poses instead of the first 100, implying two conclusions: if CombDock is able to detect a native binding mode, it can be found among the top 100, and because the scoring function only accounts for shape complementarity, the size of the interface must be comparatively large.

13.3.2 *Comparison of Performance of CombDock and 3D-MOSAIC using CombDock's Pairwise Global Docking Poses*

In the following, we address and compare the overall performance of CombDock and 3D-MOSAIC, both using the 100 best-scored poses (generated by CombDock) between all pairs of monomers in the complex (including those between monomers that do not interact natively).

Similar to 3D-MOSAIC, CombDock tries to solve a combinatorial problem. However, while 3D-MOSAIC does so in an iterative fashion, i.e., by attaching one monomer to all sub-complexes obtained from the previous iteration, CombDock first constructs a complete graph with an edge between two monomers for each corresponding pose. Subsequently, a clash-free minimum spanning tree is determined by iteratively join-

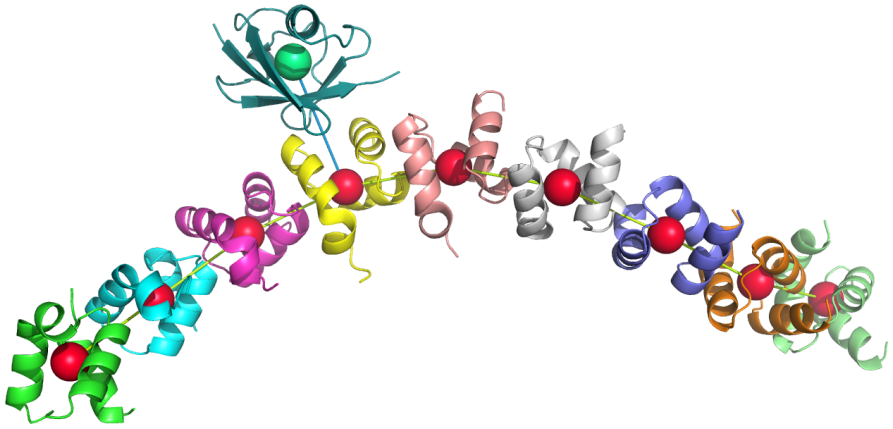


Figure 13.15: Structure and underlying topology graph of 2BWE. Spheres of the same color correspond to the same protein type, analogously equally colored edges between two spheres correspond to the same binding mode.

ing sub-trees for each edge between a pair of vertices in both sub-trees that does not lead to steric clashes. The best trees are then greedily selected for further processing.

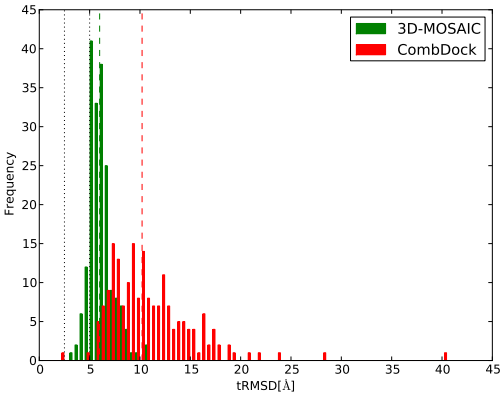
This procedure is computationally very intensive, leading to an immediate abortion of CombDock arising from a too large input space in 58 cases. The assemblies for further 60 complexes could not be completed within the running time threshold of the queue (5 days), leaving 190 complexes for the comparison (3D-MOSAIC generated complexes for all 308 assemblies).

Overall, CombDock and 3D-MOSAIC were able to reconstruct only one respectively none of the complexes when using the pairwise global docking poses generated by CombDock. The one complex that CombDock was able to assemble (tRMSD 2.12Å, rank 1966) is 2BWE_1_same, a linear, slightly turreted complex of nine small monomers of the same protein type and an additional single monomer of another type attached to one of the other nine (see Fig. 13.15). The overall topology is rather unusual, yet explicitly discussed in the accompanying article [372]. The binding mode between two helical monomers (yellow) is one of the 61 cases for which a pose with a good RMSD (1.99Å) was found.

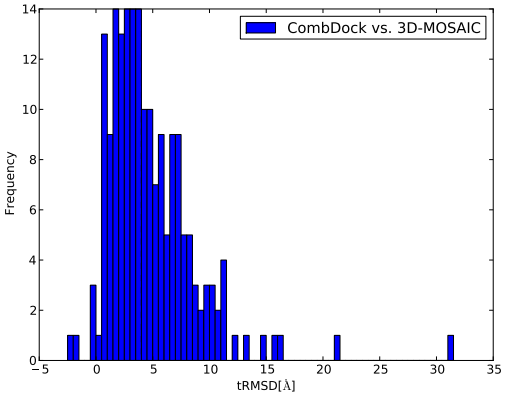
In the general case, none of both algorithms performed well. Besides the aforementioned 2BWE, none of the complexes is even close to being correct, as indicated by the corresponding topology RMSDs. However, even though 3D-MOSAIC did not yield

	tRMSD [Å] (rank)		
Assembly	CombDock	3D-MOSAIC	Δ_{tRMSD}
1E7P_1_same	7.34 (42)	9.65 (27)	-2.31
1MQM_2_same	8.44 (15)	8.60 (14)	-0.16
2BWE_1_same	2.12 (1966)	3.95 (38)	-1.83
2HEY_1_foreign	6.23 (27)	6.51 (13)	-0.28
2HEY_1_same	6.39 (1)	6.83 (46)	-0.44

Table 13.11: Cases where CombDock obtained a minimum tRMSD complex better than 3D-MOSAIC.



(a) Distribution of best tRMSDs for 3D-MOSAIC and CombDock



(b) Difference of best tRMSDs between CombDock and 3D-MOSAIC

Figure 13.16: Distributions of minimum tRMSDs obtained over all complexes generated by CombDock and 3D-MOSAIC. Dotted lines correspond to tRMSD thresholds of 2.5Å and 5.0Å.

any solution with tRMSD $\leq$ 2.5Å for any of the benchmark complexes, it was in general able to generate solutions with significantly better tRMSDs for almost all of the complexes. Only five cases could be determined where the tRMSD of the best CombDock complex is better than that generated by 3D-MOSAIC, shown in Tab. 13.11.

Fig. 13.16 shows the overall distributions of the minimum topolgy RMSD per complex, for both 3D-MOSAIC and CombDock. While the overall performance is poor, it is still clearly observable that 3D-MOSAIC produces assemblies of significantly better quality (ranksum statistics p-value $1.31e-46$): the means (and standard deviations σ) for the tRMSD of 3D-MOSAIC and CombDock are 6.13Å (1.17Å) and 10.95Å (4.34Å), respectively. The mean (median) difference between the corresponding tRMSDs of 3D-MOSAIC and CombDock is 4.82Å (3.88Å).

As can be seen from Fig. 13.16a, 3D-MOSAIC generates assemblies with a topology RMSD below 5.0Å in 21 cases, compared to two for CombDock. While such complexes lack the structural accuracy to be considered near-native, they can potentially still provide information on approximate binding modes, which are addressed in Sects. 13.1 and 13.2.

In such a case, an iterative refinement employing several subsequent turns of dockings and assemblies could possibly be performed. However, besides the questionable chance of success, such an approach is only applicable in a single-case study as it requires intensive investigation of the individual generated complexes. In addition, secondary software to assess the structural quality must possibly be applied, hence this approach is not applicable in a benchmark scenario as presented here.

Yet, even in such a coarse-grained scenario as framed by the global dockings obtained from CombDock, the results indicate that 3D-MOSAIC can determine complexes of better structural quality, even though the structural quality is still insufficient in most of the cases.

13.3.3 *All vs. Natively Interacting Protein Types*

In the previous subsection, we performed a comparison between 3D-MOSAIC and CombDock at the conditions of CombDock: only 100 poses per pair for each pair

Comparison	a	b	c	d	e
Mean [Å]	-0.60	-0.54	-0.54	-0.60	-1.20
σ [Å]	1.18	1.16	1.07	1.10	1.81

Table 13.12: Mean and standard deviations for the tRMSD distributions comparing the five different scenarios shown in Fig. 13.17.

of protein types present in the complex. We thus have neglected two facts: i) some of the interfaces in complexes are comparatively small, and can thus not be found among the first 100 poses, and ii) even though no knowledge about the approximate interfaces is available, one might at least be able to exclude pairs of protein types that are certain to not interact in the complex.

While this information can not be incorporated into CombDock, we can nevertheless investigate and compare the performance of 3D-MOSAIC in the following four scenarios: 100 poses at all interfaces, 100 poses for each pair of natively interacting protein types, and the same scenarios again with all CombDock-generated poses instead of only the first 100.

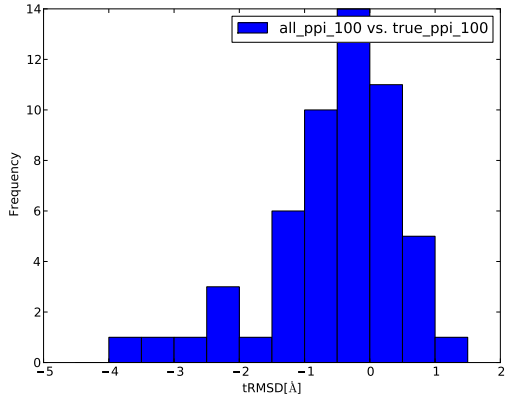
A comparison of these scenarios is presented in Fig. 13.17, the main characteristics of the respective comparisons are summarized in Tab. 13.12. 56 complexes could be determined for which CombDock produced dockings between pairs of protein types that do not interact in the native complex. As expected, only considering the docking poses corresponding to natively interacting pairs of protein types, yields an increase in assembly performance: when considering only the top 100 as well as all poses, we obtain (Fig. 13.17a and 13.17b) distributions favoring the scenario where only native pairs of protein types are considered. The corresponding mean tRMSDs (σ) are -0.60Å (1.18Å) and -0.54Å (1.16Å).

In contrast, it is not so intuitive to answer what happens, when more poses are included. To this end, we can consider the scenarions 100 vs. all poses for both cases, where all and only the true interactions are considered. In the former case, all 308 complexes are considered, in the latter case only those 56 where dockings between non-interacting pairs of protein types have been performed. These comparisons are shown in Fig. 13.17c and 13.17d, respectively. Interestingly, we also observe slight improvements in these cases, when considering all poses: the distributions show means (σ) of -0.54Å (1.07Å) and -0.60Å (1.10Å).

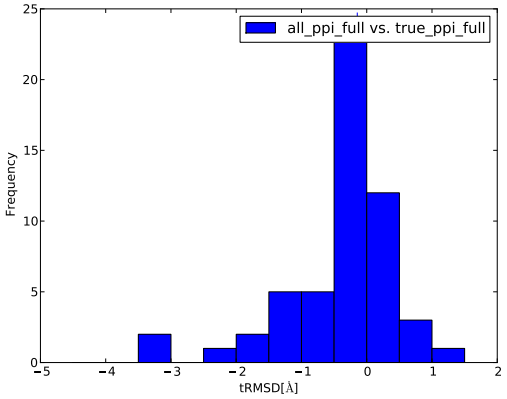
The reason here is, that when only considering the best 100 poses, poses corresponding to smaller native binding modes might be missed. Hence, even though the use of all poses also entails the inclusion of a large number of decoy poses, poses similar to smaller binding modes can help the transformation match score to detect and better score those complex candidates that better correspond to native complexes.

This assumption is especially supported by the distribution shown in Fig. 13.17e. Here, we compare the 100 best poses of all pairs of protein types vs. all poses of the natively interacting protein type pairs: while poses corresponding to non-native pairs of protein types are removed, the full set of poses generated by CombDock is considered for the remaining pairs. Here, a mean tRMSD difference of -1.20Å and a standard deviation of 1.81Å are observed.

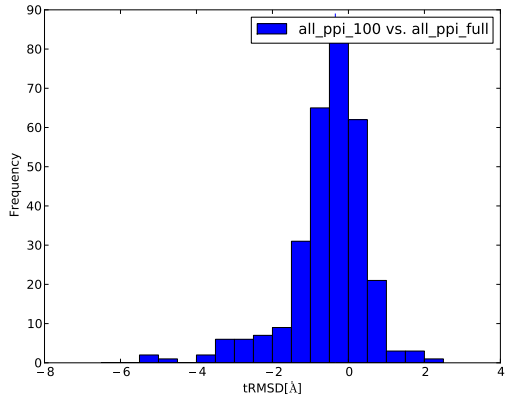
From these observations, we can thus conclude that not only those docking poses involving the largest interfaces are of relevance for the complex topology, and that



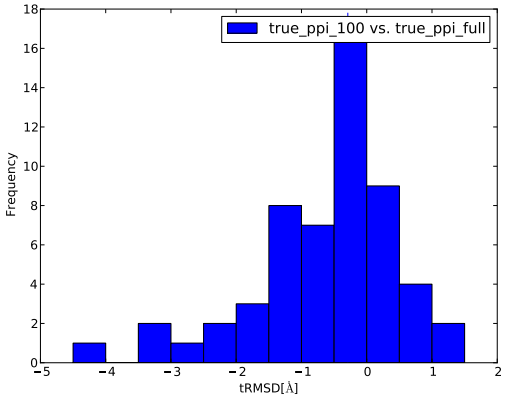
(a) 100 poses for all pairs of protein types and truly interacting pairs of protein types



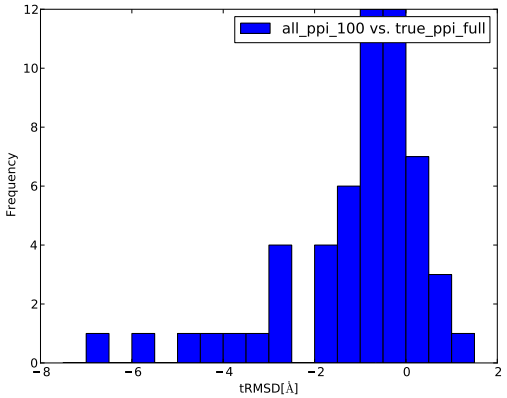
(b) All poses for all pairs of protein types and pairs of truly interacting protein types



(c) 100 vs. all poses for all pairs of protein types



(d) 100 vs. all poses for all pairs of truly interacting protein types



(e) 100 poses for all pairs of protein types vs. all poses for all pairs of truly interacting protein types

Figure 13.17: Distribution of difference between corresponding best tRMSDs for 3D-MOSAIC in five different pairs of scenarios. In each case, the second scenario produces smaller (better) tRMSDs, yielding a shift of the distribution towards the negative.

chance to detect corresponding poses and thus the quality of the complexes assembled from the obtained dockings.

13.3.4 *Comparison of CombDock and 3D-MOSAIC in Their Own Workflows*

In the previous sections, we have compared the performance of 3D-MOSAIC and CombDock when using the pairwise global docking poses generated by CombDock. In this section, we want to compare the performances of both methods at their own respective conditions and assumptions. In the benchmark scenario used to assess the performance of 3D-MOSAIC (Sect. 13.1.2), we assume that the interaction geometries of the binding modes which are established when forming the native complex are roughly known. According to these assumptions, each assumed binding mode is locally sampled, yielding 10,000 docking poses per binding mode which are then used for assembly. In contrast, CombDock does not incorporate such information and in addition only uses 100 (global) docking poses per pair of interacting monomers. Furthermore, it also generates docking poses for pairs of protein types that do not natively interact in the complex.

When comparing both methods in their own workflows, 3D-MOSAIC clearly outperforms CombDock: 3D-MOSAIC yielded solutions for all 308 complexes, while an immediate abortion of the CombDock was observed in 58 cases. Further 60 cases could not be completed within the allowed running time of the queue. 3D-MOSAIC was able to overcome this limitation, if required, by using the implemented restart feature.

Even when using one single parameter set, 223 and 221 benchmark complexes could be reconstructed with 3D-MOSAIC using S^{da} and S^{rmsd}, respectively. These numbers increase to 258 and 252, respectively, when using the best-covering triple of parameter sets. In contrast, CombDock was able to correctly reconstruct one benchmark complex (2BWE_1_same, tRMSD 2.12Å, Sect. 13.3.2), however with a very low rank (1966). In comparison, in 94.9% (97.1%) of the benchmark runs where 3D-MOSAIC yielded a correctly reconstructed complex, the first correct solution was found among the top 10 (25). Furthermore, the worst rank at which a correct solution can be found when using 3D-MOSAIC is rank 125: only 125 solutions are generated for hexamers (due to the solution reduction scheme, see Sect. 12.6.1), 100 for complexes with up to 20 monomers, and 50 (25) for complexes comprising at most 40 (60) monomers.

Even with as little information as one single interacting residue pair per assumed binding mode and under the aggravating conditions of additional false binding modes (Sect. 13.2.2), 3D-MOSAIC performs better than CombDock: none of the ten complexes tested in the SRPIC experiments could be reconstructed using CombDock; 3D-MOSAIC yielded correct reconstructions for seven of them when using the docking poses corresponding to native binding modes, 3 and 2 when additionally using docking poses corresponding to up to 6 and 10 false binding modes, respectively.

We thus can conclude that, because CombDock is not able to incorporate additional information on potential binding modes in its workflow, 3D-MOSAC represents the better alternative if such information is available.

14 DISCUSSION

14.1 SUMMARY

In this work, we have presented a novel combinatorial greedy algorithm, called 3D-MOSAIC, to assemble large oligomeric protein complexes from pairwise docking data. We introduced two new scoring functions of biological relevance to measure the similarity between two rigid (docking) transformations of the same protein: S^{da} which is based on a heuristic displacement/angle distance threshold and S^{rmsd} which employs an exact constant-time calculation of transformation RMSDs from protein covariance matrices. We derived a large and comprehensive benchmark set of protein complexes which are diverse w.r.t. to their size, composition and toplogy. In addition, we devised a new measure, called topology RMSD, for the comparison of oligomeric macromolecular assemblies generated from pairwise dockings. The robustness of this measure against conformational differences between monomers in the assembled and the reference complex makes it especially suitable when monomers with conformational differences are used during assembly. And finally, we validated our algorithm on a broad range of different scenarios.

Due to the high diversity of our benchmark set, we tested our algorithm with 96 different parametrizations, 48 for each of the two transformation matching scores. Using these parameter sets, we have shown that we can assemble 278 of the complexes in our benchmark set, a large fraction (258) thereof using no more than three different parameter sets. We performed a cross-validation to assess the predictive power of these parameter sets for unseen complexes and could show that they are indeed well suitable to be used for the assembly of unknown complexes. In addition, we evaluated these parameter sets on a second data set consisting of homo-hexameric structures, yielding successful assemblies for 15 of 17 complexes, underlining the general capability of our algorithm to assemble a large variety of different complexes.

Running time investigations showed that even complexes with 60 monomers could be successfully assembled in approximately one to two days. Compared to the effort required for a structural determination (if possible at all), we have thus developed an algorithm that can greatly contribute to and facilitate the efforts to structurally determine large oligomeric complexes. Running time limitations arise when clustering large complexes, but these can be alleviated by interactively reducing or disabling the clustering threshold after several levels using the provided restart feature. The required memory was typically in the range of at most 2GB, but can exceed that limit when assembling heteromeric complexes with a large number of different interfaces.

A baseline study of runs without intra- and post-clustering comparing enabled and disabled *transformation match scoring* clearly demonstrated the importance and efficiency of our scoring function for complex assembly from pairwise dockings: the number of correctly assembled complexes was increased by a factor of 2.96 for an enabled score when using the respective three best-performing parameter sets. We could thus demonstrate its capability to filter the near-native poses from large, diverse dimer decoy sets produced by RosettaDock.

An evaluation of 3D-MOSAIC on pairwise dockings using a non-local sampling for docking start dimers generated in a straightforward fashion from single-pair residue interaction constraints showed that the algorithm can find near-native solutions in 7 of 10 cases when considering only the docking poses corresponding to the native binding modes. In two of the failed cases, the docking algorithm could not provide docking poses that are sufficiently close to the native binding mode. When adding an equal rate of noise with false positive constraints (3 artificial binding modes), 3 complexes could still be reconstructed, 2 of which could still be successfully assembled with as many as 10 artificial binding modes.

A comparison to CombDock using the 100 best global dockings generated by CombDock's integrated docking algorithm demonstrated that the use of global docking poses is unlikely to yield any reliable results. In particular, CombDock could assemble only one and 3D-MOSAIC none of the benchmark complexes. Overall, we could show that incorporating additional docking poses corresponding to smaller patches of complementary surfaces can improve the assembly results to some extent, even though the results were still not sufficiently good to yield a near-native solution.

In a number of cases, 3D-Mosaic did not perform well. These were complexes with a low connectivity, i.e., such with a low number of interfaces for all monomers. This is especially prevalent for cage-like or mono-layered ring-like structures with many subunits: here, the algorithm must rely on a good ranking based on the docking scores, until in the final iteration a ring closure is possible and the *transformation match score* can be applied. Furthermore, helical monomers as well as heavily intertwined proteins can hamper the assembly process.

Summarizing, 3D-MOSAIC extends the potential of docking-based reconstruction of large complexes by increasing the number of subunits as well as the number of protein types that are involved. The algorithm can already find near-native complexes with a few number of parameter sets, yet for difficult cases, it provides many additional features, which can be adjusted to suit specific use cases. It is applicable to asymmetric and symmetric complexes with the benefit of an additional symmetry optimization of the latter ones.

14.2 CONCLUSION

In this project, based on the transformation match score and 3D-MOSAIC, we could thus show that oligomeric protein complexes can be computationally assembled using pairwise dockings sampling the assumed native complex binding modes.

With 3D-MOSAIC, we aimed at overcoming the limitations of current algorithms in terms of complex size, number of different protein types, symmetry assumptions, and topological properties. On our diverse benchmark set of 308 complexes, we could successfully demonstrate that this goal was indeed achieved.

We also wanted to reduce the amount of information required for a successful assembly: 3D-MOSAIC only requires a representative high-resolution structure for each protein type, the respective multiplicity in the complex, and docking poses sampling each of the assumed native complex binding modes. The benchmark and SRPIC experiments demonstrated that this information is indeed sufficient to obtain correct reconstructions, however, using the global dockings provided by CombDock, we could also show that missing knowledge on potential complex binding modes will likely not lead to a successful reconstruction.

While 3D-MOSAIC fails when no assumptions about complex binding modes can be made, such information can often be obtained from various experiments, e.g., cross-linking and correlated-mutation studies. Our method is thus applicable whenever such information is available and can thus represent an alternative or complementary approach for integrative methods: here, many different data sources providing information on distances between the complex components are combined to obtain structural models of oligomeric protein complexes at medium-to-high resolution. In this context, the incorporation of additional data sources such as cryo-EM data which can help to guide the assembly process, and the generation of sub-complexes to overcome 3D-MOSAIC's limitations w.r.t. assembling weakly connected complexes, are currently explored.

Part IV

FUTURE WORK

15 RETROSPECTIVE AND OUTLOOK

As stated in introductory chapter of this thesis (Section 1.2), the aim of this work was to deepen the knowledge related to protein structure and protein interaction prediction. We have addressed two different questions: the first one whether elastic network model normal modes can be used to predict the conformational changes upon binding of small molecules, and the second whether macromolecular oligomeric assemblies can be constructed from pairwise dockings.

We could show that elastic network model based normal modes on coarse-grained protein structures are in general not suitable to predict the conformational changes associated with ligand binding. The main reasons here can be considered the locality of the movements to be performed as well as the presence of the ligand itself. The interactions with the ligand can lead to additional energetic contributions that influence the energy required for certain parts of the backbone to move and consequently also the normal modes. In addition, the binding itself represents a continuous process, where not all parts of the protein are equally involved during all stages of the ligand uptake. This again can affect the conformational behavior of the protein.

A first step towards a better understanding of this processes is, for example, the use of tools to investigate the residue-residue interaction networks of a particular protein in different protein conformations, e.g. by tools such as RINalyzer [373]. Such knowledge in turn could then be used to derive residue-residue interaction-specific distance constraints that could for example be used to establish more sophisticated spring force functions used during normal mode analysis. In contrast to current approaches where the forces only depend on residue-residue distances, such constraints can for example weigh the contributions of individual residue-residue interactions based on how conserved they are in a set of alternative protein conformations. However, in general, we consider the potential gain from using elastic network model normal modes during protein-small molecule docking to be low in the general case, and have thus abandoned this field of research.

The assembly of macromolecular oligomeric complexes from pairwise docking poses has proven to be more successful. We could assemble a large majority of the complexes in our diverse benchmark set, and have developed a simple scoring function that is able to recognize near-native docking poses by scoring the mutual information from interfaces in the immediate neighborhood of a monomer to be attached.

We could demonstrate that 3D-MOSAIC, the algorithm we developed to this end, in its current form is already capable to assemble complexes of up to 60 monomers within a few days. Compared to the time required for crystallization of the respective complexes (if possible at all), this can be considered a great success. Even more important, modeling such assemblies before the actual crystallization process can help to identify (or limit) the conditions at which a successful crystallization is likely.

Likewise, 3D-MOSAIC is also suitable to assist in integrative approaches by providing an automated assembly tool. Distance constraints for example can be used to guide the docking process. The resulting poses are then used as the input to our algorithm.

However, several developments could help to improve the performance of 3D-MOSAIC even further: for example, the incorporation of the recognition and use of patches of monomers during the assembly process. For example, in the case of the pyruvate dehydrogenase E2 core, we have seen that trimers of E2 are formed which weakly interact over small bridges with other trimers (see Subsection 2.5.1). Here, detecting these trimeric patches could help to reduce the combinatorial space and thus to increase the likelihood that such complexes can be assembled. Furthermore, these patches can also be expected to lead to a significant increase in running time: we have seen that in later stages of the algorithm, the running time drastically increases with the number of components present in the complex. The identification and use of such patches instead of the individual monomers would thus require, for example in the case of a 60-mer with trimers as described above, only 20 iterations instead of 60.

Further improvements in running time can certainly be achieved through more efficient implementations, especially during complex similarity matching. For example, the RMSD computation using rigid transformations [351] that we established during the later stages of the development of 3D-MOSAIC, would render the use of the individual protein representations used during matching (see Section B.8) obsolete and would thus lead to a considerable speed-up. In addition, the exact computation in combination with a geometric hashing approach, can reduce the running time complexity by several orders of magnitudes.

Moreover, features such as the scoring of sub-complexes against low-resolution electron density maps or simple constraints such as the expected diameter of the complex can be expected to improve the performance of 3D-MOSAIC in difficult cases.

The scientific work with 3D-MOSAIC will also be continued. The following three problems are currently tackled: the assembly of the glycolysis complex, a hetero-oligomeric complex of several proteins on the glycolysis pathway, for which neither the exact stoichiometry nor the proteins that comprise the complex are exactly known. A large variety of data, for example on the potentially participating proteins and their interactions, has been collected by Sebastian Mock (Johannes-Gutenberg University, Mainz) under guidance of professors Elmar Jaenicke and Heinz Decker (Institute for Molecular Biophysics, Johannes-Gutenberg University, Mainz) and will be used as constraints to generate appropriate docking poses and guide the assembly process.

Furthermore, viral capsids are a field of interest. Often high-resolution monomeric structures are known, and in some cases also low-resolution capsid structures, for example the Dengue or Hepatitis B Virus. Here the aim is to determine high-resolution assemblies of the full viral capsids and to provide detailed information on their topology. Here, a successful application of 3D-MOSAIC can be considered to be an integral contribution to the research regarding anti-viral drug treatment.

The third project, which is currently in a very early stage, addresses the generation of protein complexes from homologous monomeric proteins and the fitting of the obtained complexes into electron density maps. Due to the use of homologous proteins, a successful assembly in that case may also entail the application of homology modeling approaches such as MODELLER [201, 202].

Finally, providing a web service for 3D-MOSAIC, possibly in collaboration with third-party docking servers such as RosettaDock, is desirable and would serve the applicability of 3D-MOSAIC and its accessibility by the scientific community.

Part V

APPENDIX

A

ADDITIONAL RESULTS FOR EXPERIMENTS WITH ENM NORMAL MODES

A.1 DATA SET COMPOSITION

Astex Diverse Set [294] pdb code	Protein name	#holos	#apo/holo pairs
1g9v	Deoxy hemoglobin	4	62
1gm8	Penicillin G acylase	1	1
1gpk	Acetylcholinesterase	4	28
1ia1	Dihydrofolate reductase	5	5
1l2s	Beta-lactamase	2	2
1l7f	Neuraminidase	12	12
1n1m	Dipeptidyl Peptidase IV	3	3
1n2v	TGT	9	9
1oq5	Carbonic anhydrase	32	137
1oyt	Thrombin	2	2
1p2y	Cytochrome P450cam	21	24
1r55	ADAM33	1	1
1s3v	Dihydrofolate reductase	7	7
1sg0	Quinone reductase 2	4	4
1t40	Aldose reductase	20	40
1uml	Adenosine deaminase	10	10
1v0p	Protein kinase 5	1	1
1w1p	Chitinase B	4	4
1ywr	p38 kinase	20	39
2br1	Chk1	4	4
2bsm	Heat shock protein 90	19	38

Table A.1: Overview of the proteins, corresponding holo structures, and apo/holo pairs used in the data set.

A.2 VALIDATION OF THE RECONSTRUCTION PROCEDURE

To ensure that the docking performance on the different normal mode subsets is not an artifact of the reconstruction procedure, we also compared the results of our dockings into the reconstructed holo conformations to those into the crystal holo structures. The results are given in Figure A.1.

The three histograms demonstrate that our reconstruction procedure indeed produces valid conformations. When only the standard protocol is taken into account, the distribution has a slight tail towards greater RMSD values, however, this tail is re-

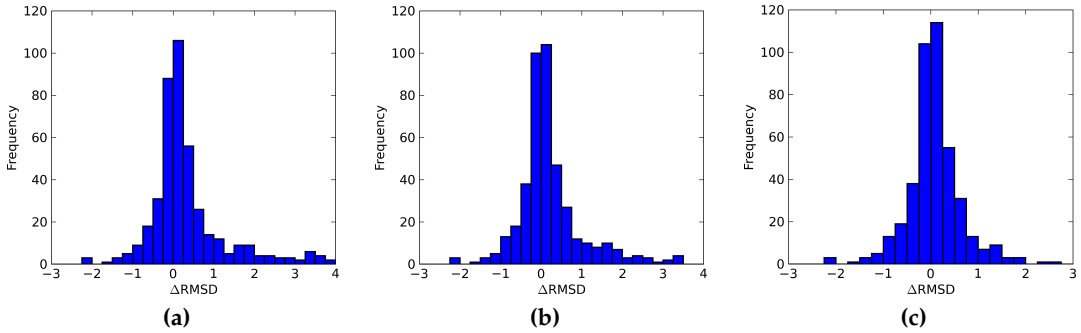


Figure A.1: Histograms of the difference in RMSD [Å] between the best poses obtained from docking (across all docking protocols) for each apo/holo pair into the 100% reconstructed holo conformation and the respective crystal holo structure (ΔRMSD = RMSD_reconstruction - RMSD_crystal). a) Standard protocols only, b) with soft protocols, c) with soft protocols and explicitly accounting for side-chain flexibility.

duced when additionally taking the soft dockings into account and it totally vanishes when, in addition, explicitly accounting for side-chain flexibility. The corresponding mean (and standard deviation) values are 0.33Å (0.93Å), 0.21Å (0.79Å), and 0.08Å (0.56Å), respectively. In the last case, 355 apo/holo pairs have a ΔRMSD < 0.5Å and only 24 have a ΔRMSD > 1.0Å. None of these 24 apo/holo pairs is part of the subset of 59 apo/holo pairs considered in Section 6.5. Thus, it can be expected that these results can be further improved.

Figure A.2 demonstrates in detail how these results translate to Figure 6.5 of the main article: again the baseline is the best pose RMSD obtained from docking into the crystal holo structure but, in contrast to Figure A.1, here we consider only the best pose produced by the respective docking protocol (both standard and soft protocols take the respective standard docking into the crystal structure as basis, because this is the native conformation and soft docking is not needed).

The difference in RMSD between the best pose obtained from all three minimization protocols for the 100% reconstructed holo and the best pose from docking into the crystal holo for the standard protocols of AutoDock, FlexX and GOLD is 0.23Å, 0.24Å and −0.14Å respectively (*ME* lines in the plot). Given the fact that a docking pose is considered correct when its RMSD from the native crystal pose is below 2.0, the obtained values are very small, with GOLD showing an even better performance on the 100% reconstructed holo conformations. The corresponding values for the soft protocols are 0.5Å, 0.09Å, 1.59Å. The values here are more diverse, with the GOLD soft protocol being an outlier. Yet, also in this case, we obtain docking results on the 100% reconstructed holo structure that are comparable with those of the bound docking using the same protocol.

However, some proteins may undergo considerable side-chain movements upon transition from apo to holo state; these are cases our reconstruction procedure cannot account for. Such cases can be dealt with by introducing additional side-chain rotamers, as done in section Docking with side-chain flexibility in the main article.

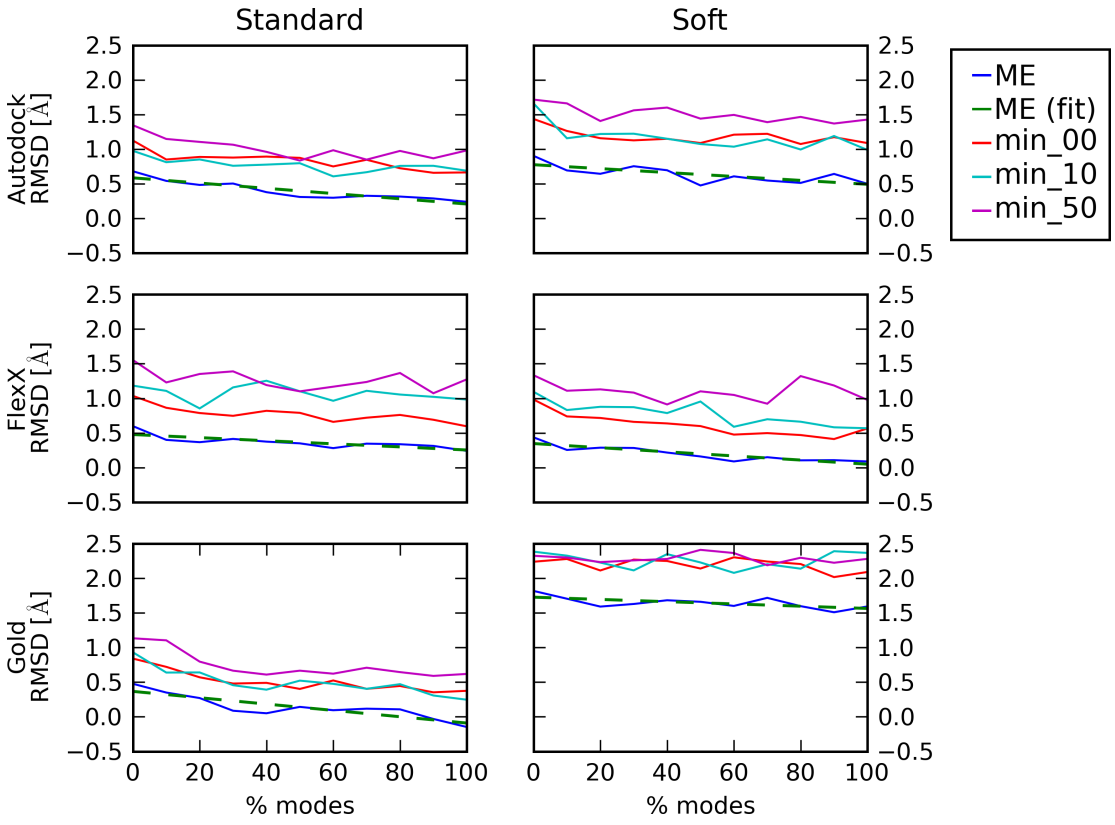


Figure A.2: Baseline best pose RMSDs for the reconstructions after subtracting the best pose RMSD from the docking into the corresponding holo crystal structure.

B

ALGORITHMIC DETAILS OF 3D-MOSAIC

B.1 INTERFACE LOCKING

As already stated in the introductory chapter (Ch. 8), we assume that the locations of the interfaces and the corresponding binding modes are roughly known and that these binding modes are distinct. Each binding mode corresponds to two complementary interfaces (Sect. 9.2), and once a docking pose from such an interface has been used, we can assume that this interface of the receptor is occupied, i.e., locked.

Moreover, the ligand monomer which is generated from the docking pose w.r.t. to the placement of the receptor has also used up one interface: the complementary or reverse interface to that of the receptor, since this is the interface via which the ligand interacts with the receptor.

Finally, as we will see in Sect. B.5, the docking poses from interfaces at other monomers in a complex can provide additional information on the reliability of the ligand placement. If this placement is supported by a binding mode presented by another monomer in the vicinity of the ligand, i.e., if there exists a docking pose for that monomer that would yield a ligand placement that is similar to the current one, we can assume that the ligand and that particular monomer interact as well. The interface of that monomer this docking pose belongs to as well as the reverse one of the ligand can thus also be locked.

The locking of all interfaces at which an interaction occurs hence prevents the algorithms from attaching multiple monomers over the same distinct binding mode. It thus avoids the generation of false positive or implausible solutions and reduces the combinatorial space.

However, in some cases, we cannot assume that the binding modes are distinct, especially in the case of global dockings. For such cases, the interface locking can be turned off.

B.2 SYMMETRIC BINDING MODE DETECTION

In many assemblies, we encounter symmetric binding modes, i.e., where each of the monomers in a homo-dimer has the same relative orientation to the other. Given a set of docking poses corresponding to a particular interface, determining whether this interface is symmetric can help to reduce the combinatorial space: first, if a symmetric interface has been detected, all poses in that interface that do not result in a symmetric binding mode can be discarded. Second, two interfaces are associated with one binding mode (cmp. Sect. 9.2). However, in a symmetric binding mode, these are not distinct w.r.t. to their location on the underlying protein, in fact, they are identical. Hence, when locking one of these interfaces, we must ensure that the other one is locked as well to prevent the algorithm from twice attaching a monomer at the same interface.

To determine the properties of symmetric docking poses, we take a closer look at Fig. 9.1), showing two exemplary transformations: $\mathbf{T}_1$ brings the monomer $M_{1,O}$ from the origin into its ligand position $M_{1,D}$. Likewise, $\mathbf{T}_2$ moves monomer $M_{2,O}$ from the origin to position $M_{2,D}$. It is obvious that the inverse transformations $\mathbf{T}_1^{-1}$ and $\mathbf{T}_2^{-1}$ bring the corresponding ligand monomers back to their original placement at the origin.

Intuitively, when applying a rigid transformation $\mathbf{T}_1^{-1}$ to the dimer ($M_{1,D}$,$M_{2,O}$) (the left dimer in the figure), we obtain the dimer orientation ($M_{1,O}$,$M_{2,D}$) (the right dimer), because rigid transformations do not alter any distances of the object and $\mathbf{T}_1$, $\mathbf{T}_2$ are obtained from the dimer itself. Analogously, $\mathbf{T}_2^{-1}$ yields ($M_{1,O}$,$M_{2,D}$) when applied to ($M_{1,D}$,$M_{2,O}$).

We can thus conclude that:

$$\mathbf{T}_1^{-1} = \mathbf{T}_2 \text{ and } \mathbf{T}_2^{-1} = \mathbf{T}_1 \tag{B.1}$$

In a homo-dimer corresponding to a symmetric binding mode, we also have:

$$\mathbf{T}_1 = \mathbf{T}_2 \tag{B.2}$$

because, as already stated, both monomers have the same relative orientation to each other and $\mathbf{T}_1$ transforms monomer $M_{1,O}$ w.r.t. to $M_{2,O}$ (and $\mathbf{T}_2$ $M_{2,O}$ w.r.t. to $M_{1,O}$). Since for a homo-dimer, $M_{1,O} = M_{2,O}$ (i.e., the protein type is the same and we use only one centered instance of each protein type), so is $M_{1,D} = M_{2,D}$ and hence $\mathbf{T}_1 = \mathbf{T}_2$.

From Eq. B.1 and Eq. B.2, we can transitively conclude that in symmetric dimers:

$$\mathbf{T}_1 = \mathbf{T}_1^{-1} \text{ and } \mathbf{T}_2 = \mathbf{T}_2^{-1} \tag{B.3}$$

When working with protein structures or docking poses, a binding mode cannot be expected to be exactly but only approximately symmetric. Hence, we need to allow for a certain amount of deviation from the ideal geometry: given an RMSD-threshold d_{symm}, in analogy to Eq. B.3, we say that a docking pose with a transformation $\mathbf{T}$ is symmetric, if (cmp. Eq. 10.6):

$$RMSD(\mathbf{T}, \mathbf{T}^{-1}) < d_{symm} \tag{B.4}$$

However, depending on the outcome of the docking algorithm, we might find symmetric poses even for non-symmetric binding modes. Conversely, not all dockings obtained for a symmetric binding mode are necessarily symmetric. Hence, we consider a binding mode to be symmetric only if a certain fraction f_{symm} of all corresponding poses is found to be symmetric.

For two corresponding interfaces I_1 and I_{-1} in a binding mode, one containing the docking poses of M_1 w.r.t. M_2 and the other those of M_2 w.r.t. M_1, the binding mode is considered to be symmetric if in both interfaces, the fraction of symmetric poses is above f_{symm}.

If a binding mode is found to be symmetric, all asymmetric poses are disabled, because they are not expected to be relevant for complex assembly. Furthermore, the two complementary interfaces corresponding to that symmetric binding mode will be implicitly treated as one single interface throughout the course of the algorithm.

COMPLEXITY The check for symmetric binding modes is done in the preprocessing phase of the algorithm. Overall, each pose at each interface has to be considered once for the symmetry check (RMSD calculation between two transformations is in $\mathcal{O}(1)$) and probably a second time when a symmetric interface is detected, hence, the overall complexity of the symmetry detection is in $\mathcal{O}(d)$ where $d = |\mathfrak{D}|$.

B.3 RING-STRUCTURE DETECTION

One of the underlying assumptions of our algorithm is, that complexes are highly connected. However, this is not always the case: for example, some of them form one-layered rings where each protein interacts only with its two neighbors in the ring. The simplest example would be a trimer, however, much larger rings can exist. The problem with such rings is that a transformation matching can only take place upon ring closure. Before, the algorithm has to rely on the interaction energies obtained from the docking alone.

Typically, the ring closure should occur when the final monomer is attached. However, depending on the docking poses, a premature transformation might also happen: for example, in a hexa-homomeric ring with ideal geometry, the angle between three sequentially attached monomers is exactly 60°. The angle between such a sequence of proteins in a corresponding penta-homomeric ring is exactly 54°. If the set of docking poses now contains one or several docking poses that cause a bend of this angle by 6° towards the center of the ring, we might find a matching transformation already after the attachment of the fifth monomer and not only after the sixth.

This is not only an unfavorable situation because it artificially and erroneously increases the overall match score, but it in fact stops the assembly process when distinct interfaces are used: when attaching a new monomer, not only the corresponding interfaces of receptor and ligand are locked but also those between ligand and any monomer with a matching docking pose. In a partial ring, only two distinct interfaces are open at any time, one at the initial and one at the most recently attached monomer; if more interfaces were open, at least one monomer could accept a third monomer, leading to a branching at that particular monomer and the underlying complex would be no one-layered ring. If a premature transformation matching happens, both of these interfaces will be locked, leaving no further interface to proceed with the attachment of the remaining monomers.

Hence, it is beneficial to investigate whether a given set of binding modes, corresponding protein types, and stoichiometries thereof can only result in a one-layered ring-like structure, and not in any other kind of complex with higher connectivity. For structures found to be ring-like, the search for matching transformations can then be disabled for all but the attachment of the last monomer.

Detecting such rings is done as follows: first, all terminal protein types, i.e., those that provide only one interface for attachment, are detected, because they cannot be part of a cycle. Then, for each non-terminal type, we try to find all potential cycles, starting with an initially empty path, as follows: i) the current protein type is added to the current path, ii) its stoichiometry is decremented, and iii) all protein types reachable via one of the interfaces of the current type are detected. These steps are then recursively repeated for each of the reachable types, until no further protein types are reachable, no more instances thereof are left, or more than one unique minimal cycle

has been detected. A minimal cycle hereby denotes the smallest possible cycle using a particular order of interfaces and protein types.

This procedure is based on the possible connections between interfaces and detects only ring-like structures for which we can guarantee that no branching leading to a ring can take place. Conversely, not every structure where a branching occurs is necessarily highly connected: for example, a homomeric complex may have a protein type with two distinct interfaces which can either alternate in a ring or lead to a branching. However, in such a case, the topology cannot be detected from the interface composition alone but only in complex assembly.

COMPLEXITY The check for terminal protein types requires the iteration over all protein types and is hence in $\mathcal{O}(t)$ where $t = |\mathfrak{P}|$. In the worst case, we have a homo-multimer with $i = |\mathfrak{I}|$ interfaces, where none leads to an ultimate ring closure. Hence, for a complex of size m, we have to consider each interface in each of the m iterations, leading to a worst case complexity of $\mathcal{O}(i^m)$ for the detection of cycles in non-terminal protein types. While in practice, this complexity is never reached because i) the number of interfaces is small, ii) cycles are found much earlier than in the last iteration, and iii) only two rings that are different under all rotational permutations of their members need to be found for the algorithm to stop, the total complexity in the worst case is $\mathcal{O}(i^m) + \mathcal{O}(t)$.

B.4 HIERARCHICAL CLASH CHECKING

When attaching a new monomer to a complex candidate, it may occur that the placement of this monomer is incompatible with the other monomers in the complex candidate, because it significantly overlaps with a particular monomer and produces steric clashes, i.e., large repulsive forces, with the atoms in that monomer. Hence, a procedure to determine such clashes is required. The first approach we implemented relied on a hash-grid approach, but proved to be impractical for our purposes [363]. However, the clash distance table for pairs of AMBER force field atom types described therein was kept and extended to all possible pairs of atom types.

We then decided to implement a clustering scheme based on a hierarchical subdivision of atom sets into spheres. The initial suggestion was made by André Müller (Computational Geometry/Computer Graphics, Mainz University), the underlying principles can be found in [374]. Based on these principles, we implemented the following hierarchical clash checking algorithm which uses a hierarchical sphere representation of (subsets of) a protein's atoms.

Such a sphere representation is especially comfortable when testing the intersection of two spheres. Because spheres are invariant under rotations, two spheres S_1 and S_2 with radii r_1 and r_2 and centers $C_1, C_2 \in \mathbb{R}^{(3,1)}$ intersect under transformations $\mathbf{T}_1$ and $\mathbf{T}_2$ if the distance of their transformed centers is less than the sum of their radii:

$$||(\mathbf{R}_1 \cdot C_1 + \mathbf{t}_1) - (\mathbf{R}_2 \cdot C_2 + \mathbf{t}_2)|| < r_1 + r_2 \tag{B.5}$$

where $\mathbf{R}_1$,$\mathbf{R}_2$ and $\mathbf{t}_1$,$\mathbf{t}_2$ denote the rotation and translation given by $\mathbf{T}_1$,$\mathbf{T}_2$ respectively.

Initially, all atoms of a protein are contained in one single root sphere. To make the intersection test efficient, we are looking for the minimum-bounding sphere, i.e., the

sphere with minimum radius comprising all atoms. This sphere is computed using an integrated version of the Smallest Enclosing Balls (SEB) software [375, 376].

Subsequently, the covariance matrix of the atoms in the sphere is computed and the vector n of the first principal component [377], i.e., the direction with the highest variance in the atomic coordinates, is determined. The set of atoms is then divided by the plane orthogonal to the first principal component through the center C of the sphere, yielding two hemispheres as shown in Fig. B.1. Each hemisphere contains a subset of the atoms, one those in the non-negative and the other those in the negative half-space w.r.t. that plane (denoted by P^{0+} and P^{-}, respectively). For example, an arbitrary point P lies on a plane defined by A and n as the normal vector or in its positive half-space P^{+}, if

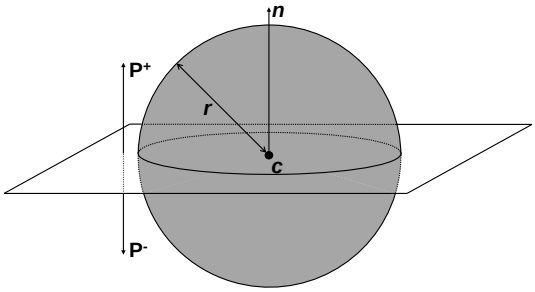


Figure B.1: Two hemispheres of a sphere of radius r w.r.t. a plane through center C with normal vector n.

$$(P - C) \cdot n \geq 0 \tag{B.6}$$

Consequently, all points for which this inequality does not hold are located in P^{-}.

For each of the two sets, a minimum-bounding sphere is computed, yielding two child spheres. The atom sets are again subdivided as described above and the whole process is repeated until the spheres contain a single atom (atomic spheres). Taking into account that the above SEB algorithm operates on point sets rather than on sets of atoms with volumes, we add the maximum radius of all atoms contained in the sphere to each sphere's radius.

The clash checking between two proteins is then performed as follows: first, the two root spheres are checked for intersection. If these two spheres do not intersect, no clashes can occur, because the spheres comprise all of the proteins' respective atoms. If an intersection between two parent spheres is detected, the four pairs of child spheres are tested for intersection. This process is recursively repeated as long as intersections are found, until the atomic spheres are reached. Because only intersecting spheres a further investigated, the clash checking quickly focuses on the contact region between two proteins.

A clash is then eventually reported if the distance between two atomic spheres is below the clash distance value of the corresponding AMBER atom types in the clash distance table. The clash checking stops either if no more tests have to be performed or if a predefined threshold for the allowed number of clashes has been reached.

COMPLEXITY Sphere trees are computed for each protein type in the preprocessing phase of the algorithm. The computation of a sphere containing n points in d dimensions can be done in $\mathcal{O}(d^2 n + e^{O(\sqrt{d \log d})})$ [378]. In our case, we are operating on points in three-dimensional space ($d = 3$ is fixed), hence the computation of a single sphere is linear in the number of points and thus takes $\mathcal{O}(n)$ time.

To divide a set S of n points within a sphere into two parts w.r.t. the above described half-planes, we must determine the principal components of S. This requires the calculation of a covariance matrix of the n points which can be done in $\mathcal{O}(n^2)$ and

the subsequent calculation of the eigenvectors, requiring $\mathcal{O}(n^3)$ (the used Eigen library [379] implements a Schur decomposition [380]). The subsequent division of the points in S in both subspheres requires $\mathcal{O}(n)$ time. We thus yield a total complexity of $\mathcal{O}(n^3)$.

For a hierarchical sphere tree to be constructed, in the worst case, i.e., an unbalanced scenario, we yield a tree of depth n. In each level, such a sphere has to be computed, followed by the division into two subsets. In the preprocessing phase of the 3D-MOSAIC algorithm, we thus have a complexity of $\mathcal{O}(n) \cdot (\mathcal{O}(n) + \mathcal{O}(n^3))$ which is in $\mathcal{O}(n^4)$ for the sphere tree construction of particular protein. For $t = |\mathfrak{P}|$ protein types, we thus have $\mathcal{O}(tn^4)$.

The actual clash-checking is applied in the iterative phase of 3D-MOSAIC. To check for clashes between to point sets of size n_1 and n_2, we must check all pairs of spheres in the worst case. This corresponds to the complexity of the trivial algorithm and requires $\mathcal{O}(n_1 \cdot n_2)$.

B.5 FINDING MATCHING TRANSFORMATIONS

To determine matching transformations, we use a grid-based approach: the Euclidean space is divided into axis-aligned cubic boxes with a certain edge length (spacing), each box filled with the set of docking poses whose transformation places the ligand's center of mass into the space confined in that box.

Each docking pose is initially represented by the ligand's center of mass $C = (0, 0, 0)$ (each ligand is centered at the origin of the coordinate system). A docking pose with transformation $\mathbf{T}$ and corresponding translation $\mathbf{t}$ then obviously translates C to $\mathbf{t}$, yielding c^*. For a given grid spacing $s \in \mathbb{R}$, the integer box coordinates are then given by $b = \lfloor t/s \rfloor$.

In addition, given a threshold for a maximum allowed displacement l_{max}, we also insert the pose into all neighboring boxes b_n which are less distant from c^* than l_{max}. All corresponding boxes must lie within or intersect a solid sphere of radius l_{max} around c^*, a fast and simple algorithm (including source code) to check for such intersections is presented in [381]. To obtain all such boxes, we simply have to check all neighboring boxes by stepwise moving outward from b until no more intersecting boxes can be found. The docking pose is then inserted into the box at position b and the corresponding neighboring boxes.

Hence, when a look-up is performed for a query pose transformation $\mathbf{T}'$, the grid box at position $b' = \lfloor t'/s \rfloor$ contains all docking poses $d \in \mathfrak{D}$ with $||\mathbf{t}_d - \mathbf{t}'|| \leq l_{max}$, where $\mathbf{t}_d, \mathbf{t}'$ denote the translations induced by the transformation $T(d)$ associated with d and the query transformation $\mathbf{T}'$, respectively.

However, due to the coarse-graining of the Euclidean space by the used grid-based approach, the boxes might also contain some poses that actually have larger deviations than l_{max}. This is a necessary overhead resulting from the need to ensure that the boxes contain at least all valid poses. Hence, each docking pose in the obtained pose set has to be post-checked to make sure that only docking pose transformations with a displacement of at most l_{max} are considered for the determination of the best-matching docking pose. Additionally, for the displacement/angle-based score, we have to discard all docking poses with an angular deviation larger than a_{max} from the query transformation (cmp. Sect. 10.3 and Eq. 10.5).

From all poses meeting the respective requirements, the docking pose d whose associated transformation $\mathbf{T}_d$ yields a maximum transformation match score to $\mathbf{T}$ (see Section 10.3 and Eq. 10.3) is then iteratively determined.

COMPLEXITY In the preprocessing phase, each pose has to be inserted into the corresponding grid boxes. Because we use a fixed l_{max}, the number of boxes into which a particular pose has to be inserted is constant, i.e., in $\mathcal{O}(1)$. Using a hash map to index the grid boxes, insertion takes amortized constant time, hence the overall insertion of $d = |\mathfrak{D}|$ also takes time $\mathcal{O}(d)$.

A look-up of a particular transformation during iterative assembly is also in $\mathcal{O}(1)$, however we may, in the worst case, obtain all d poses. The best-matching one has to be computed from all of these poses. Angular deviation, displacement as well as the RMSD calculation all take constant time, thus the overall complexity of a particular look-up including the subsequent determination of the best-matching pose is in $\mathcal{O}(d)$.

B.6 SCORING OF DOCKING POSES

In the beginning of the complex assembly, the set of docking poses can be considered to contain a large fraction of poses that do not correspond to a near-native binding mode and are thus not relevant for the complex assembly. However, typically, these poses must be considered, because the information whether a particular pose is useful cannot be deduced from the pose itself.

However, we have seen in the previous section that docking poses at interfaces of other monomers can help to determine whether a particular placement of a ligand monomer is reasonable in the context of the surrounding monomers.

For each such optimally matching pose, a score is obtained that can be used to gain information on the usefulness of that particular pose: poses with a high score can be considered to well support a particular complex topology. Hence, this information can help to discriminate between useful and unprofitable poses.

Because this topology is unknown at first and the optimal match score of each docking pose depends on the so-far assembled sub-complexes, their determination and updates are repeatedly carried out after each iteration. Due to the fact that they depend on the input data and the ultimate complex is unknown, this can be considered an unsupervised learning process.

The algorithm can then only consider the subset of the best-scoring N solutions per interface, where N is freely configurable by the user.

COMPLEXITY The actual scoring is performed during transformation matching. However the sorting of all $d = |\mathfrak{D}|$ poses according to their scores must be performed to determine the best N poses. Sorting d elements typically requires $\mathcal{O}(d \log_2 d)$ time.

B.7 INTERPOLATION BETWEEN TRANSFORMATIONS

The necessity of interpolating between a set of transformations may arise in two different stages of the algorithm: i) when a new monomer is attached to a receptor w.r.t. a docking pose and a corresponding set of matching transformations to interfaces of other monomers in the complex candidate has been found, or ii) when sets of symme-

try transformations (cmp. Def. 9.8) for the monomers in a complex candidate could be determined.

In the former case, the interpolation leads to a monomer placement that equally takes the docking pose and all matching transformations into account (the contributions could also be weighed by interaction energy, however, we consider each matching transformation to be equally important). In the latter case, the interpolation between all symmetry transformations for each monomer in the complex leads to a symmetry-optimized version of the complex candidate.

Let $\mathfrak{T}$ denote a set of transformations for which an interpolated transformation shall be generated. We denote the rotational and translational contributions of each transformation $\mathbf{T}_i \in \mathfrak{T}$ by $\mathbf{R}_i$ and $\mathbf{t}_i$.

The interpolation of the translations is straightforward, simply the average of all translations:

$$\bar{\mathbf{t}} = \frac{1}{|\mathfrak{T}|} \sum_{i=1}^{|\mathfrak{T}|} \mathbf{t}_i \tag{B.7}$$

To interpolate rotations, several methods exist [382]. The method we use throughout this work is presented in *Curtis et al.* [383] and, contrary to intuition, also relies on additive averaging. In the following, we will briefly describe the method. First the sum of all rotations is computed:

$$\mathbf{R}_{sum} = \sum_{i=1}^{|\mathfrak{T}|} \mathbf{R}_i \tag{B.8}$$

In a second step, a singular value decomposition is performed on $\mathbf{R}_{sum}$:

$$\mathbf{R}_{sum} = \mathbf{U}\mathbf{\Sigma}\mathbf{V} \tag{B.9}$$

Finally, the average rotation is obtained as:

$$\overline{\mathbf{R}} = \mathbf{U}\mathbf{V}^* \tag{B.10}$$

where $\mathbf{V}^*$ is the adjoint matrix of $\mathbf{V}$. The final average transformation is then a 4×4 matrix of the form:

$$\overline{\mathbf{T}} = \begin{pmatrix} \overline{\mathbf{R}} & \bar{\mathbf{t}} \\ \vec{0} & 1 \end{pmatrix} \tag{B.11}$$

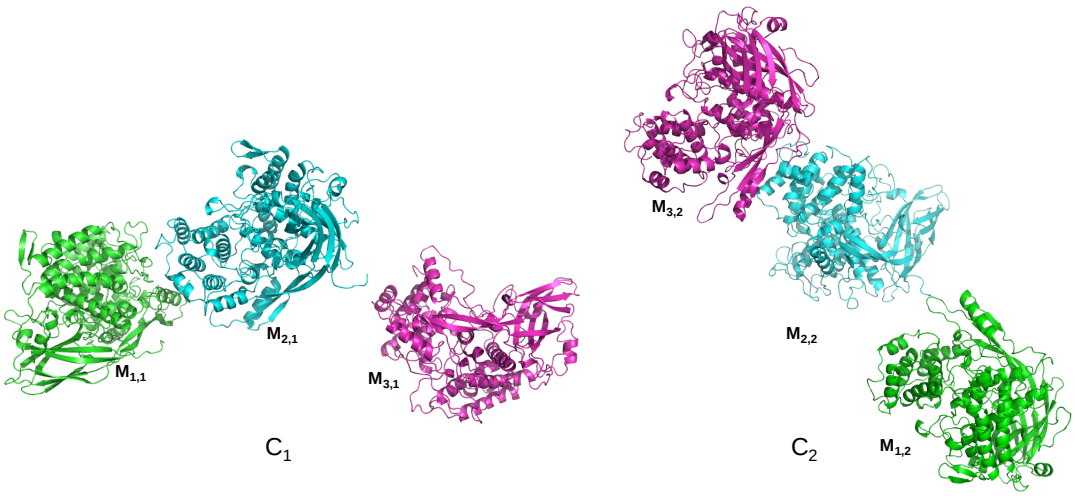
When interpolating a monomer placement during complex construction, the interpolation is only accepted if it does not produce severe steric clashes (i.e., more than a given treshold c_{max}) to any other monomer present in the complex candidate (see Sect. B.4). If the interpolation is accepted, the complex candidate with the interpolated monomer position replaces the original one.

COMPLEXITY In a complex of size m, a transformation can be matched by at most $m-1$ other transformations, yielding a total set of m transformations for the interpolation. Computing the sums of translations and rotations can be done in $\mathcal{O}(m)$ because a transformation matrix has a fixed size of 4×4 (the sum of two such matrices is then in $\mathcal{O}(1)$). Consequently, singular value decomposition of the summed rotation matrix can also be considered to be in $\mathcal{O}(1)$.

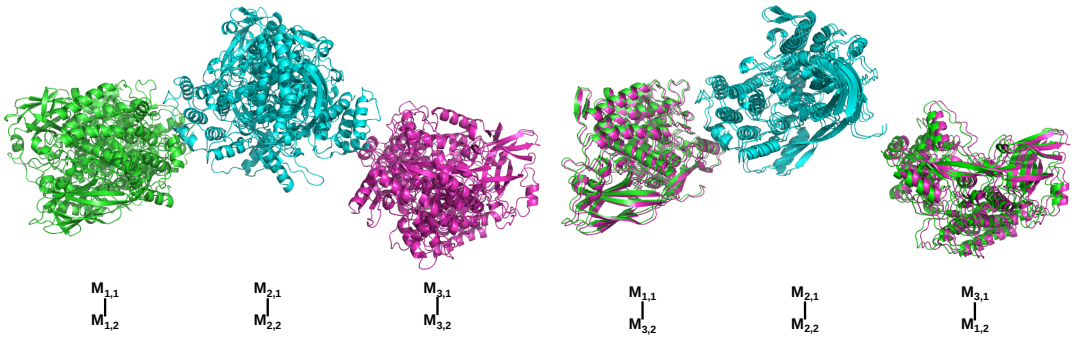
The subsequent clash checking of two particular monomers of sizes n_1 and n_2 takes $\mathcal{O}(n_1 \cdot n_2)$ time and has to be performed for $m \cdot (m-1)/2$ pairs in the worst case, which is in $\mathcal{O}(m^2)$. Let n denote the maximum number of atoms contained in any of the monomers, we can thus give the overall complexity by $\mathcal{O}(m) + \mathcal{O}(m^2) \cdot \mathcal{O}(n^2)$ which is in $\mathcal{O}(m^2 n^2)$.

B.8 STRUCTURAL MATCHING OF (SUB-)COMPLEXES

As already stated in Sect. 9.4, the correct structural matching between pairs of protein complexes is very important to maintain a diverse solution space. The problem we are faced with during the course of our algorithm is that two (sub-)complexes may have a similar overall topology but seem to be very different because the corresponding monomers have been attached in different iterations of the algorithm.



(a) Two complexes C_1 and C_2 with three monomers each. Monomers with the same color correspond to the same iteration (level) of attachment.



(b) Two possible matchings between the monomers of C_1 and C_2. Naïve (left): C_α RMSD 40.16. Optimal (right): C_α RMSD of 1.59.

Figure B.2: The problem of finding an optimal matching between the monomers of two complexes.

For example, consider two complexes $C1$ and $C2$ as shown in Fig. B.2a. The task is to find the RMSD-minimizing matching, i.e., the optimal complex similarity mapping (see Def. 9.5), between monomers $M_{i,1}$ and $M_{i,2}$ for $i \in \{1, ..., 3\}$. Fig. B.2b presents

two possible solutions, the naïve one with a C_α RMSD of 40.16 (left), the optimal one with a C_α RMSD of 1.59 (right).

The task of finding such a mapping is similar to the so-called point set registration [384], where an optimal superimposition between two sets of points is required. However, there are two main differences between our problem of determining a complex similarity mapping w.r.t. a similarity threshold d_{max} and the point set registration: in our case, we do not have independent points, rather, the atomic coordinates are grouped into point sets corresponding to monomers with an intrinsic order of atoms. Applying point set registration to such a problem would almost inevitably lead to mismatches of atoms from one chain of one complex to atoms from several chains in the second complex. Even when considering the chain in the second complex to which the majority of a chain in the first complex is matched as the matching chain, the matching might become ambiguous or even wrong.

Furthermore, we are not only interested in the optimal matching but also in sub-optimal ones, especially in the case of symmetry optimization: here all symmetry mappings (cmp. Def. 9.7) with an RMSD of at most d_{max}. We thus use a heuristic greedy two-stage algorithm to match the corresponding chains of two complexes with m components each.

To this end, each protein is represented in two different levels of coarse-graining: i) by its centroid c, ii) by six representative points (SRP representation), as follows. First, the principal components (pc) of the centered protein are calculated; subsequently each atomic position is orthogonally projected onto each pc. For each pc the orthogonal projection with the largest distance from the origin is retained, yielding three points p_1,p_2,p_3 w.r.t. to the three principal components. The six representatives are then obtained by $c \pm p_i, i \in \{1,2,3\}$.

In the first stage, we generate triplets of matching monomers from complexes C_1 and C_2 as follows. For each pair of monomers $M_{i,1}$, $M_{j,2}$ $i,j \in \{1,...,m\}$ in complexes C_1 and C_2 with the same protein type, we compare the respective distances to the complex centroid. If these distances are similar (deviate by less then a certain threshold; 20% by default), this pair of monomers represents an initial match $\varphi = \{(M_{i,1}, M_{j,2})\}$. The indices of the matched monomers from C_1 and C_2 are stored in two sets $I_{\varphi,1}$ and $I_{\varphi,2}$, i.e., $I_{\varphi,1} = \{i\}$ and $I_{\varphi,2} = \{j\}$ after the initial match.

Each matching is then iteratively extended to matching triplets as follows: the respective neighborhoods of the most recently added matching pairs (consisting of the six closest monomers and the corresponding distances) is investigated, and if for a pair $M_{k,1}$, $M_{l,2}$ $k,l \in \{1,...,m\}, k \notin I_{\varphi,1}, l \notin I_{\varphi,2}$ the respective distances to all previously added monomers and the centroid are similar and they belong to the same type of protein, we generate a new matching $\varphi' \leftarrow \varphi \cup \{(M_{k,1}, M_{l,2})\}$. Analogously, the index sets are updated: $I'_{\varphi,1} \leftarrow I_{\varphi,1} \cup \{k\}$ and $I'_{\varphi,2} \leftarrow I_{\varphi,2} \cup \{l\}$. This process is repeated once more for any so-obtained matching, yielding a unique set Φ_3 of matching triplets.

Having obtained such matching point triplets, we can now determine for each $\varphi \in \Phi_3$ an unambiguous RMSD-minimizing transformation $\mathbf{T}_\varphi$ between the respective centroids and sort the triplets by increasing RMSD. Given an RMSD tolerance threshold ϵ (15.0Å by default), and let $RMSD(\varphi_1)$ denote the centroid RMSD of the best-matching triplet φ_1, any triplet $\varphi \in \Phi_3$ with $RMSD(\varphi) > RMSD(\varphi_1) + \epsilon$ is discarded from Φ. This process is then repeated using the computationally more expen-

sive SRP representation. In doing so, we determine those triplets that provide good starting points to determine the complex similarity mappings Φ between C_1 and C_2.

In the second stage of the algorithm, we iteratively extend each matching triplet φ using the following greedy scheme: the centroids of both complexes are superimposed w.r.t. $\mathbf{T}_\varphi$, subsequently the pairwise distances between the centroids of any pair of unmatched monomers $M_{i,1}, i \notin I_{\varphi,1}$ and $M_{j,2}, j \notin I_{\varphi,2}$ belonging to the same protein type are computed and the matching is extended: the pair with the smallest distance is added to φ and the index sets of matched monomers are updated accordingly. $\mathbf{T}_\varphi$ is then recomputed w.r.t. the extended matching.

Any complete matching is added to the set of complex similarity mappings Φ which is then again subject to an outlier removal as described above; furthermore it is sorted according to the increasing RMSDs w.r.t. the coarse-grained SRP representation. Depending on the purpose, Φ can then be used during clustering (Sect. B.9), symmetry optimization, or evaluation against a reference complex.

COMPLEXITY In the preprocessing phase of 3D-MOSAIC, the protein representations for each protein type must be computed once. To obtain these representations for a particular protein with n atoms, we need to calculate the covariance matrix and the subsequent calculation of the eigenvectors, requiring $\mathcal{O}(n^3)$ (cmp. Section B.4). The centroid has already been obtained during covariance matrix computation, subsequent calculation of the six further points in the representation requires the orthongonal projection of each of the n points onto the three eigenvectors, which is in $\mathcal{O}(n)$. The preprocessing thus takes $\mathcal{O}(n^3) + \mathcal{O}(n)$ which is in $\mathcal{O}(n^3)$.

If two complexes with m monomers have to be matched, the neighbour lists of the m monomers in both complexes have to be calculated once. This entails the calculation and sorting of the distances of the respective centroids and is thus in $\mathcal{O}(m^2 \log_2(m^2))$ for both complexes. Subsequently, the initial matches are computed. In the worst case, all pairs of monomers from both complexes can be potential initial matches, leading to a complexity $\mathcal{O}(m^2)$. Then, they are extended by two matching triplets (i.e., by two additional monomers), which because of the neighbor list requires $\mathcal{O}(6^2 \cdot 6^2)$ which is in $\mathcal{O}(1)$ and the overall generation of matching triplets is thus still in $\mathcal{O}(m^2)$.

During attachment of each of the remaining $m - 3$ unmatched monomers in both complexes, the following is repeatedly performed, once for each pair to add, i.e., $m - 3 \in \mathcal{O}(m)$ times: the calculation of the optimally superimposing transformation w.r.t. the already matched points requires $\mathcal{O}(m^2)$ (in centroid and SRP representation), as does the subsequent computation of the distances between all remaining pairs of monomers (the number of monomers per complex is $m - 3 \in \mathcal{O}(m)$ in the worst case). This iterative extension thus takes $\mathcal{O}(m) \cdot \mathcal{O}(m^2)$ for each of the $\mathcal{O}(m^2)$ initial matching triples, yielding a total complexity of $\mathcal{O}(m^5)$.

Including the RMSD computation which will then be applied during clustering, symmetry optimization, and evaluation (see the following sections B.9, B.10, B.11), the overall complexity is increased by a factor n where n is the number of atoms contained in a fully assembled complex (i.e., the sum over the atom counts of all proteins times their respective stoichiometries), yielding a total complexity of $\mathcal{O}(m^5 n) + \mathcal{O}(m^2 \log_2(m^2)) = \mathcal{O}(m^5 n)$.

B.9 CLUSTERING OF (SUB-)COMPLEXES

After every iteration of the algorithm, the obtained solutions can be clustered to obtain a diverse solution space. Due to the potentially large number of complex candidates, an all-vs-all clustering is computationally intractable. Hence we compare each complex candidate only to the previously retained cluster representatives, following an all-vs-first strategy.

Let the list L of complex candidates be sorted by decreasing complex match score. Then, intuitively, the first complex candidate c_1 from L is the initial element in the list R of representatives, $R = [c_1]$.

Given an RMSD-threshold d_{max}, for each following complex candidate c, we iterate over all previous representatives c_r from L and compute the complex similarity mappings Φ between c and any c_r.

For each mapping $\varphi \in \Phi$ for c and a particular c_r, the C_α-RMSD $RMSD(\varphi)$ between c_r and c is computed. If $RMSD(\varphi) \leq d_{max}$, c is discarded, otherwise if $RMSD(\varphi) > d_{max}\ \forall \varphi \in \Phi$, L is extended by a new representative: $L \leftarrow L + [c]$.

As soon as the required number k of solutions to be retained for the next level is reached, the clustering procedure terminates.

COMPLEXITY We denote the number of complex candidates obtained after the most recent iteration of level population by u. In the worst case, each candidate has to be compared to $k - 1 \in \mathcal{O}$ other complexes (the retained representatives), where k is the number of solutions to be retained. Hence the total number of matchings is in $\mathcal{O}(ku)$.

The structural matching including RMSD computation takes $\mathcal{O}(m^5 n)$ (Section B.8), hence the overall complexity of clustering is $\mathcal{O}(m^5 nku)$.

B.10 SYMMETRY OPTIMIZATION

As already stated in Sect. 9.5, many complexes exhibit at least partial symmetries. When modeling macromolecular oligomeric assemblies from binary dockings, where the native binding modes are only roughly known, we often obtain structures that are to some extent distorted as compared to an ideal symmetry. Consequently, optimizing the symmetry of such a complex can help to improve the structural quality of the model.

To this end, given a complex C, centered at the origin, we can first determine the set Φ of all complex similarity mappings of C onto itself as described in Sect. B.8. From Φ, we can then construct the set S of symmetry mappings and the corresponding symmetry transformations T_S for C as explained in Def. 9.7.

Let m be the number of monomers in C, we now have to determine the symmetry-optimized placement of each monomer $M_i, i \in \{1, ..., m\}$. Each mapping $\varphi \in S$ maps M_i onto a different monomer $M_{\varphi(i)}$ in the C. Consequently, $T_{i,\varphi(i)} := T_S(\varphi) \cdot T_i$ represents an alternative placement of $M_{\varphi(i)}$ w.r.t. to symmetry transformation $T_S(\varphi)$ (Def. 9.7) induced by φ and the transformation T_i representing the placement of monomer M_i in C.

Let S_i denote the set of symmetry placements for monomer M_i, i.e., $S_i := \left\{ T_{h,\varphi(h)} | \varphi \in S^* \wedge \varphi(h) = i \right\}$, we can interpolate the transformation for the symmetry-optimized placement of M_i from S_i as described in Sect. B.7.

S^* is determined in an iterative fashion: initially S^* contains the RMSD-minimal mapping from S. In every iteration, the mapping φ' with the next-smallest RMSD is included in S^*, each S_i is updated and the corresponding symmetry-optimizing transformations are re-interpolated for the new S_i. If the inclusion of φ' produces severe steric clashes for any pair of monomers in the symmetry-optimized complex, φ' is discarded from S^*.

Finally, a complex C^* which is symmetry-optimized w.r.t. S^* is returned.

COMPLEXITY The determination of all potential symmetry mappings is performed as described in Section B.8, yielding a complexity of $\mathcal{O}(m^5 n)$, where m is the number of monomers in the complex and n is the overall number of atoms. The iterative extension of S^* requires the re-interpolation of all symmetry-optimizing transformations of all m. Each monomer can have at least m such transformations, hence one such an interpolation takes $\mathcal{O}(m)$, for m monomers thus $\mathcal{O}(m^2)$. The subsequent clash checking entails a pairwise comparison of all n atoms in both complexes and is thus in $\mathcal{O}(n^2)$. Hence, the overall complexity is $\mathcal{O}(m^5 n) \cdot (\mathcal{O}(m^2) + \mathcal{O}(n^2))$ which is in $\mathcal{O}(m^7 n) + \mathcal{O}(m^5 n^3)$.

B.11 COMPLEX EVALUATION AGAINST A REFERENCE

To evaluate a particular complex candidate c against a reference complex R, the following procedure is applied: using a pairwise sequence-alignment between the individual protein types of the complex and the corresponding chains in R, each reference chain is first superimposed to the input orientation of the corresponding protein type used for the assembly. For each chain, the centroid and SRP representation are then generated and transformed back to the original orientation of the corresponding chain.

We can then match each assembled complex candidate c onto the reference complex R as described in Sect. B.8. Each matching φ in the set of obtained complex similarity mappings Φ is then in turn applied to c which is then subsequently evaluated against R.

For each c, the evaluation generates a statistic containing the following basic measurements: the complex match score (CMS, see Eq. 11.17), the accumulated interface scores (AIS) over all docking poses used during assembly and matching (which is only an approximation due to the non-perfect matching), the overal number of clashes in the complex as well as the overall C_α and backbone RMSD to the reference.

Furthermore, it also comprises several standard measures from protein-protein interaction prediction, especially the CAPRI assessments [385, 386]: the fractions of native and non-native-contacts f_{nat} and $f_{non-nat}$ as well as the interface RMSD I_{rms}, using an heavy-atom contact range of 5Å to identify interacting residues.

In addition, we provide the number of native and non-native interfaces, as well as the RMSD obtained from the SRP representation, called P_{rms}. Both, I_{rms} and P_{rms} are available in a global and local version: the global version is based on the optimal superimposition of the whole complex, the local version on an average RMSD over the optimal superimpositions of any pair of corresponding interacting dimers in R and c.

Finally, the statistic provides a measure called topology-RMSD (tRMSD), which is the main measure on which the results presented in this thesis rely. This measure is described in detail in Sect. 11.2.9.

COMPLEXITY Again the structural matching between the reference and the complex to be evaluated takes $\mathcal{O}(m^5 n)$ time, with m the number of monomer in the complexes and n the total number of atoms. For the dimers interacting in the reference, the topology RMSD is calculated; the maximum number of pairs of interacting monomers is obviously in $\mathcal{O}(m^2)$ and because, in the case of the tRMSD, each monomer is represented by seven points (see Subsection 11.2.9), the computation of all contributions to the tRMSD also takes $\mathcal{O}(m^2)$ time. All other RMSD and interaction computations are performed pairwise on a subset of all n atoms in both complexes, leading to a $\mathcal{O}(n^2)$ complexity.

In total, we thus have for the evaluation of a particular complex a complexity of $\mathcal{O}(m^5 n) + \mathcal{O}(m^2) + \mathcal{O}(n^2)$ which is in $\mathcal{O}(m^5 n) + \mathcal{O}(n^2)$.

B.12 RESTART FILES

A typical run of 3D-MOSAIC performs a complex assembly from scratch and only stops when the full complex is assembled. However, in some cases it might prove useful to stop and resume at sub-complexes of a certain size. Such a feature is especially useful when parameter options, e.g., thresholds for clustering have to be changed or disabled. In addition, in queuing systems where the jobs on the queues must not exceed a certain running time, the generation of intermediate checkpoints from which the assembly can be restarted, is especially important.

Furthermore, such restart files provide the possibility to first generate core protein complexes: the stoichiometries of protein types not corresponding to that core can be initially set to zero to prevent them from being considered for attachment. Hence, the assembly only uses the core protein types with a non-zero stoichiometry. Once such complexes have been assembled, the resulting restart file can, after reset of the non-core types to the original stoichiometries act as the anchor point for the attachment of the satellite proteins.

Moreover, an expert-guided selection of promising complex candidates (the restart files use a simple text-based format and are thus easily editable) can be effortlessly accomplished.

Such a restart file contains all relevant information on the complex candidate tree, the complex candidates as defined in Sect. 9.3 and their relationship as well as the docking poses and their best match score obtained so far. Together with the restart file, the original input data must be provided.

B.13 ADDITIONAL FEATURES

3D-MOSAIC is highly configurable via various options, the most important ones being RMSD thresholds for pre-, intra- and post-clustering, clash checking, transformation matching parameters and symmetry optimization RMSD. These options include some additional features that are worth to be mentioned briefly.

First, while the possibility to generate restart files provides full control over the course of the algorithm, some of the parameters should adapt themselves in each iteration. The most important feature here is the the solution reduction scheme the algorithm provides: starting with a predefined number of solutions to be retained after attachment of the first ligand, the user can specify a factor by which this number

should be reduced in any subsequent iteration, until a final lower bound of solutions has been reached. This feature takes account of the fact that the algorithm has to rely on the docking scores alone during the first iteration, but also considers that the most plausible solutions should already be among the first ranks of the solutions set, which reduces the overall number of candidates to be considered for the next iteration.

In addition, several assembly order modes can be employed. By default, the algorithm allows attachment of new monomers at all available interfaces of a complex candidate. Two more modes are available: i) attachment is only allowed at the most recently attached protein, and ii) a strict order is estimated based on several heuristics. These include, in decreasing order of importance, usage of different interfaces for subsequent attachments, balancing of protein stoichiometries, palindromic assembly orders (in analogy to the structural symmetry of complexes), and the number of potential matches to previously added monomers. In both modes, dead-ends can be possible, for example if proteins with terminal nodes, i.e., ones that provide only interactions to one other protein, are encountered. If such cases are detected, the default attachment algorithm is used as a fall-back.

A second feature is the re-ranking of complex candidates by the number symmetry mappings that can be determined for a complex candidate. In situations where the set of underlying docking poses results from a docking run performing coarse wide-range sampling, docking poses corresponding to near-native binding modes can be expected to be scarce relative to the total number of docking poses. Here, the probability for matching non-native docking poses is greatly increased and can reduce the discriminative power of the transformation match score. However, as already stated, many complexes exhibit (partial) symmetries, and detecting such symmetries in early stages of the algorithm can help to discriminate between near-native sub-complexes and decoys. Though, due to the computational complexity, this feature is only recommendable for early iterations.

Third, after a complex has been assembled, a model refinement may be required, especially regarding the side-chain orientations in the contact regions between individual monomers. To this end, 3D-MOSAIC implements a side-chain optimization of the first n complexes using SCWRL and a subsequent re-evaluation of the complex w.r.t. a reference complex.

Finally, it benefits from the free choice of the docking algorithm, which is intentionally left to suit the user's preferences. Several application scenarios are thinkable here: low or intense docking samplings, local pertubation runs, global dockings, as well as the combination of docking results from different scenarios and/or algorithms.

B.14 IMPLEMENTATION IN BALL

The 3D-MOSAIC algorithm is implemented in C++ as part of the open-source framework BALL (**B**iochemical **Al**gorithms **L**ibrary) [298]. Besides the main algorithm, an all-in-one tool (MOSAICInputGenerator) to easily generate the required input data from the sets of pairwise dockings, the protein stoichiometries, and, in case an evaluation against a reference is requested, a sequence alignment between the monomers and the corresponding chains in the reference complex is provided.

Within BALL, the algorithm can be swiftly extended to employ subsequent energy minimizations, MD simulations, or docking of small-molecules as well as immediate visual inspection. Its highly object-oriented design allows for quick adaptations to

specific needs, even the use of stand-alone components such as the hierarchical clustering or the complex clustering is effortlessly possible. Finally, the integration into external pipelines can be easily achieved using BALLaxy, BALL's gallaxy interface.

C

PERFORMANCE AND PARAMETER DETAILS FOR EXPERIMENTS WITH 3D-MOSAIC

This chapter contains supplementary information on parameter sets, performances and data sets.

C.1 DATA SETS

	Number of						
Assembly	Protein types	Monomers	Binding modes	Monomer sources	Quality	Date of deposition	Monomer PDB sources
1A5L_1_foreign	3	9	4	1	0.230	1998-02-17	1A5O
1A5M_1_same	3	9	4	1	0.267	1998-02-17	1A5M
1A8R_1_foreign	1	10	2	1	0.230	1998-03-27	1N3R
1A92_1_same	1	8	3	1	0.277	1998-04-15	1A92
1AHV_1_same	1	8	3	1	0.083	1997-04-10	1AHV
1AUS_1_foreign	2	16	7	2	0.216	1995-06-21	1RCO, 1RCX
1AVO_1_same	2	14	3	1	0.068	1997-09-18	1AVO
1B4A_1_same	1	6	3	1	0.121	1998-12-18	1B4A
1B4F_1_unbound	1	8	2	1	0.240	1998-12-20	1F0M
1B5S_1_same	1	60	2	1	-0.403	1999-01-10	1B5S
1BE3_2_same	11	22	25	1	0.013	1998-05-19	1BE3
1BGG_1_foreign	1	8	2	1	0.185	1997-05-12	1BGA
1C3K_2_foreign	1	8	2	1	0.245	1999-07-28	1C3N
1C3K_2_same	1	8	2	1	0.245	1999-07-28	1C3K
1COA_1_same	1	6	3	1	0.269	1999-09-28	1EOI
1DE4_1_same	3	6	3	1	0.092	1999-11-12	1DE4
1DM5_1_same	1	6	2	1	0.255	1999-12-13	1DM5
1E32_1_foreign	1	6	1	1	0.062	2000-06-05	1S3S
1E32_1_same	1	6	1	1	0.062	2000-06-05	1E32
1E4I_1_same	1	8	2	1	0.250	2000-07-06	1E4I
1E7P_1_same	3	6	5	1	0.032	2000-09-01	1E7P
1EAA_1_foreign	1	24	2	1	0.159	1992-12-16	1EAC
1EXB_1_same	2	8	3	1	0.249	2000-05-02	1EXB
1FR9_1_unbound	1	8	3	1	0.364	2000-09-07	1E5K
1FSF_1_foreign	1	6	2	1	0.294	2000-09-08	1FS5
1FSF_1_unbound	1	6	2	1	0.294	2000-09-08	2WU1
1FX0_1_foreign	2	6	2	1	-0.037	2000-09-25	1KMH
1FX0_1_same	2	6	2	1	-0.037	2000-09-25	1FX0
1FZE_1_foreign	3	6	4	2	0.015	1998-12-23	2XNY, 1RF0
1FZE_1_same	3	6	4	1	0.015	1998-12-23	1FZE
1G31_2_same	1	14	2	1	0.181	1998-03-27	1G31
1GQ6_1_foreign	1	6	3	1	0.401	2001-11-20	1GQ7
1GQ6_1_same	1	6	3	1	0.401	2001-11-20	1GQ6
1GQM_1_foreign	1	6	2	1	0.089	2001-11-26	2WC8
1GQM_1_dimer	1	6	2	1	0.089	2001-11-26	2WCF
1GQM_1_same	1	6	2	1	0.089	2001-11-26	1GQM
1GUT_1_foreign	1	6	3	1	0.445	2002-01-28	1GUN
1GUT_1_same	1	6	3	1	0.445	2002-01-28	1GUT
1H05_1_foreign	1	12	2	1	0.476	2002-06-11	3N86
1H2I_1_foreign	1	11	2	1	0.108	2002-08-09	1KN0
1H2I_1_same	1	11	2	1	0.108	2002-08-09	1H2I
1HI9_1_same	1	10	3	1	0.149	2001-01-04	1HI9
1HKX_1_same	1	14	2	1	0.098	2003-03-12	1HKX
1HQK_1_foreign	1	60	3	1	0.389	2000-12-18	1NQW

Continued on next page

Continued from previous page

Assembly	Number of Protein types	Monomers	Binding modes	Monomer sources	Quality	Date of deposition	Monomer PDB sources
1HQK_1_same	1	60	3	1	0.389	2000-12-18	1HQK
1HX5_1_foreign	1	7	1	1	0.021	2001-01-11	1P3H
1I40_1_foreign	1	6	2	1	0.756	2001-02-19	2AU6
1IHP_2_unbound	1	6	2	1	0.189	1997-02-04	3K4P
1J1J_5_foreign	1	8	2	1	0.190	2002-12-06	3QB5
1J1J_5_same	1	8	2	1	0.190	2002-12-06	1J1J
1J70_1_foreign	1	6	3	1	0.168	2001-05-15	1JEC
1J70_1_same	1	6	3	1	0.168	2001-05-15	1J70
1JH5_2_foreign	1	60	3	1	0.081	2001-06-27	1OTZ
1JH5_2_same	1	60	3	1	0.081	2001-06-27	1JH5
1JPU_1_foreign	1	8	2	1	0.349	2001-08-03	1JQ5
1JSM_2_dimer	2	6	5	1	0.262	2001-08-17	1JSN
1JYO_1_same	2	6	5	1	0.268	2001-09-12	1JYO
1K6W_1_unbound	1	6	2	1	0.399	2001-10-17	3RN6
1KIB_1_same	1	24	2	1	0.063	2001-12-03	1KIB
1KP8_1_same	1	14	3	1	0.242	2001-12-30	1KP8
1KQ3_2_same	1	8	3	1	0.478	2002-01-03	1KQ3
1KW6_1_foreign	1	8	3	1	0.519	2002-01-28	1EIQ
1KW6_1_same	1	8	3	1	0.519	2002-01-28	1KW6
1L2W_5_same	2	6	5	1	0.225	2002-02-25	1L2W
1L7A_1_same	1	6	3	1	0.478	2002-03-14	1L7A
1L9V_1_same	1	8	2	1	0.098	2002-03-26	1L9V
1L9V_1_unbound	1	8	2	1	0.098	2002-03-26	4G0J
1LNL_1_same	1	6	3	1	0.015	2002-05-03	1LNL
1MGQ_1_foreign	1	7	1	1	0.327	2002-08-16	1I81
1MQM_2_foreign	2	6	5	2	0.095	2002-09-16	1MQL, 1MQN
1MQM_2_same	2	6	5	1	0.095	2002-09-16	1MQM
1MTY_1_same	3	6	6	1	0.366	1996-07-10	1MTY
1NOG_2_same	1	12	2	1	0.430	2003-01-16	1NOG
1NQT_3_foreign	1	6	3	1	0.010	2003-01-23	1NR7
1NQT_3_same	1	6	3	1	0.010	2003-01-23	1NQT
1NSF_1_foreign	1	6	1	1	0.282	1998-06-26	1D2N
1NTH_1_foreign	1	6	3	1	0.457	2003-01-30	1TV3
1NTH_1_same	1	6	3	1	0.457	2003-01-30	1NTH
1OFH_1_same	2	18	6	1	0.123	2003-04-14	1OFH
1OGC_1_foreign	1	10	3	1	0.269	2003-04-30	1OGF
1OGC_1_same	1	10	3	1	0.269	2003-04-30	1OGC
1P3H_3_same	1	14	3	1	0.077	2003-04-17	1P3H
1PKH_1_foreign	1	6	3	1	0.501	2003-06-05	1OGH
1PKH_1_same	1	6	3	1	0.501	2003-06-05	1PKH
1PMM_1_same	1	6	3	1	0.287	2003-06-11	1PMM
1POI_1_same	2	8	4	1	0.171	1997-01-24	1POI
1PVV_1_foreign	1	12	3	1	0.266	2003-06-29	1A1S
1PVV_1_same	1	12	3	1	0.266	2003-06-29	1PVV
1QK1_1_same	1	8	3	1	0.151	1999-07-08	1QK1
1QW9_1_foreign	1	6	2	1	0.654	2003-09-01	1QW8
1QW9_1_same	1	6	2	1	0.654	2003-09-01	1QW9
1R4C_5_foreign	1	8	3	1	0.200	2003-10-06	1G96
1RA0_1_foreign	1	6	2	1	0.721	2003-10-31	1R9X
1RGX_2_same	1	6	3	1	0.314	2003-11-13	1RGX
1RYP_1_foreign	14	28	35	5	0.196	1997-02-26	1Z7Q, 1FNT, 3L5Q, 3UN4, 1VSY
1RYP_1_same	14	28	35	1	0.196	1997-02-26	1RYP
1S2L_1_foreign	1	6	2	1	0.201	2004-01-08	1S2G
1S2L_1_same	1	6	2	1	0.201	2004-01-08	1S2L
1STM_1_same	1	60	3	1	0.277	1995-07-12	1STM
1SVD_2_same	2	16	6	1	0.390	2004-03-29	1SVD
1SVT_1_foreign	2	21	6	1	0.082	2004-03-29	1PCQ
1SVT_1_same	2	21	6	1	0.082	2004-03-29	1SVT
1SX3_1_foreign	1	14	3	1	0.235	2004-03-30	1SS8
1T6Q_1_foreign	1	6	3	1	0.242	2004-05-07	1T6U
1T6Q_1_same	1	6	3	1	0.242	2004-05-07	1T6Q

Continued on next page

Continued from previous page

Assembly	Number of Protein types	Monomers	Binding modes	Monomer sources	Quality	Date of deposition	Monomer PDB sources
1T9G_1_same	3	6	4	1	0.082	2004-05-17	1T9G
1TH7_1_same	1	7	1	1	0.356	2004-06-01	1TH7
1TR0_1_foreign	1	12	3	1	0.354	2004-06-18	1SI9
1TZY_1_same	4	8	8	1	0.304	2004-07-12	1TZY
1U11_2_foreign	1	8	3	1	0.446	2004-07-14	2FWJ
1U11_2_same	1	8	3	1	0.446	2004-07-14	1U11
1U6I_1_foreign	1	6	3	1	0.220	2004-07-30	3IQE
1U6I_1_same	1	6	3	1	0.220	2004-07-30	1U6I
1UMR_3_foreign	2	8	2	1	0.153	2003-08-28	1UOS
1UP8_1_same	1	12	4	1	0.232	2003-09-29	1UP8
1V4L_1_same	2	8	2	1	0.059	2003-11-14	1V4L
1V7Z_1_foreign	1	6	2	1	0.427	2003-12-26	1J2U
1V7Z_1_same	1	6	2	1	0.427	2003-12-26	1V7Z
1VAO_1_foreign	1	8	3	1	0.110	1997-04-10	1AHV
1VDM_1_same	1	12	3	1	0.159	2004-03-23	1VDM
1WPB_1_same	1	24	3	1	0.273	2004-09-01	1WPB
1WPS_1_foreign	1	6	3	1	0.097	2004-09-13	1VEA
1WPS_1_same	1	6	3	1	0.097	2004-09-13	1WPS
1WRV_2_foreign	1	6	3	1	0.449	2004-10-27	2EJ3
1WRV_2_same	1	6	3	1	0.449	2004-10-27	1WRV
1X1O_1_same	1	6	3	1	0.280	2005-04-08	1X1O
1X36_1_foreign	1	60	3	1	0.123	2005-04-29	1VB4
1X36_1_same	1	60	3	1	0.123	2005-04-29	1X36
1X9F_1_foreign	4	12	7	1	0.139	2004-08-20	2GTL
1X9J_1_same	1	8	2	1	0.049	2004-08-21	1X9J
1XSJ_1_foreign	1	6	2	1	0.256	2004-10-19	1XSK
1XSJ_1_same	1	6	2	1	0.256	2004-10-19	1XSJ
1XXC_1_foreign	1	6	2	1	-0.013	1995-11-03	1XXA
1XXC_1_same	1	6	2	1	-0.013	1995-11-03	1XXC
1Y88_2_same	1	6	3	1	0.308	2004-12-10	1Y88
1YG6_1_foreign	1	14	2	1	0.275	2005-01-04	2FZS
1YG6_1_same	1	14	2	1	0.275	2005-01-04	1YG6
1YHU_1_same	4	24	8	1	0.046	2005-01-10	1YHU
1YI5_1_same	2	10	3	1	-0.140	2005-01-11	1YI5
1YNB_2_same	1	6	3	1	0.325	2005-01-24	1YNB
1YNT_1_same	4	7	5	1	0.040	2005-01-25	1YNT
1YQ2_1_same	1	6	2	1	0.331	2005-02-01	1YQ2
1YZV_2_same	1	24	2	1	0.313	2005-02-28	1YZV
1Z6B_4_foreign	1	6	2	1	0.256	2005-03-22	3AZ8
1Z7Q_1_same	15	42	50	1	0.003	2005-03-26	1Z7Q
1ZCC_1_same	1	6	3	1	0.119	2005-04-11	1ZCC
1ZKE_4_same	1	6	2	1	0.393	2005-05-02	1ZKE
1ZYE_1_same	1	24	4	1	0.038	2005-06-10	1ZYE
2A2L_1_same	1	8	3	1	0.147	2005-06-22	2A2L
2A6Q_3_same	2	6	5	1	0.234	2005-07-04	2A6Q
2AEQ_1_same	3	12	4	1	0.021	2005-07-23	2AEQ
2AHM_1_same	2	16	10	1	0.166	2005-07-28	2AHM
2AVU_1_same	2	6	5	1	0.078	2005-08-30	2AVU
2BDN_1_same	3	6	4	1	0.118	2005-10-20	2BDN
2BJK_1_foreign	1	6	3	1	0.544	2005-02-04	2EHU
2BJK_1_same	1	6	3	1	0.544	2005-02-04	2BJK
2BM8_1_foreign	1	6	2	1	0.155	2005-03-10	2BM9
2BM8_1_same	1	6	2	1	0.155	2005-03-10	2BM8
2BOB_1_foreign	3	12	7	3	0.111	2005-04-09	3IGA, 3OR6, 2ATK
2BSE_1_foreign	2	6	2	2	0.076	2005-05-20	2X54, 2WZP
2BSE_1_same	2	6	2	1	0.076	2005-05-20	2BSE
2BUK_1_foreign	1	60	3	1	0.094	2005-06-14	3RQV
2BWE_1_same	2	10	2	1	0.056	2005-07-13	2BWE
2CoW_1_same	1	55	2	1	0.064	2005-09-08	2CoW
2CB5_2_same	1	6	3	1	0.331	1999-03-02	2CB5
2CHV_1_foreign	1	6	1	1	-0.248	2006-03-16	2CHQ

Continued on next page

Continued from previous page

Assembly	Number of Protein types	Monomers	Binding modes	Monomer sources	Quality	Date of deposition	Monomer PDB sources
2CZ8_1_foreign	1	12	2	1	0.438	2005-07-11	2DEG
2D00_2_same	1	12	3	1	0.193	2005-07-21	2D00
2D69_1_foreign	1	8	3	1	0.317	2005-11-10	2CXE
2D69_1_same	1	8	3	1	0.317	2005-11-10	2D69
2F1D_1_same	1	24	3	1	0.047	2005-11-14	2F1D
2FY8_1_dimer	1	8	2	1	0.104	2006-02-07	3RBX
2FZ6_5_foreign	1	8	4	1	0.200	2006-02-09	2GVM
2FZ6_5_same	1	8	4	1	0.200	2006-02-09	2FZ6
2G0J_3_same	1	8	4	1	0.098	2006-02-13	2G0J
2G9T_1_dimer	1	12	3	1	0.229	2006-03-07	3R24
2GJV_2_same	1	6	1	1	0.169	2006-03-31	2GJV
2GMY_1_same	1	6	2	1	0.434	2006-04-07	2GMY
2H1L_1_foreign	2	12	2	2	0.008	2006-05-16	1KZY, 4E2I
2H1L_1_same	2	12	2	1	0.008	2006-05-16	2H1L
2H64_1_same	3	6	4	1	0.257	2006-05-30	2H64
2H7C_1_foreign	1	6	2	1	0.279	2006-06-02	1YA4
2H85_1_same	1	6	2	1	0.145	2006-06-06	2H85
2HDA_2_same	1	6	2	1	0.256	2006-06-20	2HDA
2HEX_2_foreign	1	10	3	1	0.188	1998-08-01	1BOC
2HEX_2_dimer	1	10	3	1	0.188	1998-08-01	1EAW
2HEX_2_same	1	10	3	1	0.188	1998-08-01	2HEX
2HEX_2_unbound	1	10	3	1	0.188	1998-08-01	5PTI
2HEY_1_foreign	2	6	3	2	0.253	2006-06-22	2HEW, 2HEV
2HEY_1_same	2	6	3	1	0.253	2006-06-22	2HEY
2HFN_1_foreign	1	10	2	1	0.290	2006-06-24	2HFO
2HFN_1_same	1	10	2	1	0.290	2006-06-24	2HFN
2HMV_1_foreign	1	8	2	1	0.210	2006-07-11	2HMT
2HMV_1_same	1	8	2	1	0.210	2006-07-11	2HMV
2HMZ_1_foreign	1	8	3	1	0.387	1990-10-18	2HMQ
2HMZ_1_same	1	8	3	1	0.387	1990-10-18	2HMZ
2HY5_1_foreign	3	6	7	1	0.370	2006-08-04	2HYB
2HY5_1_same	3	6	7	1	0.370	2006-08-04	2HY5
2I00_4_same	1	6	2	1	0.165	2006-08-09	2I00
2I1O_1_foreign	1	6	3	1	0.167	2006-08-14	1YTE
2IBZ_1_same	11	11	19	1	0.179	2006-09-12	2IBZ
2J12_1_foreign	2	6	3	2	0.498	2006-08-08	1UXA, 1KAC
2J12_1_dimer	2	6	3	2	0.498	2006-08-08	2WBW, 1P6A
2J12_1_same	2	6	3	1	0.498	2006-08-08	2J12
2JB7_1_foreign	1	6	2	1	0.405	2006-12-04	2GL0
2JB7_1_same	1	6	2	1	0.405	2006-12-04	2JB7
2NS1_1_same	2	6	3	1	0.312	2006-11-02	2NS1
2NUU_3_foreign	2	6	3	2	0.151	2006-11-09	1GNK, 2NS1
2NV1_1_foreign	1	12	3	1	0.290	2006-11-10	2NV2
2O39_1_foreign	2	6	3	2	0.078	2006-12-01	3INB, 3EXV
2O39_1_same	2	6	3	1	0.078	2006-12-01	2O39
2PO0_1_foreign	2	6	2	1	0.192	2007-04-25	2PNZ
2PO0_1_same	2	6	2	1	0.192	2007-04-25	2PO0
2QGF_1_same	2	12	6	1	0.232	2007-06-28	2QGF
2QMH_1_same	1	6	3	1	0.087	2007-07-16	2QMH
2QVJ_1_same	1	10	2	1	0.071	2007-08-08	2QVJ
2RSL_1_same	1	6	2	1	0.196	1993-09-08	2RSL
2UX9_1_same	1	12	2	1	0.521	2007-03-27	2UX9
2UYU_1_foreign	1	8	3	1	0.265	2007-04-20	2V9N
2UYU_1_same	1	8	3	1	0.265	2007-04-20	2UYU
2V1W_1_same	1	8	3	1	0.315	2007-05-30	2V1W
2V78_1_foreign	1	6	3	1	0.302	2007-07-27	2VAR
2V78_1_same	1	6	3	1	0.302	2007-07-27	2V78
2VHX_1_foreign	1	6	3	1	0.290	2007-11-26	2VHW
2VHX_1_same	1	6	3	1	0.290	2007-11-26	2VHX
2VWS_1_foreign	1	6	3	1	0.528	2008-06-26	2VWT
2WCD_1_same	1	12	1	1	0.060	2009-03-11	2WCD

Continued on next page

Continued from previous page

Assembly	Number of Protein types	Monomers	Binding modes	Monomer sources	Quality	Date of deposition	Monomer PDB sources
2WQA_1_foreign	2	6	6	2	0.098	2009-08-14	3BSZ, 1ICT
2WQA_1_same	2	6	6	1	0.098	2009-08-14	2WQA
2X6L_1_same	1	10	3	1	0.125	2010-02-17	2X6L
2XW6_1_foreign	1	6	3	1	0.765	2010-11-01	2X8W
2XW6_1_same	1	6	3	1	0.765	2010-11-01	2XW6
2Z9H_1_same	1	6	1	1	0.067	2007-09-20	2Z9H
2ZBT_1_same	1	12	3	1	0.396	2007-10-29	2ZBT
2ZTD_1_foreign	1	8	2	1	0.149	2008-10-01	2ZTE
2ZTE_1_same	1	8	2	1	0.005	2008-10-01	2ZTE
3A9S_1_foreign	1	6	3	1	0.438	2009-11-05	3A9R
3A9S_1_same	1	6	3	1	0.438	2009-11-05	3A9S
3AQD_1_foreign	1	11	1	1	0.045	2010-10-29	2ZD0
3AQD_1_same	1	11	1	1	0.045	2010-10-29	3AQD
3B07_1_same	2	8	2	1	0.166	2011-06-06	3B07
3BE7_1_same	1	8	3	1	0.155	2007-11-16	3BE7
3BH3_1_same	1	12	3	1	0.225	2007-11-27	3BH3
3BJK_1_same	1	6	3	1	0.316	2007-12-04	3BJK
3BP9_1_same	1	6	1	1	0.093	2007-12-18	3BP9
3BXV_1_same	1	24	3	1	0.157	2008-01-14	3BXV
3CIM_1_foreign	1	6	1	1	0.547	2008-03-11	3DNC
3D03_1_foreign	1	6	3	1	0.303	2008-04-30	2ZO9
3D5O_1_foreign	2	6	2	2	0.078	2008-05-16	3RY6, 2A3X
3D5O_1_same	2	6	2	1	0.078	2008-05-16	3D5O
3D6N_1_same	2	12	5	1	0.231	2008-05-20	3D6N
3DBY_1_same	1	6	3	1	0.218	2008-06-02	3DBY
3DDO_1_same	1	6	2	1	0.455	2008-06-06	3DDO
3EJ3_1_foreign	2	6	3	2	0.370	2008-09-17	1S0Y, 3EJ9
3EJ9_1_same	2	6	3	1	0.420	2008-09-17	3EJ9
3EK6_1_foreign	1	6	2	1	0.181	2008-09-18	3EK5
3EK6_1_same	1	6	2	1	0.181	2008-09-18	3EK6
3EVO_1_foreign	1	6	2	1	0.482	2008-10-13	3EJM
3EVO_1_same	1	6	2	1	0.482	2008-10-13	3EVO
3EZZ_1_same	1	24	2	1	0.103	2008-10-24	3EZZ
3F9K_13_same	2	9	6	1	0.079	2008-11-14	3F9K
3GCB_1_foreign	1	6	3	1	0.318	1998-02-27	1A6R
3GE3_1_foreign	4	8	7	4	0.465	2009-02-24	3GE8, 3RMK, 3I63, 3DHH
3GE3_1_same	4	8	7	1	0.465	2009-02-24	3GE3
3H47_1_foreign	1	6	1	1	0.257	2009-04-18	3H4E
3H4E_1_same	1	6	1	1	0.107	2009-04-19	3H4E
3H8G_1_foreign	1	6	3	1	0.494	2009-04-29	3H8F
3H8G_1_same	1	6	3	1	0.494	2009-04-29	3H8G
3KA3_1_foreign	1	24	3	1	0.555	2009-10-18	3RE7
3L7Z_1_same	3	9	4	1	0.126	2009-12-29	3L7Z
3LWZ_1_same	1	12	2	1	0.424	2010-02-24	3LWZ
3M0E_1_same	1	7	1	1	0.139	2010-03-02	3M0E
3M4B_3_same	1	12	2	1	0.144	2010-03-10	3M4B
3M4D_1_foreign	1	7	1	1	0.261	2010-03-10	3M4E
3N2N_1_same	1	6	2	1	0.327	2010-05-18	3N2N
3NAH_1_dimer	1	6	2	1	0.061	2010-06-02	3QID
3NAH_1_same	1	6	2	1	0.061	2010-06-02	3NAH
3P83_1_same	2	6	6	1	0.112	2010-10-13	3P83
3Q46_1_same	1	6	3	1	0.887	2010-12-23	3Q46
3Q83_3_unbound	1	6	2	1	0.131	2011-01-06	3Q89
3QJG_1_same	1	12	2	1	0.227	2011-01-28	3QJG
3RHS_1_foreign	1	6	2	1	0.373	2011-04-12	3LCJ
3RHS_1_same	1	6	2	1	0.373	2011-04-12	3RHS
3RUV_1_foreign	1	16	2	1	0.226	2011-05-05	3RUS
3RUV_1_same	1	16	2	1	0.226	2011-05-05	3RUV
3S0C_1_foreign	1	10	3	1	0.333	2011-05-13	3S1U
3S1W_1_same	1	10	3	1	0.348	2011-05-16	3S1W
3SBA_1_dimer	1	6	2	1	0.086	2011-06-03	3SB9

Continued on next page

Continued from previous page

Assembly	Number of Protein types	Monomers	Binding modes	Monomer sources	Quality	Date of deposition	Monomer PDB sources
3SHH_2_foreign	1	8	3	1	0.410	2011-06-16	3TU2
3SHH_2_same	1	8	3	1	0.410	2011-06-16	3SHH
3SHX_1_same	1	24	3	1	0.578	2011-06-17	3SHX
3SSR_1_same	1	12	2	1	0.447	2011-07-08	3SSR
3T2B_1_foreign	1	8	3	1	0.470	2011-07-22	3T2F
3T2F_1_same	1	8	3	1	0.335	2011-07-22	3T2F
3T63_1_same	2	24	5	1	0.471	2011-07-28	3T63
3TJF_1_foreign	1	10	2	1	0.293	2011-08-24	3TJG
3UAV_1_foreign	1	6	2	1	0.550	2011-10-22	1XE3
3V60_1_same	2	6	2	1	0.136	2011-12-18	3V60
3V60_1_unbound	2	6	2	1	0.136	2011-12-18	3V61
3VAV_1_same	1	10	2	1	0.376	2011-12-29	3VAV
4ADS_1_same	2	24	4	1	-0.037	2012-01-03	4ADS
4AGE_1_same	1	7	1	1	-0.147	2012-01-26	4AGE
4AT1_1_foreign	2	12	5	2	0.184	1990-04-26	6AT1, 1TUG
4DU6_1_same	1	10	3	1	0.249	2012-02-21	4DU6
4EGG_1_same	1	12	2	1	0.214	2012-03-30	4EGG
4ELD_1_same	1	48	3	1	0.133	2012-04-10	4ELD
4FNK_1_foreign	2	6	5	1	0.337	2012-06-19	4FQY

Table C.1: Benchmark data set composition.

PDB	Description	Topology	Remark
1G41	Heat shock protein Hslu	One-layered ring	
1G6O	Traffic ATPase	One-layered ring	
1LJO	Archaeal Sm-like protein Af-Sm2	One-layered ring	
1NSF	N-Ethylmaleimide sensitive factor	One-layered ring	*foreign* in benchmark data set
1RRE	ATP-dependent protease	One-layered ring	
1BE4	Nucleoside diphosphate kinase isoform B	Dimer of trimers	
1EHW	Nucleoside diphosphate kinase	Dimer of trimers	
1EKR	Molybdenum cofactor biosynthesis protein C	Dimer of trimers	
1I40	Inorganic pyrophosphatase	Dimer of trimers	*foreign* in benchmark data set
1JX7	Hypothetical protein Ychn	Dimer of trimers	
1LCP	Leucine aminopeptidase	Dimer of trimers	
1PJH	Enoyl-CoA isomerase	Dimer of trimers	
1J2T	Creatinine amidohydrolase	Trimer of dimers	
1JE0	50-Methylthioadenosine phosphorylase	Trimer of dimers	
1NNG	Putative acyl-CoA thioester hydrolase	Trimer of dimers	superseded by 1YLI
1PJC	L-Alanine dehyrogenase	Trimer of dimers	
1R6L	Ribonuclease pH	Trimer of dimers	

Table C.2: Evaluation data set including annotation taken from *Comeau et al.* [193]. PDB entry 1NNG is superseded by 1YLI; 1NSF and 1I40 are part of the benchmark data set, however, using *foreign* monomer sources.

Assembly	Constraint Residue Pairs	Native	Assembly	Constraint Residue Pairs	Native
1HI9_1	(CA 81A, CA 179A)	y	1KW6_1	(CA 276A, CA 121A)	y
	(CA 270A, CA 82A)	y		(CA 131A, CA 129A)	y
	(CA 226A, CA 113A)	y		(CA 282A, CA 221A)	y
	(CA 129A, CA 225A)	n		(CA 103A, CA 175A)	n
	(CA 133A, CA 41A)	n		(CA 123A, CA 124A)	n
	(CA 147A, CA 189A)	n		(CA 154A, CA 114A)	n
	(CA 172A, CA 185A)	n		(CA 155A, CA 165A)	n
	(CA 184A, CA 186A)	n		(CA 16A, CA 224A)	n
	(CA 215A, CA 226A)	n		(CA 176A, CA 186A)	n
	(CA 222A, CA 33A)	n		(CA 258A, CA 246A)	n
	(CA 239A, CA 20A)	n		(CA 266A, CA 103A)	n
	(CA 45A, CA 25A)	n		(CA 78A, CA 33A)	n
	(CA 52A, CA 74A)	n		(CA 83A, CA 267A)	n
1PVV_1	(CA 310A, CA 313A)	y	1QK1_1	(CA 31A, CA 264A)	y
	(CA 57A, CA 85A)	y		(CA 206A, CA 57A)	y
	(CA 37A, CA 42A)	y		(CA 5A, CA 45A)	y
	(CA 168A, CA 222A)	n		(CA 113A, CA 336A)	n
	(CA 189A, CA 77A)	n		(CA 133A, CA 356A)	n
	(CA 193A, CA 18A)	n		(CA 163A, CA 323A)	n
	(CA 196A, CA 221A)	n		(CA 240A, CA 14A)	n
	(CA 1A, CA 6A)	n		(CA 255A, CA 165A)	n
	(CA 203A, CA 218A)	n		(CA 301A, CA 189A)	n
	(CA 211A, CA 273A)	n		(CA 324A, CA 2A)	n
	(CA 21A, CA 233A)	n		(CA 325A, CA 227A)	n
	(CA 248A, CA 155A)	n		(CA 347A, CA 205A)	n
	(CA 44A, CA 230A)	n		(CA 66A, CA 335A)	n
1X1O_1	(CA 143A, CA 17A)	y	1YNB_2	(CA 47A, CA 72A)	y
	(CA 28A, CA 13A)	y		(CA 135A, CA 31A)	y
	(CA 242A, CA 195A)	y		(CA 28A, CA 22A)	y
	(CA 125A, CA 160A)	n		(CA 102A, CA 173A)	n
	(CA 13A, CA 235A)	n		(CA 113A, CA 68A)	n
	(CA 204A, CA 256A)	n		(CA 114A, CA 48A)	n
	(CA 58A, CA 40A)	n		(CA 118A, CA 141A)	n
	(CA 62A, CA 210A)	n		(CA 143A, CA 88A)	n
	(CA 72A, CA 40A)	n		(CA 148A, CA 96A)	n
	(CA 74A, CA 158A)	n		(CA 150A, CA 12A)	n
	(CA 84A, CA 268A)	n		(CA 75A, CA 90A)	n
	(CA 89A, CA 252A)	n		(CA 76A, CA 11A)	n
	(CA 96A, CA 98A)	n		(CA 76A, CA 24A)	n
2BJK_1	(CA 461A, CA 512A)	y	2F1D_1	(CA 135A, CA 71A)	y
	(CA 225A, CA 453A)	y		(CA 100A, CA 103A)	y
	(CA 2A, CA 168A)	y		(CA 123A, CA 123A)	y
	(CA 207A, CA 173A)	n		(CA 129A, CA 145A)	n
	(CA 249A, CA 513A)	n		(CA 130A, CA 129A)	n
	(CA 349A, CA 12A)	n		(CA 140A, CA 94A)	n
	(CA 385A, CA 227A)	n		(CA 185A, CA 36A)	n
	(CA 387A, CA 276A)	n		(CA 30A, CA 157A)	n
	(CA 392A, CA 473A)	n		(CA 59A, CA 21A)	n
	(CA 406A, CA 405A)	n		(CA 61A, CA 96A)	n
	(CA 427A, CA 395A)	n		(CA 73A, CA 162A)	n
	(CA 440A, CA 516A)	n		(CA 78A, CA 132A)	n
	(CA 62A, CA 494A)	n		(CA 91A, CA 167A)	n
2UYU_1	(CA 251A, CA 186A)	y	3Q46_1	(CA 40A, CA 84A)	y
	(CA 110A, CA 47A)	y		(CA 135A, CA 47A)	y
	(CA 7A, CA 8A)	y		(CA 128A, CA 116A)	y
	(CA 126A, CA 257A)	n		(CA 14A, CA 103A)	n
	(CA 130A, CA 189A)	n		(CA 152A, CA 80A)	n
	(CA 132A, CA 132A)	n		(CA 167A, CA 26A)	n
	(CA 163A, CA 49A)	n		(CA 172A, CA 138A)	n
	(CA 186A, CA 182A)	n		(CA 177A, CA 47A)	n
	(CA 272A, CA 264A)	n		(CA 177A, CA 78A)	n
	(CA 36A, CA 16A)	n		(CA 56A, CA 84A)	n
	(CA 39A, CA 157A)	n		(CA 6A, CA 83A)	n
	(CA 83A, CA 167A)	n		(CA 76A, CA 127A)	n
	(CA 97A, CA 209A)	n		(CA 98A, CA 61A)	n

Table C.3: Randomly selected homo-multimers from the *same* data set including randomly chosen residue id pairs. The three topmost per structure correspond to natively interacting residue pairs, one per native binding mode. In addition, ten artifical, non-native interactions have been derived.

C.2 DOCKING RESULTS

This section contains additional information on the performance of docking runs carried out for the different assembly experiments.

Structure	Binding Mode A	B	C
1HI9	0.165 (124)	0.567 (68)	0.839 (68)
1KW6	1.138 (62)	0.351 (162)	0.792 (188)
1PVV	1.938 (38)	1.251 (56)	2.299 (6)
1QK1	1.053 (100)	0.670 (74)	0.491 (30)
1X1O	- (0)	2.280 (8)	0.615 (6)
1YNB	2.343 (4)	0.347 (96)	0.653 (58)
2BJK	- (0)	- (0)	0.202 (50)
2F1D	1.026 (104)	0.546 (104)	1.580 (108)
2UYU	- (0)	0.723 (34)	1.034 (274)
3Q46	0.167 (346)	0.461 (122)	0.195 (108)

(a) Native binding modes

Structure	Binding Mode A	B	C
1HI9	0.165 (124)	0.567 (68)	0.839 (70)
1KW6	1.138 (62)	0.351 (162)	0.792 (188)
1PVV	1.938 (38)	1.251 (58)	2.299 (6)
1QK1	1.053 (100)	0.670 (74)	0.491 (30)
1X1O	- (0)	2.280 (10)	0.615 (6)
1YNB	2.343 (6)	0.347 (96)	0.653 (58)
2BJK	- (0)	- (0)	0.202 (50)
2F1D	1.026 (104)	0.546 (104)	1.580 (110)
2UYU	- (0)	0.723 (34)	1.034 (274)
3Q46	0.167 (346)	0.461 (122)	0.195 (108)

(b) Native binding modes and three non-natives

Structure	Binding Mode A	B	C
1HI9	0.165 (122)	0.567 (68)	0.839 (70)
1KW6	1.138 (62)	0.351 (162)	0.792 (188)
1PVV	1.938 (38)	1.251 (58)	2.299 (6)
1QK1	1.053 (100)	0.670 (74)	0.491 (30)
1X1O	- (0)	2.280 (10)	0.615 (6)
1YNB	2.343 (6)	0.347 (96)	0.653 (58)
2BJK	- (0)	- (0)	0.202 (50)
2F1D	1.026 (104)	0.546 (104)	1.580 (110)
2UYU	- (0)	0.723 (34)	1.034 (274)
3Q46	0.167 (346)	0.461 (122)	0.195 (108)

(c) Native binding modes and six non-natives

Structure	Binding Mode A	B	C
1HI9	0.165 (122)	0.567 (68)	0.839 (68)
1KW6	1.138 (62)	0.351 (162)	0.792 (192)
1PVV	1.938 (38)	1.251 (58)	2.299 (6)
1QK1	1.053 (100)	0.670 (72)	0.491 (32)
1X1O	- (0)	2.280 (10)	0.615 (6)
1YNB	2.343 (6)	0.347 (96)	0.497 (64)
2BJK	- (0)	- (0)	0.202 (50)
2F1D	1.026 (104)	0.546 (104)	1.580 (108)
2UYU	- (0)	0.959 (32)	1.034 (274)
3Q46	0.167 (524)	0.461 (122)	0.195 (112)

(d) Native binding modes and ten non-natives

Table C.4: C_αRMSD of the best pose (and the number of poses) with a C_α RMSD $\leq$ 3.0Å for the four different constraint data sets. Differences (underlined) to the table for the three native binding modes (upper left) are introduced through the procedure of i) assignment of each pose to the (native or non-native) binding mode for which the lowest constraint penalty is achieved, and ii) subsequent clustering. Only minor differences are found, demonstrating the validity and robustness of the procedure.

Complex	Binding Mode	Number of poses	C_α RMSD (Å) Minimum	Median	Maximum
1B4F_1_unbound	A_C	1062	2.40477	12.60620	17.23500
1COA_1_same	A_G	232	2.64568	11.46795	14.77530
1FSF_1_unbound	A_B	4670	1.70606	20.02570	27.36400
1Ho5_1_foreign	A_C	3082	1.71168	16.20505	21.31170
1HKX_1_same	A_C	2722	2.20795	16.59135	25.16790
1I4o_1_foreign	A_B	3564	2.26239	17.38875	23.16630
1JH5_2_foreign	o_y	1102	2.29004	14.28135	23.09530
1MGQ_1_foreign	A_B	1662	2.64627	13.23880	19.13440
1NQT_3_same	A_E	1026	2.63105	26.46755	36.65520
1PVV_1_foreign	A_G	4006	1.71501	20.45280	29.01770
1PVV_1_same	A_G	3430	1.89115	20.47285	28.94420
1RYP_1_foreign	1_J	2158	2.63893	19.72550	27.67650
1RYP_1_foreign	K_Z	1165	2.19702	16.50270	23.39540
1RYP_1_same	D_L	2615	2.72419	20.47900	28.32150
1RYP_1_same	K_Z	1362	2.33735	16.97135	24.33850
1STM_1_same	o_H	898	2.78696	15.65315	27.68160
1SVT_1_foreign	O_P	1994	2.51698	16.56615	24.63360
1T6Q_1_foreign	A_E	480	2.78487	12.38515	20.01440
1T6Q_1_same	A_E	676	2.80217	13.45995	18.89670
1U6I_1_foreign	A_F	892	2.91429	18.63165	27.10630
1U6I_1_same	A_F	1154	2.87041	19.38770	27.46910
1WPB_1_same	A_T	1442	1.94007	21.79635	36.86480
1XXC_1_foreign	A_B	1444	2.62902	11.88930	16.26340
1XXC_1_foreign	A_D	394	2.11021	10.66495	14.54750
1XXC_1_same	A_B	1366	2.97289	11.74790	16.16110
1Y88_2_same	A_B	2858	1.96670	18.64985	28.81980
1YHU_1_same	A_M	2034	2.69661	17.37440	22.77990
1Z7Q_1_same	H_V	5090	2.17127	19.17450	25.66560
2BWE_1_same	A_B	702	1.99184	10.46940	14.49300
2CoW_1_same	o_p	58	1.99115	26.00655	36.38660
2F1D_1_same	A_I	1030	2.74020	16.48495	22.51160
2FZ6_5_same	B_H	84	2.99573	11.11770	12.39430
2G9T_1_dimer	A_C	1264	2.96804	15.11365	19.81720
2HDA_2_same	A_B	1064	1.91210	10.90855	15.42930
2HEY_1_foreign	A_C	3486	2.69606	15.63405	22.31770
2HEY_1_same	A_C	3666	2.05475	16.17690	22.73600
2HFN_1_foreign	A_B	1250	2.84099	14.17280	19.22650
2HFN_1_same	A_B	1550	2.90909	14.14595	19.30440
2HY5_1_foreign	A_C	1182	2.54804	14.93840	21.15330
2NUU_3_foreign	A_B	5198	2.87646	23.81670	31.21660
2V1W_1_same	A_B	152	2.65130	11.78095	16.78880
2VHX_1_foreign	A_B	806	1.38012	19.13030	28.29080
2VWS_1_foreign	A_B	2732	2.54116	20.42620	29.12030
2X6L_1_same	C_D	822	1.78907	13.24215	18.40400

Continued on next page

Continued from previous page

Complex	Binding Mode	Number of poses	C_α RMSD (Å) Minimum	Median	Maximum
3AQD_1_foreign	A_B	670	2.42805	12.71255	18.58670
3AQD_1_same	A_B	1282	1.42724	12.86745	17.68650
3D5O_1_foreign	B_F	1726	2.85327	19.50855	27.10090
3D5O_1_same	B_F	1535	2.49097	19.70290	27.10290
3EJ9_1_same	A_C	1222	2.46750	15.20800	21.88210
3EVO_1_same	A_C	1682	1.91074	15.86605	21.46160
3GE3_1_foreign	A_C	1773	1.86621	17.11900	30.88050
3GE3_1_same	A_C	2246	2.83510	17.35080	32.87590
3KA3_1_foreign	A_G	2260	0.88395	17.31155	28.96070
3LWZ_1_same	A_D	1138	2.27340	15.92850	20.36630
3M4B_3_same	A_E	2618	2.73586	15.28175	22.93070
3M4B_3_same	A_G	854	1.18737	12.77455	17.58670
3P83_1_same	A_D	395	2.93880	17.95700	25.09990
3P83_1_same	E_F	2278	2.64979	20.69110	33.19940
3RHS_1_same	A_B	4474	1.83169	17.67305	23.95900
3SHX_1_same	A_G	2128	1.38985	18.76985	29.37180
4AT1_1_foreign	A_E	3292	2.86391	22.87240	30.83220

Table C.5: The 61 cases for which CombDock could reproduce at least one docking pose with a C_α RMSD $\leq$ 3.0Å. The native binding mode is specified by the pair of chains in the corresponding reference complex.

C.3 ASSEMBLY PARAMETERS AND RESULTS

This section comprises the full parameter sets and the results obtained from running 3D-MOSAIC with the corresponding configuration on the respective data set, i.e., the benchmark data set, the evaluation set (Comeau), the SRPIC experiments, or the CombDock-generated dockings. Pairs of tables for parameter sets and respective 3D-MOSAIC results are arranged side-by-side on a double page for easy cross-checking.

The parameters varied throughout the experiments, sorted by categories, are listed in Tab. C.6.

Parameters in category	**Scope**
Clustering	
C_α pre-cluster RMSD	$\mathbb{R}_0^+$
C_α intra-cluster RMSD	$\mathbb{R}_0^+$
C_α post-cluster RMSD	$\mathbb{R}_0^+$
Clash checking	
Max number of clashes per dimer	$\mathbb{N}$
Number of solutions to consider	
Number of solutions to keep in first iteration	$\mathbb{N}^+$
Number of solutions to keep at least per iteration	$\mathbb{N}^+$
Solutions reduction rate	$\mathbb{R}^+ \setminus [0,1[$
Transformation matching	
Transformation Match Score	S^{da} / S^{rmsd}
Max transformation angle	$\mathbb{R}_0^+$
Max transformation displacement	$\mathbb{R}_0^+$
Max transformation RMSD	$\mathbb{R}_0^+$
Discard non-matching solutions	true/false (y/n)
Enable transformation matching	true/false (y/n)
Limitation of solution/search space	
Number of best poses per interface to keep	$\mathbb{N}$
Symmetric pose fraction per interface	$\mathbb{R}_0^+$
Symmetric pose RMSD	$\mathbb{R}_0^+$
Keep symmetric interface poses only	true/false (y/n)
Distinct interfaces	true/false (y/n)
Others	
Interpolate ligand transformation	true/false (y/n)
Max complex size for pre-ranking by number of symmetry mappings	$\mathbb{N}$

Table C.6: List of parameters varied within and across different experiments and their respective categories and scopes.

	Clustering			Allowed	Transformation Matching		Ligand
Code	Pre	Intra	Post	Clashes	Displacement	Angle	Interpolation
ben_01	1.0	2.0	3.0	10	1.0	5.0	n
ben_02	1.0	2.0	3.0	25	1.0	5.0	n
ben_03[3]	1.0	2.0	3.0	50	1.0	5.0	n
ben_04	1.0	2.0	3.0	150	1.0	5.0	n
ben_05	1.0	2.0	3.0	10	1.5	10.0	n
ben_06	1.0	2.0	3.0	25	1.5	10.0	n
ben_07	1.0	2.0	3.0	50	1.5	10.0	n
ben_08	1.0	2.0	3.0	150	1.5	10.0	n
ben_09	1.0	2.0	3.0	10	2.5	15.0	n
ben_10	1.0	2.0	3.0	25	2.5	15.0	n
ben_11	1.0	2.0	3.0	50	2.5	15.0	n
ben_12	1.0	2.0	3.0	150	2.5	15.0	n
ben_13	1.0	3.0	5.0	10	1.0	5.0	n
ben_14	1.0	3.0	5.0	25	1.0	5.0	n
ben_15	1.0	3.0	5.0	50	1.0	5.0	n
ben_16	1.0	3.0	5.0	150	1.0	5.0	n
ben_17	1.0	3.0	5.0	10	1.5	10.0	n
ben_18	1.0	3.0	5.0	25	1.5	10.0	n
ben_19	1.0	3.0	5.0	50	1.5	10.0	n
ben_20	1.0	3.0	5.0	150	1.5	10.0	n
ben_21	1.0	3.0	5.0	10	2.5	15.0	n
ben_22	1.0	3.0	5.0	25	2.5	15.0	n
ben_23	1.0	3.0	5.0	50	2.5	15.0	n
ben_24[2,3]	1.0	3.0	5.0	150	2.5	15.0	n
ben_25	1.0	2.0	3.0	10	1.0	5.0	y
ben_26	1.0	2.0	3.0	25	1.0	5.0	y
ben_27[2]	1.0	2.0	3.0	50	1.0	5.0	y
ben_28	1.0	2.0	3.0	150	1.0	5.0	y
ben_29	1.0	2.0	3.0	10	1.5	10.0	y
ben_30	1.0	2.0	3.0	25	1.5	10.0	y
ben_31	1.0	2.0	3.0	50	1.5	10.0	y
ben_32[3]	1.0	2.0	3.0	150	1.5	10.0	y
ben_33	1.0	2.0	3.0	10	2.5	15.0	y
ben_34	1.0	2.0	3.0	25	2.5	15.0	y
ben_35	1.0	2.0	3.0	50	2.5	15.0	y
ben_36	1.0	2.0	3.0	150	2.5	15.0	y
ben_37	1.0	3.0	5.0	10	1.0	5.0	y
ben_38	1.0	3.0	5.0	25	1.0	5.0	y
ben_39	1.0	3.0	5.0	50	1.0	5.0	y
ben_40	1.0	3.0	5.0	150	1.0	5.0	y
ben_41	1.0	3.0	5.0	10	1.5	10.0	y
ben_42	1.0	3.0	5.0	25	1.5	10.0	y
ben_43	1.0	3.0	5.0	50	1.5	10.0	y
ben_44	1.0	3.0	5.0	150	1.5	10.0	y
ben_45	1.0	3.0	5.0	10	2.5	15.0	y
ben_46	1.0	3.0	5.0	25	2.5	15.0	y
ben_47	1.0	3.0	5.0	50	2.5	15.0	y
ben_48[1]	1.0	3.0	5.0	150	2.5	15.0	y

Table C.7: Parameters for benchmark runs with S^{da} transformation matching. Superscript numbers indicate the best-performing single (1), pair (2), and triple (3) of parameter sets.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	≤ 10		≤ 5		≤ 3		≤ 2.5		≤ 2.0		≤ 1.0	
ben_01	307	170	(4.21)	94	(2.99)	47	(6.85)	164	(5.40)	137	(5.53)	57	(8.05)
ben_02	306	192	(2.96)	136	(1.88)	95	(3.39)	190	(3.82)	171	(2.63)	98	(4.48)
ben_03	307	195	(1.90)	157	(1.98)	126	(3.61)	198	(3.65)	177	(2.05)	129	(3.68)
ben_04	308	208	(1.81)	172	(2.22)	139	(3.26)	208	(2.32)	192	(2.29)	141	(3.37)
ben_05	306	182	(3.91)	124	(2.66)	72	(4.86)	180	(4.13)	161	(4.72)	79	(5.23)
ben_06	308	198	(2.21)	144	(1.65)	116	(3.76)	196	(2.69)	177	(2.46)	121	(6.26)
ben_07	308	209	(2.15)	168	(1.60)	139	(3.37)	216	(3.29)	199	(2.45)	143	(4.55)
ben_08	307	211	(1.82)	179	(1.74)	152	(4.01)	212	(2.38)	202	(2.19)	154	(3.57)
ben_09	306	188	(3.30)	129	(2.09)	77	(4.48)	188	(3.59)	164	(3.61)	91	(7.60)
ben_10	306	206	(2.26)	161	(2.09)	121	(2.87)	208	(2.83)	190	(2.14)	131	(4.18)
ben_11	306	204	(1.92)	171	(2.04)	143	(3.42)	214	(3.87)	193	(2.88)	152	(4.64)
ben_12	306	212	(1.73)	187	(2.00)	154	(3.19)	217	(2.31)	206	(2.15)	165	(3.75)
ben_13	305	168	(5.32)	77	(2.19)	19	(1.42)	160	(5.78)	130	(4.88)	41	(2.46)
ben_14	307	189	(2.20)	123	(2.28)	74	(2.54)	183	(3.55)	162	(3.11)	77	(3.42)
ben_15	308	201	(1.87)	156	(1.63)	101	(2.19)	197	(1.96)	184	(2.14)	107	(2.22)
ben_16	308	202	(2.11)	163	(1.46)	122	(2.66)	201	(2.15)	190	(1.82)	125	(3.07)
ben_17	307	186	(4.20)	125	(3.82)	39	(1.82)	173	(4.23)	156	(4.67)	59	(3.20)
ben_18	308	202	(2.26)	147	(2.03)	97	(2.34)	204	(2.79)	186	(2.80)	100	(3.97)
ben_19	308	208	(1.50)	174	(1.60)	128	(2.96)	214	(2.00)	204	(1.97)	126	(4.15)
ben_20	307	210	(1.43)	179	(1.41)	144	(2.61)	216	(1.73)	206	(1.84)	146	(4.51)
ben_21	305	187	(3.69)	124	(3.68)	41	(1.73)	181	(5.04)	163	(5.21)	61	(3.67)
ben_22	307	201	(2.25)	152	(2.57)	105	(3.11)	207	(2.62)	184	(2.65)	107	(3.91)
ben_23	306	210	(1.88)	171	(1.47)	133	(2.69)	214	(1.87)	200	(1.62)	135	(3.56)
ben_24	307	214	(1.46)	182	(1.68)	147	(2.82)	218	(2.18)	206	(1.87)	155	(3.02)
ben_25	305	173	(4.61)	100	(2.42)	50	(6.06)	158	(5.28)	140	(4.91)	60	(5.73)
ben_26	306	192	(2.70)	137	(1.68)	95	(2.76)	194	(3.15)	169	(2.21)	96	(5.56)
ben_27	307	198	(1.68)	155	(1.58)	120	(3.36)	199	(3.42)	180	(2.08)	125	(4.38)
ben_28	308	209	(1.77)	176	(1.91)	133	(2.34)	212	(2.30)	198	(2.17)	137	(2.87)
ben_29	307	188	(4.47)	130	(3.27)	72	(4.43)	188	(4.64)	166	(4.13)	83	(6.33)
ben_30	308	197	(2.19)	145	(1.79)	117	(3.00)	197	(3.09)	182	(2.81)	126	(4.73)
ben_31	308	205	(1.98)	168	(2.55)	136	(2.23)	213	(4.15)	199	(2.50)	144	(3.87)
ben_32	308	214	(1.96)	184	(2.65)	151	(3.17)	220	(2.94)	208	(2.53)	157	(3.92)
ben_33	306	193	(3.02)	136	(2.74)	82	(3.83)	193	(3.90)	171	(3.55)	91	(6.16)
ben_34	306	202	(2.17)	156	(2.88)	121	(2.61)	209	(3.81)	187	(2.82)	131	(4.50)
ben_35	306	207	(2.01)	173	(2.28)	142	(2.41)	217	(4.75)	198	(2.12)	152	(2.93)
ben_36	306	215	(1.75)	187	(2.12)	153	(3.70)	219	(2.57)	209	(1.89)	165	(3.45)
ben_37	307	168	(5.12)	77	(2.40)	27	(2.78)	154	(5.70)	123	(5.09)	40	(2.90)
ben_38	308	182	(2.08)	121	(2.02)	71	(2.90)	179	(2.78)	160	(2.71)	76	(3.54)
ben_39	308	189	(1.47)	147	(1.65)	98	(3.20)	186	(1.82)	176	(2.26)	100	(2.45)
ben_40	308	202	(1.65)	163	(1.72)	119	(2.82)	205	(1.85)	193	(1.93)	117	(2.68)
ben_41	308	191	(3.92)	117	(3.24)	33	(2.18)	185	(5.37)	156	(4.80)	53	(4.47)
ben_42	308	203	(3.65)	148	(1.97)	96	(2.00)	202	(3.30)	188	(3.41)	106	(3.48)
ben_43	308	208	(1.68)	170	(2.04)	130	(2.55)	213	(2.29)	198	(1.73)	126	(3.59)
ben_44	308	207	(1.35)	171	(1.64)	140	(2.30)	214	(2.05)	202	(1.76)	140	(2.58)
ben_45	306	193	(4.47)	128	(3.63)	40	(1.38)	187	(4.72)	169	(5.14)	62	(4.02)
ben_46	305	207	(2.52)	151	(1.85)	98	(3.60)	210	(3.46)	192	(4.06)	106	(3.52)
ben_47	307	206	(1.72)	172	(1.82)	127	(2.03)	214	(2.19)	200	(2.08)	130	(2.28)
ben_48	307	215	(1.56)	185	(2.11)	151	(3.00)	223	(1.93)	209	(2.28)	154	(2.66)
div2	308	241	(13.90)	207	(14.48)	170	(18.04)	250	(4.35)	232	(2.34)	180	(5.79)
div3	308	249	(18.75)	218	(17.89)	185	(22.95)	258	(5.90)	244	(3.58)	188	(6.11)
all	308	262	(265.49)	235	(255.52)	204	(303.07)	272	(47.09)	265	(48.74)	216	(48.25)

Table C.8: Performance of benchmark models for the S^{da} transformation match score. First (second) half corresponds to disabled (enabled) ligand interpolation.

	Clustering			Allowed	Transformation Matching		Ligand
Code	Pre	Intra	Post	Clashes	Displacement	RMSD	Interpolation
ben_49	1.0	2.0	3.0	10	1.0	3.0	n
ben_50[3]	1.0	2.0	3.0	25	1.0	3.0	n
ben_51[2]	1.0	2.0	3.0	50	1.0	3.0	n
ben_52	1.0	2.0	3.0	150	1.0	3.0	n
ben_53	1.0	2.0	3.0	10	1.5	4.5	n
ben_54	1.0	2.0	3.0	25	1.5	4.5	n
ben_55	1.0	2.0	3.0	50	1.5	4.5	n
ben_56	1.0	2.0	3.0	150	1.5	4.5	n
ben_57	1.0	2.0	3.0	10	2.5	7.5	n
ben_58	1.0	2.0	3.0	25	2.5	7.5	n
ben_59	1.0	2.0	3.0	50	2.5	7.5	n
ben_60	1.0	2.0	3.0	150	2.5	7.5	n
ben_61	1.0	3.0	5.0	10	1.0	3.0	n
ben_62	1.0	3.0	5.0	25	1.0	3.0	n
ben_63	1.0	3.0	5.0	50	1.0	3.0	n
ben_64[3]	1.0	3.0	5.0	150	1.0	3.0	n
ben_65	1.0	3.0	5.0	10	1.5	4.5	n
ben_66	1.0	3.0	5.0	25	1.5	4.5	n
ben_67	1.0	3.0	5.0	50	1.5	4.5	n
ben_68	1.0	3.0	5.0	150	1.5	4.5	n
ben_69	1.0	3.0	5.0	10	2.5	7.5	n
ben_70	1.0	3.0	5.0	25	2.5	7.5	n
ben_71	1.0	3.0	5.0	50	2.5	7.5	n
ben_72[1,2,3]	1.0	3.0	5.0	150	2.5	7.5	n
ben_73	1.0	2.0	3.0	10	1.0	3.0	y
ben_74	1.0	2.0	3.0	25	1.0	3.0	y
ben_75	1.0	2.0	3.0	50	1.0	3.0	y
ben_76	1.0	2.0	3.0	150	1.0	3.0	y
ben_77	1.0	2.0	3.0	10	1.5	4.5	y
ben_78	1.0	2.0	3.0	25	1.5	4.5	y
ben_79	1.0	2.0	3.0	50	1.5	4.5	y
ben_80	1.0	2.0	3.0	150	1.5	4.5	y
ben_81	1.0	2.0	3.0	10	2.5	7.5	y
ben_82	1.0	2.0	3.0	25	2.5	7.5	y
ben_83	1.0	2.0	3.0	50	2.5	7.5	y
ben_84	1.0	2.0	3.0	150	2.5	7.5	y
ben_85	1.0	3.0	5.0	10	1.0	3.0	y
ben_86	1.0	3.0	5.0	25	1.0	3.0	y
ben_87	1.0	3.0	5.0	50	1.0	3.0	y
ben_88	1.0	3.0	5.0	150	1.0	3.0	y
ben_89	1.0	3.0	5.0	10	1.5	4.5	y
ben_90	1.0	3.0	5.0	25	1.5	4.5	y
ben_91	1.0	3.0	5.0	50	1.5	4.5	y
ben_92	1.0	3.0	5.0	150	1.5	4.5	y
ben_93	1.0	3.0	5.0	10	2.5	7.5	y
ben_94	1.0	3.0	5.0	25	2.5	7.5	y
ben_95	1.0	3.0	5.0	50	2.5	7.5	y
ben_96	1.0	3.0	5.0	150	2.5	7.5	y

Table C.9: Parameters for benchmark runs with S^{rmsd} transformation matching. Superscript numbers indicate the best-performing single (1), pair (2), and triple (3) of parameter sets.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	≤ 10		≤ 5		≤ 3		≤ 2.5		≤ 2.0		≤ 1.0	
ben_49	306	189	(4.33)	118	(2.97)	70	(4.09)	180	(4.26)	161	(5.26)	82	(7.89)
ben_50	306	196	(2.49)	150	(2.49)	115	(3.71)	198	(3.01)	183	(2.55)	120	(5.70)
ben_51	306	200	(2.01)	169	(2.30)	138	(2.90)	207	(3.98)	193	(2.60)	144	(4.22)
ben_52	308	202	(1.58)	174	(1.75)	152	(3.47)	209	(2.73)	197	(2.03)	156	(4.61)
ben_53	303	180	(3.26)	132	(3.05)	79	(4.13)	180	(3.86)	162	(4.06)	93	(8.31)
ben_54	307	191	(2.00)	153	(1.69)	120	(2.50)	198	(3.16)	183	(2.40)	122	(4.17)
ben_55	307	204	(2.07)	171	(1.65)	143	(2.94)	208	(3.64)	195	(2.24)	154	(3.40)
ben_56	307	208	(1.63)	181	(1.87)	156	(4.17)	215	(2.72)	203	(2.63)	162	(3.49)
ben_57	305	190	(3.62)	132	(2.72)	75	(3.84)	192	(4.44)	170	(4.23)	93	(6.15)
ben_58	304	200	(2.12)	160	(2.08)	124	(3.07)	203	(2.72)	192	(2.83)	129	(3.98)
ben_59	305	205	(2.08)	176	(1.51)	144	(3.03)	211	(3.70)	196	(1.95)	153	(3.33)
ben_60	305	208	(1.45)	185	(1.64)	155	(3.83)	216	(3.12)	203	(2.00)	164	(3.27)
ben_61	305	183	(4.76)	109	(3.37)	37	(3.24)	174	(4.95)	150	(5.09)	54	(3.24)
ben_62	307	199	(2.55)	150	(2.11)	90	(4.47)	203	(2.80)	180	(2.79)	97	(3.89)
ben_63	308	208	(1.65)	168	(1.64)	124	(3.61)	215	(3.41)	198	(2.36)	131	(4.27)
ben_64	308	214	(1.55)	183	(1.37)	147	(2.67)	216	(1.70)	206	(1.96)	148	(2.80)
ben_65	305	187	(3.21)	120	(2.78)	47	(2.30)	184	(4.84)	159	(4.77)	67	(5.00)
ben_66	305	200	(2.60)	152	(1.95)	101	(2.55)	198	(2.41)	183	(2.58)	99	(2.86)
ben_67	305	201	(1.71)	169	(1.54)	132	(3.44)	207	(2.00)	192	(2.03)	133	(4.07)
ben_68	305	213	(1.61)	177	(1.65)	147	(2.61)	215	(2.56)	201	(1.71)	147	(2.73)
ben_69	304	194	(3.13)	132	(2.30)	50	(2.10)	190	(4.13)	171	(3.53)	68	(4.19)
ben_70	305	202	(2.19)	157	(2.00)	106	(3.07)	208	(2.92)	190	(2.83)	110	(3.48)
ben_71	305	209	(1.77)	175	(1.58)	135	(3.36)	214	(2.42)	193	(1.65)	143	(3.32)
ben_72	305	211	(1.66)	178	(1.63)	146	(3.42)	221	(2.96)	207	(2.39)	151	(2.62)
ben_73	307	185	(4.02)	119	(4.18)	70	(4.97)	180	(3.94)	163	(5.20)	83	(7.43)
ben_74	307	198	(2.17)	150	(2.12)	118	(3.33)	203	(2.82)	186	(2.74)	121	(3.79)
ben_75	306	204	(1.83)	169	(1.66)	139	(3.20)	207	(3.15)	194	(1.82)	145	(3.63)
ben_76	306	209	(1.51)	178	(1.76)	147	(3.65)	209	(2.19)	201	(1.85)	151	(3.26)
ben_77	307	184	(4.25)	137	(2.52)	80	(4.03)	184	(4.34)	165	(4.04)	94	(7.05)
ben_78	306	199	(2.71)	156	(2.36)	120	(2.74)	206	(3.68)	190	(3.16)	125	(4.87)
ben_79	307	206	(2.01)	172	(2.27)	138	(2.64)	212	(3.51)	199	(2.51)	149	(3.78)
ben_80	307	212	(1.58)	184	(2.15)	150	(3.09)	217	(2.72)	206	(2.26)	165	(3.07)
ben_81	303	191	(3.27)	134	(2.51)	75	(3.81)	196	(5.19)	175	(3.75)	93	(4.99)
ben_82	306	199	(2.23)	161	(2.01)	125	(2.60)	203	(2.98)	188	(2.96)	134	(3.48)
ben_83	306	204	(2.08)	169	(2.18)	141	(3.67)	209	(3.98)	196	(2.20)	151	(3.30)
ben_84	304	207	(1.56)	178	(1.57)	152	(4.09)	208	(2.54)	196	(1.67)	164	(3.87)
ben_85	305	181	(4.63)	108	(2.56)	40	(2.40)	172	(4.40)	145	(4.34)	50	(3.36)
ben_86	306	193	(2.82)	136	(1.97)	90	(2.79)	193	(2.77)	173	(2.97)	89	(2.96)
ben_87	306	204	(1.90)	166	(1.84)	114	(2.65)	205	(1.90)	194	(1.93)	125	(4.10)
ben_88	308	210	(1.64)	176	(1.76)	141	(2.71)	213	(2.19)	203	(1.98)	148	(4.16)
ben_89	306	191	(3.96)	125	(3.73)	36	(2.69)	181	(4.75)	160	(4.19)	60	(4.53)
ben_90	303	200	(2.48)	146	(2.10)	100	(3.47)	197	(2.78)	181	(3.18)	109	(5.37)
ben_91	305	204	(1.86)	169	(1.62)	131	(2.17)	207	(2.12)	196	(2.45)	136	(3.31)
ben_92	306	216	(1.63)	183	(2.09)	150	(2.76)	218	(2.30)	205	(1.90)	153	(2.70)
ben_93	304	196	(2.91)	132	(2.56)	43	(1.56)	189	(3.31)	169	(3.51)	72	(5.15)
ben_94	305	203	(2.96)	155	(2.27)	104	(2.38)	214	(3.48)	188	(3.11)	113	(3.12)
ben_95	305	211	(2.07)	170	(1.70)	136	(2.72)	215	(2.12)	201	(2.01)	143	(3.09)
ben_96	305	215	(1.73)	182	(2.25)	146	(3.36)	220	(2.63)	208	(2.45)	154	(2.97)
div2	308	232	(8.84)	199	(9.85)	164	(11.05)	245	(4.34)	228	(3.15)	173	(5.95)
div3	308	238	(13.48)	205	(13.94)	170	(15.50)	252	(4.58)	234	(4.06)	181	(8.35)
all	308	259	(177.22)	229	(160.47)	199	(205.01)	267	(59.88)	256	(36.46)	217	(41.51)

Table C.10: Performance of benchmark models for the S^{rmsd} transformation match score. First (second) half corresponds to disabled (enabled) ligand interpolation.

	Clustering			Allowed	Transformation Matching		Ligand
Code	Pre	Intra	Post	Clashes	Displacement	Angle	Interpolation
sc_01	1.0	0.0	1.0	10	1.0	5.0	n
sc_02	1.0	0.0	1.0	25	1.0	5.0	n
sc_03	1.0	0.0	1.0	50	1.0	5.0	n
sc_04	1.0	0.0	1.0	150	1.0	5.0	n
sc_05	1.0	0.0	1.0	10	1.5	10.0	n
sc_06	1.0	0.0	1.0	25	1.5	10.0	n
sc_07	1.0	0.0	1.0	50	1.5	10.0	n
sc_08	1.0	0.0	1.0	150	1.5	10.0	n
sc_09	1.0	0.0	1.0	10	2.5	15.0	n
sc_10	1.0	0.0	1.0	25	2.5	15.0	n
sc_11	1.0	0.0	1.0	50	2.5	15.0	n
sc_12	1.0	0.0	1.0	150	2.5	15.0	n
sc_13	1.0	1.0	2.0	10	1.0	5.0	n
sc_14	1.0	1.0	2.0	25	1.0	5.0	n
sc_15	1.0	1.0	2.0	50	1.0	5.0	n
sc_16	1.0	1.0	2.0	150	1.0	5.0	n
sc_17	1.0	1.0	2.0	10	1.5	10.0	n
sc_18	1.0	1.0	2.0	25	1.5	10.0	n
sc_19	1.0	1.0	2.0	50	1.5	10.0	n
sc_20	1.0	1.0	2.0	150	1.5	10.0	n
sc_21	1.0	1.0	2.0	10	2.5	15.0	n
sc_22	1.0	1.0	2.0	25	2.5	15.0	n
sc_23	1.0	1.0	2.0	50	2.5	15.0	n
sc_24	1.0	1.0	2.0	150	2.5	15.0	n
sc_25	1.0	0.0	1.0	10	1.0	5.0	y
sc_26	1.0	0.0	1.0	25	1.0	5.0	y
sc_27	1.0	0.0	1.0	50	1.0	5.0	y
sc_28	1.0	0.0	1.0	150	1.0	5.0	y
sc_29	1.0	0.0	1.0	10	1.5	10.0	y
sc_30	1.0	0.0	1.0	25	1.5	10.0	y
sc_31	1.0	0.0	1.0	50	1.5	10.0	y
sc_32	1.0	0.0	1.0	150	1.5	10.0	y
sc_33	1.0	0.0	1.0	10	2.5	15.0	y
sc_34	1.0	0.0	1.0	25	2.5	15.0	y
sc_35	1.0	0.0	1.0	50	2.5	15.0	y
sc_36	1.0	0.0	1.0	150	2.5	15.0	y
sc_37	1.0	1.0	2.0	10	1.0	5.0	y
sc_38	1.0	1.0	2.0	25	1.0	5.0	y
sc_39	1.0	1.0	2.0	50	1.0	5.0	y
sc_40	1.0	1.0	2.0	150	1.0	5.0	y
sc_41	1.0	1.0	2.0	10	1.5	10.0	y
sc_42	1.0	1.0	2.0	25	1.5	10.0	y
sc_43	1.0	1.0	2.0	50	1.5	10.0	y
sc_44	1.0	1.0	2.0	150	1.5	10.0	y
sc_45	1.0	1.0	2.0	10	2.5	15.0	y
sc_46	1.0	1.0	2.0	25	2.5	15.0	y
sc_47	1.0	1.0	2.0	50	2.5	15.0	y
sc_48	1.0	1.0	2.0	150	2.5	15.0	y

Table C.11: Parameters for runs with S^{da} transformation matching and small clustering parameters.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	≤ 10		≤ 5		≤ 3		≤ 2.5		≤ 2.0		≤ 1.0	
sc_01	303	147	(2.58)	90	(2.74)	49	(7.06)	149	(3.74)	126	(3.18)	70	(6.09)
sc_02	304	163	(2.27)	123	(2.33)	87	(3.41)	167	(2.34)	148	(3.24)	98	(4.07)
sc_03	307	171	(1.67)	135	(2.33)	101	(3.60)	173	(2.38)	159	(2.19)	115	(3.63)
sc_04	307	180	(1.90)	145	(1.65)	121	(3.60)	183	(1.83)	167	(2.01)	128	(4.27)
sc_05	306	158	(3.93)	102	(2.89)	59	(6.90)	159	(5.16)	134	(3.17)	65	(6.02)
sc_06	306	170	(1.74)	132	(2.46)	96	(5.08)	176	(2.00)	158	(3.16)	110	(5.53)
sc_07	307	182	(1.79)	144	(1.66)	116	(3.67)	181	(2.23)	169	(2.40)	127	(4.56)
sc_08	307	182	(1.91)	152	(2.14)	126	(5.17)	186	(2.15)	173	(2.71)	129	(3.68)
sc_09	302	155	(3.05)	106	(2.98)	59	(5.29)	162	(3.32)	137	(2.87)	76	(8.28)
sc_10	305	167	(2.20)	125	(2.66)	93	(4.77)	175	(2.61)	153	(3.30)	104	(3.44)
sc_11	307	179	(2.04)	143	(2.75)	113	(4.31)	187	(2.68)	170	(2.07)	120	(3.57)
sc_12	307	180	(2.21)	154	(2.71)	121	(4.25)	193	(3.42)	176	(2.19)	131	(3.76)
sc_13	307	170	(5.85)	104	(3.15)	61	(6.61)	158	(4.04)	139	(5.40)	75	(6.68)
sc_14	307	188	(4.14)	139	(2.27)	107	(4.13)	183	(1.90)	171	(2.92)	111	(4.06)
sc_15	307	195	(2.68)	155	(2.33)	126	(2.68)	190	(2.18)	181	(2.50)	135	(3.18)
sc_16	308	203	(1.84)	167	(1.94)	138	(3.93)	201	(1.66)	190	(1.89)	145	(3.57)
sc_17	306	181	(4.33)	129	(2.45)	86	(5.58)	178	(3.69)	163	(5.01)	96	(10.06)
sc_18	308	193	(2.61)	150	(2.08)	124	(4.79)	189	(1.75)	182	(3.08)	129	(4.98)
sc_19	308	201	(1.88)	164	(1.71)	142	(3.94)	203	(2.03)	191	(2.43)	149	(3.56)
sc_20	308	204	(1.92)	175	(2.05)	152	(4.27)	204	(2.80)	194	(2.34)	159	(4.43)
sc_21	306	186	(3.95)	128	(1.85)	92	(5.45)	186	(4.38)	166	(5.04)	99	(8.58)
sc_22	308	193	(3.19)	151	(3.85)	120	(4.49)	200	(2.93)	182	(3.59)	125	(5.58)
sc_23	308	204	(1.80)	170	(2.29)	141	(4.40)	213	(1.98)	195	(1.89)	147	(3.84)
sc_24	308	208	(1.74)	182	(2.24)	150	(3.88)	214	(2.29)	199	(2.02)	161	(4.00)
sc_25	303	147	(3.54)	88	(3.47)	46	(4.54)	145	(3.10)	126	(4.72)	65	(5.80)
sc_26	306	163	(2.54)	121	(2.24)	80	(2.89)	168	(2.11)	150	(2.86)	91	(3.00)
sc_27	306	168	(1.26)	132	(2.08)	98	(2.86)	170	(1.45)	154	(2.31)	107	(3.93)
sc_28	307	177	(2.38)	144	(1.65)	107	(3.63)	183	(2.08)	167	(2.00)	115	(4.10)
sc_29	305	157	(2.73)	100	(2.55)	56	(7.34)	157	(3.92)	132	(2.91)	66	(6.32)
sc_30	307	165	(2.05)	131	(2.15)	96	(5.08)	171	(1.67)	157	(3.48)	107	(4.21)
sc_31	305	176	(1.51)	146	(1.81)	118	(3.62)	180	(2.28)	169	(2.49)	125	(3.38)
sc_32	307	181	(2.13)	152	(1.69)	129	(4.61)	183	(2.01)	173	(2.38)	132	(3.96)
sc_33	305	159	(2.74)	108	(2.91)	57	(5.98)	164	(3.68)	141	(2.47)	74	(6.46)
sc_34	306	163	(2.64)	128	(2.87)	93	(4.20)	166	(2.52)	149	(3.09)	101	(3.17)
sc_35	307	172	(1.81)	138	(2.36)	110	(3.21)	175	(2.69)	164	(2.51)	117	(4.61)
sc_36	305	172	(2.36)	144	(1.47)	119	(4.23)	182	(2.60)	168	(2.04)	121	(3.86)
sc_37	304	173	(4.83)	104	(2.61)	61	(5.89)	162	(4.13)	140	(4.85)	74	(6.47)
sc_38	306	191	(3.73)	141	(2.31)	103	(5.26)	186	(2.15)	171	(2.94)	107	(4.86)
sc_39	306	198	(2.43)	159	(2.95)	128	(3.41)	197	(2.15)	183	(3.09)	133	(3.94)
sc_40	308	202	(1.96)	170	(2.22)	138	(2.79)	201	(2.13)	189	(2.30)	149	(3.90)
sc_41	307	179	(4.32)	127	(2.84)	89	(5.17)	177	(3.94)	162	(5.23)	95	(8.03)
sc_42	308	196	(2.64)	156	(2.10)	127	(4.50)	195	(1.63)	187	(2.76)	137	(4.39)
sc_43	308	206	(2.04)	173	(2.07)	148	(3.19)	211	(2.69)	195	(2.45)	157	(3.63)
sc_44	308	207	(1.61)	180	(2.21)	153	(3.72)	213	(2.36)	200	(2.71)	164	(3.68)
sc_45	305	184	(3.58)	128	(2.39)	89	(6.24)	183	(3.71)	161	(4.27)	99	(7.18)
sc_46	307	197	(2.71)	153	(2.80)	126	(3.64)	193	(1.68)	182	(2.86)	135	(3.33)
sc_47	307	203	(2.23)	173	(3.18)	143	(3.03)	211	(2.46)	194	(2.32)	153	(3.04)
sc_48	308	211	(1.75)	183	(2.30)	153	(4.05)	216	(1.64)	203	(1.76)	163	(4.09)
div2	308	229	(12.72)	200	(12.68)	172	(15.20)	235	(1.78)	223	(2.47)	184	(5.57)
div3	308	234	(23.25)	202	(22.47)	170	(24.29)	242	(5.50)	228	(6.16)	180	(9.64)
all	308	249	(328.69)	216	(291.38)	187	(361.32)	261	(59.96)	242	(32.89)	204	(33.79)

Table C.12: Performance of models for the S^{da} transformation match score and small clustering parameters. First (second) half corresponds to disabled (enabled) ligand interpolation.

	Clustering			Allowed	Transformation Matching		Ligand
Code	Pre	Intra	Post	Clashes	Displacement	RMSD	Interpolation
sc_49	1.0	0.0	1.0	10	1.0	3.0	n
sc_50	1.0	0.0	1.0	25	1.0	3.0	n
sc_51	1.0	0.0	1.0	50	1.0	3.0	n
sc_52	1.0	0.0	1.0	150	1.0	3.0	n
sc_53	1.0	0.0	1.0	10	1.5	4.5	n
sc_54	1.0	0.0	1.0	25	1.5	4.5	n
sc_55	1.0	0.0	1.0	50	1.5	4.5	n
sc_56	1.0	0.0	1.0	150	1.5	4.5	n
sc_57	1.0	0.0	1.0	10	2.5	7.5	n
sc_58	1.0	0.0	1.0	25	2.5	7.5	n
sc_59	1.0	0.0	1.0	50	2.5	7.5	n
sc_60	1.0	0.0	1.0	150	2.5	7.5	n
sc_61	1.0	1.0	2.0	10	1.0	3.0	n
sc_62	1.0	1.0	2.0	25	1.0	3.0	n
sc_63	1.0	1.0	2.0	50	1.0	3.0	n
sc_64	1.0	1.0	2.0	150	1.0	3.0	n
sc_65	1.0	1.0	2.0	10	1.5	4.5	n
sc_66	1.0	1.0	2.0	25	1.5	4.5	n
sc_67	1.0	1.0	2.0	50	1.5	4.5	n
sc_68	1.0	1.0	2.0	150	1.5	4.5	n
sc_69	1.0	1.0	2.0	10	2.5	7.5	n
sc_70	1.0	1.0	2.0	25	2.5	7.5	n
sc_71	1.0	1.0	2.0	50	2.5	7.5	n
sc_72	1.0	1.0	2.0	150	2.5	7.5	n
sc_73	1.0	0.0	1.0	10	1.0	3.0	y
sc_74	1.0	0.0	1.0	25	1.0	3.0	y
sc_75	1.0	0.0	1.0	50	1.0	3.0	y
sc_76	1.0	0.0	1.0	150	1.0	3.0	y
sc_77	1.0	0.0	1.0	10	1.5	4.5	y
sc_78	1.0	0.0	1.0	25	1.5	4.5	y
sc_79	1.0	0.0	1.0	50	1.5	4.5	y
sc_80	1.0	0.0	1.0	150	1.5	4.5	y
sc_81	1.0	0.0	1.0	10	2.5	7.5	y
sc_82	1.0	0.0	1.0	25	2.5	7.5	y
sc_83	1.0	0.0	1.0	50	2.5	7.5	y
sc_84	1.0	0.0	1.0	150	2.5	7.5	y
sc_85	1.0	1.0	2.0	10	1.0	3.0	y
sc_86	1.0	1.0	2.0	25	1.0	3.0	y
sc_87	1.0	1.0	2.0	50	1.0	3.0	y
sc_88	1.0	1.0	2.0	150	1.0	3.0	y
sc_89	1.0	1.0	2.0	10	1.5	4.5	y
sc_90	1.0	1.0	2.0	25	1.5	4.5	y
sc_91	1.0	1.0	2.0	50	1.5	4.5	y
sc_92	1.0	1.0	2.0	150	1.5	4.5	y
sc_93	1.0	1.0	2.0	10	2.5	7.5	y
sc_94	1.0	1.0	2.0	25	2.5	7.5	y
sc_95	1.0	1.0	2.0	50	2.5	7.5	y
sc_96	1.0	1.0	2.0	150	2.5	7.5	y

Table C.13: Parameters for runs with S^{rmsd} transformation matching and small clustering parameters.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	≤ 10		≤ 5		≤ 3		≤ 2.5		≤ 2.0		≤ 1.0	
sc_49	306	156	(2.94)	92	(2.88)	58	(5.81)	157	(4.20)	138	(3.29)	64	(7.80)
sc_50	307	170	(3.04)	126	(1.87)	100	(4.37)	178	(2.73)	157	(3.80)	107	(3.80)
sc_51	306	183	(1.40)	141	(1.55)	119	(3.27)	190	(2.52)	173	(2.01)	128	(3.72)
sc_52	306	185	(1.97)	149	(1.76)	130	(4.78)	187	(2.04)	173	(1.92)	135	(4.24)
sc_53	304	154	(3.30)	100	(3.12)	59	(4.12)	157	(3.59)	132	(3.16)	72	(5.01)
sc_54	307	167	(2.44)	127	(2.14)	98	(3.41)	172	(2.22)	156	(3.46)	110	(4.19)
sc_55	306	176	(1.34)	139	(1.83)	111	(2.68)	186	(2.54)	167	(1.78)	122	(4.24)
sc_56	307	183	(1.80)	154	(1.53)	127	(3.93)	195	(2.51)	179	(2.42)	134	(4.57)
sc_57	302	155	(2.76)	100	(3.36)	60	(3.92)	155	(3.06)	134	(2.44)	79	(4.72)
sc_58	305	168	(2.65)	129	(2.67)	96	(5.08)	172	(2.62)	159	(3.34)	105	(3.90)
sc_59	305	177	(1.85)	142	(1.68)	112	(4.11)	186	(1.75)	175	(2.38)	119	(2.67)
sc_60	306	181	(2.22)	148	(1.43)	119	(3.73)	188	(2.59)	178	(2.54)	131	(3.88)
sc_61	307	177	(3.62)	124	(3.56)	84	(5.36)	178	(4.31)	160	(4.86)	96	(9.16)
sc_62	307	191	(3.03)	155	(3.07)	122	(5.58)	192	(2.46)	176	(2.59)	130	(4.91)
sc_63	307	202	(2.44)	166	(2.33)	140	(4.19)	204	(2.22)	192	(2.14)	150	(3.43)
sc_64	308	204	(1.65)	174	(1.99)	148	(3.72)	206	(1.55)	198	(2.51)	156	(3.40)
sc_65	305	178	(3.57)	130	(2.28)	87	(4.97)	177	(3.60)	156	(2.99)	99	(8.55)
sc_66	308	191	(2.61)	152	(2.76)	120	(3.27)	194	(1.98)	182	(2.70)	130	(4.68)
sc_67	307	198	(1.81)	168	(2.60)	143	(3.59)	206	(2.96)	190	(2.40)	147	(3.53)
sc_68	308	206	(1.94)	177	(2.25)	150	(3.55)	212	(1.52)	195	(1.58)	161	(3.81)
sc_69	303	184	(4.19)	128	(2.03)	84	(6.07)	182	(3.09)	169	(4.63)	101	(5.51)
sc_70	308	189	(3.21)	152	(2.64)	125	(3.98)	199	(2.90)	185	(2.68)	128	(3.18)
sc_71	308	201	(2.64)	169	(2.54)	143	(3.79)	210	(2.94)	194	(2.04)	151	(3.36)
sc_72	308	206	(2.04)	174	(2.11)	145	(3.54)	210	(2.55)	196	(1.56)	158	(4.11)
sc_73	304	155	(3.67)	91	(3.07)	57	(5.16)	155	(4.11)	133	(3.76)	63	(4.73)
sc_74	307	169	(2.38)	128	(1.75)	97	(3.91)	176	(2.41)	155	(2.79)	109	(4.38)
sc_75	307	175	(1.45)	140	(1.83)	103	(2.59)	179	(2.09)	163	(1.77)	120	(4.94)
sc_76	307	180	(2.09)	150	(1.91)	119	(4.82)	191	(2.37)	175	(1.74)	124	(4.66)
sc_77	305	158	(3.57)	103	(2.68)	58	(6.16)	161	(3.18)	136	(3.80)	76	(6.20)
sc_78	306	166	(1.96)	128	(1.88)	93	(3.91)	169	(1.57)	151	(2.62)	103	(4.30)
sc_79	306	171	(1.64)	138	(1.47)	109	(2.45)	178	(2.09)	161	(2.71)	117	(3.82)
sc_80	307	177	(2.90)	147	(1.85)	122	(5.32)	189	(3.10)	169	(2.66)	127	(4.91)
sc_81	305	156	(3.14)	105	(2.75)	62	(4.94)	156	(3.76)	137	(2.30)	78	(4.17)
sc_82	304	163	(2.20)	127	(2.32)	96	(4.92)	169	(2.20)	155	(2.51)	104	(3.77)
sc_83	307	168	(2.23)	136	(2.07)	113	(5.86)	175	(2.17)	164	(2.85)	120	(3.63)
sc_84	307	173	(2.46)	139	(2.05)	112	(5.53)	182	(2.52)	166	(2.07)	124	(3.55)
sc_85	306	181	(3.78)	132	(3.15)	89	(5.35)	181	(3.84)	163	(3.63)	95	(6.73)
sc_86	307	196	(3.65)	155	(2.88)	126	(5.37)	201	(2.74)	184	(2.56)	134	(4.13)
sc_87	307	201	(2.19)	167	(2.29)	141	(3.29)	200	(1.98)	187	(1.68)	148	(3.34)
sc_88	307	206	(1.77)	177	(2.04)	152	(3.35)	205	(1.40)	200	(2.25)	160	(3.97)
sc_89	305	178	(3.96)	133	(2.42)	92	(5.30)	176	(3.38)	159	(3.21)	100	(8.58)
sc_90	307	195	(2.91)	155	(2.51)	127	(3.80)	192	(1.94)	180	(2.49)	132	(2.70)
sc_91	305	200	(1.99)	167	(2.23)	144	(3.36)	203	(2.40)	191	(2.58)	151	(2.84)
sc_92	308	206	(1.72)	180	(2.12)	151	(3.98)	208	(2.17)	196	(2.22)	162	(3.43)
sc_93	306	178	(3.49)	126	(2.07)	92	(4.90)	179	(2.81)	164	(4.61)	105	(5.76)
sc_94	307	193	(2.51)	154	(2.06)	125	(3.37)	199	(2.38)	185	(2.46)	131	(2.54)
sc_95	307	204	(1.97)	169	(2.24)	143	(2.87)	208	(2.86)	195	(1.92)	150	(2.48)
sc_96	307	203	(1.63)	178	(2.59)	149	(4.19)	210	(2.29)	197	(2.28)	154	(2.61)
div2	308	219	(12.88)	184	(12.96)	158	(16.53)	232	(5.15)	214	(2.96)	169	(6.60)
div3	308	225	(16.25)	196	(16.46)	167	(17.74)	240	(5.10)	222	(3.79)	184	(8.47)
all	308	247	(258.34)	214	(298.07)	181	(260.99)	255	(37.87)	243	(34.54)	204	(61.16)

Table C.14: Performance of models for the S^{rmsd} transformation match score and small clustering parameters. First (second) half corresponds to disabled (enabled) ligand interpolation.

	Clustering			Allowed	Transformation Matching		Ligand
Code	Pre	Intra	Post	Clashes	Displacement	Angle	Interpolation
to_01	0.0	0.0	0.0	10	1.0	5.0	n
to_02	0.0	0.0	0.0	25	1.0	5.0	n
to_03	0.0	0.0	0.0	50	1.0	5.0	n
to_04	0.0	0.0	0.0	150	1.0	5.0	n
to_05	0.0	0.0	0.0	10	1.5	10.0	n
to_06	0.0	0.0	0.0	25	1.5	10.0	n
to_07	0.0	0.0	0.0	50	1.5	10.0	n
to_08	0.0	0.0	0.0	150	1.5	10.0	n
to_09	0.0	0.0	0.0	10	2.5	15.0	n
to_10	0.0	0.0	0.0	25	2.5	15.0	n
to_11	0.0	0.0	0.0	50	2.5	15.0	n
to_12	0.0	0.0	0.0	150	2.5	15.0	n
to_13	1.0	0.0	0.0	10	1.0	5.0	n
to_14	1.0	0.0	0.0	25	1.0	5.0	n
to_15	1.0	0.0	0.0	50	1.0	5.0	n
to_16	1.0	0.0	0.0	150	1.0	5.0	n
to_17	1.0	0.0	0.0	10	1.5	10.0	n
to_18	1.0	0.0	0.0	25	1.5	10.0	n
to_19	1.0	0.0	0.0	50	1.5	10.0	n
to_20	1.0	0.0	0.0	150	1.5	10.0	n
to_21	1.0	0.0	0.0	10	2.5	15.0	n
to_22	1.0	0.0	0.0	25	2.5	15.0	n
to_23	1.0	0.0	0.0	50	2.5	15.0	n
to_24	1.0	0.0	0.0	150	2.5	15.0	n
to_25	0.0	0.0	0.0	10	1.0	5.0	y
to_26	0.0	0.0	0.0	25	1.0	5.0	y
to_27	0.0	0.0	0.0	50	1.0	5.0	y
to_28	0.0	0.0	0.0	150	1.0	5.0	y
to_29	0.0	0.0	0.0	10	1.5	10.0	y
to_30	0.0	0.0	0.0	25	1.5	10.0	y
to_31	0.0	0.0	0.0	50	1.5	10.0	y
to_32	0.0	0.0	0.0	150	1.5	10.0	y
to_33	0.0	0.0	0.0	10	2.5	15.0	y
to_34	0.0	0.0	0.0	25	2.5	15.0	y
to_35	0.0	0.0	0.0	50	2.5	15.0	y
to_36	0.0	0.0	0.0	150	2.5	15.0	y
to_37	1.0	0.0	0.0	10	1.0	5.0	y
to_38	1.0	0.0	0.0	25	1.0	5.0	y
to_39	1.0	0.0	0.0	50	1.0	5.0	y
to_40	1.0	0.0	0.0	150	1.0	5.0	y
to_41	1.0	0.0	0.0	10	1.5	10.0	y
to_42	1.0	0.0	0.0	25	1.5	10.0	y
to_43	1.0	0.0	0.0	50	1.5	10.0	y
to_44	1.0	0.0	0.0	150	1.5	10.0	y
to_45	1.0	0.0	0.0	10	2.5	15.0	y
to_46	1.0	0.0	0.0	25	2.5	15.0	y
to_47	1.0	0.0	0.0	50	2.5	15.0	y
to_48	1.0	0.0	0.0	150	2.5	15.0	y

Table C.15: Parameters for runs with enabled S^{da} transformation matching but disabled intra- and post-clustering.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	≤ 10		≤ 5		≤ 3		≤ 2.5		≤ 2.0		≤ 1.0	
to_01	308	164	(4.61)	107	(5.94)	49	(6.88)	164	(3.75)	151	(6.67)	62	(8.69)
to_02	308	167	(2.32)	128	(3.42)	90	(8.91)	172	(2.98)	159	(4.59)	97	(8.34)
to_03	308	172	(1.70)	139	(2.58)	106	(9.20)	176	(2.52)	165	(2.12)	111	(8.18)
to_04	308	179	(1.44)	154	(2.40)	115	(8.00)	184	(2.17)	174	(2.65)	126	(8.64)
to_05	307	173	(3.17)	118	(7.15)	56	(5.91)	166	(3.23)	153	(5.12)	78	(12.44)
to_06	308	173	(2.20)	131	(3.97)	97	(7.00)	175	(3.52)	159	(4.82)	100	(6.52)
to_07	308	174	(2.93)	140	(3.31)	110	(6.41)	172	(2.94)	165	(3.99)	109	(5.07)
to_08	308	180	(2.05)	153	(2.98)	125	(8.15)	184	(3.11)	172	(3.88)	127	(6.32)
to_09	308	167	(2.17)	121	(7.88)	64	(7.52)	167	(3.03)	154	(4.79)	78	(9.00)
to_10	307	172	(2.55)	132	(3.10)	96	(4.75)	174	(3.35)	160	(4.50)	106	(5.07)
to_11	307	177	(2.83)	142	(3.23)	113	(5.75)	176	(2.22)	167	(3.42)	117	(3.87)
to_12	307	179	(1.87)	145	(2.44)	120	(4.88)	181	(2.65)	173	(3.71)	123	(3.73)
to_13	308	166	(3.28)	108	(5.19)	59	(16.08)	159	(3.03)	139	(6.60)	70	(14.09)
to_14	308	173	(2.65)	127	(4.31)	92	(6.67)	170	(3.28)	152	(4.12)	102	(7.39)
to_15	308	179	(1.53)	140	(2.27)	107	(7.98)	175	(2.78)	163	(3.05)	118	(9.25)
to_16	308	180	(1.69)	150	(2.13)	121	(8.88)	179	(3.94)	168	(2.61)	123	(9.57)
to_17	308	168	(4.03)	111	(5.77)	61	(8.89)	160	(4.12)	141	(5.34)	75	(9.77)
to_18	308	178	(2.75)	132	(5.32)	97	(7.01)	176	(3.40)	157	(4.34)	102	(7.13)
to_19	308	181	(2.14)	145	(4.90)	114	(5.57)	179	(4.20)	165	(3.28)	120	(5.72)
to_20	308	179	(1.90)	149	(3.72)	123	(6.05)	179	(3.66)	167	(3.19)	125	(6.89)
to_21	307	173	(4.28)	123	(5.13)	67	(6.76)	171	(3.53)	147	(4.20)	86	(9.66)
to_22	308	180	(2.87)	139	(3.35)	105	(5.03)	180	(3.53)	165	(3.35)	119	(6.31)
to_23	308	180	(2.37)	144	(2.35)	120	(5.29)	180	(3.34)	172	(3.77)	123	(5.29)
to_24	308	181	(1.74)	150	(2.96)	126	(6.14)	181	(2.93)	171	(3.84)	131	(6.21)
to_25	308	156	(3.91)	101	(8.89)	42	(10.07)	158	(4.85)	139	(6.88)	57	(12.81)
to_26	308	159	(2.81)	119	(4.15)	78	(4.92)	162	(3.24)	145	(2.91)	77	(9.52)
to_27	307	170	(2.91)	129	(3.21)	89	(5.63)	168	(3.14)	154	(3.69)	91	(7.20)
to_28	308	170	(2.81)	144	(3.39)	96	(7.38)	176	(3.02)	162	(2.44)	95	(7.35)
to_29	307	167	(3.94)	105	(6.09)	52	(7.92)	164	(3.48)	149	(5.21)	71	(14.06)
to_30	308	165	(2.92)	117	(5.15)	81	(3.17)	164	(3.09)	146	(3.52)	80	(4.99)
to_31	308	170	(3.41)	136	(3.82)	102	(5.44)	168	(3.24)	158	(4.10)	100	(4.28)
to_32	308	173	(2.58)	143	(2.17)	105	(4.31)	170	(3.19)	157	(3.83)	106	(5.13)
to_33	308	170	(4.00)	113	(5.98)	64	(9.97)	168	(4.85)	150	(4.91)	75	(9.95)
to_34	307	163	(3.02)	116	(3.86)	85	(5.11)	167	(2.76)	145	(3.76)	86	(4.41)
to_35	306	168	(3.27)	130	(2.86)	98	(3.78)	162	(3.42)	155	(2.99)	101	(4.96)
to_36	306	167	(1.70)	133	(1.86)	103	(2.73)	165	(2.81)	150	(2.59)	106	(5.36)
to_37	308	154	(4.24)	101	(7.62)	47	(15.57)	155	(3.30)	136	(8.01)	65	(18.97)
to_38	308	163	(2.96)	120	(5.04)	76	(8.13)	164	(2.65)	145	(5.25)	85	(9.62)
to_39	308	168	(1.88)	131	(2.56)	94	(6.91)	163	(2.50)	150	(4.02)	96	(8.71)
to_40	308	173	(2.14)	146	(2.93)	101	(7.12)	172	(3.44)	158	(4.23)	104	(6.43)
to_41	308	164	(2.88)	105	(4.17)	56	(10.02)	155	(3.91)	137	(5.64)	69	(12.29)
to_42	308	163	(4.40)	122	(5.45)	85	(5.69)	160	(3.37)	145	(3.93)	98	(4.41)
to_43	308	165	(2.12)	134	(3.87)	101	(7.53)	163	(2.71)	151	(3.44)	105	(5.46)
to_44	308	170	(1.89)	141	(4.34)	108	(9.31)	168	(3.46)	157	(3.01)	113	(7.44)
to_45	308	166	(3.81)	116	(5.39)	61	(7.98)	162	(3.46)	139	(3.51)	77	(12.17)
to_46	308	161	(3.68)	123	(3.57)	90	(3.84)	162	(2.40)	148	(3.80)	100	(5.08)
to_47	308	164	(1.93)	137	(3.22)	110	(5.95)	166	(2.12)	156	(2.88)	115	(5.68)
to_48	308	168	(2.16)	141	(3.83)	109	(5.83)	168	(2.96)	159	(4.22)	117	(5.81)
div2	308	200	(11.20)	162	(17.82)	125	(25.80)	205	(4.59)	198	(5.38)	132	(5.68)
div3	308	208	(21.20)	173	(18.32)	139	(27.58)	213	(7.59)	203	(8.97)	147	(12.27)
all	308	225	(274.62)	192	(199.74)	160	(360.17)	234	(82.15)	222	(52.99)	175	(93.02)

Table C.16: Performance of models with enabled S^{da} transformation match score and disabled intra- and post-clustering. First (second) half corresponds to disabled (enabled) ligand interpolation.

	Clustering			Allowed	Transformation Matching		Ligand
Code	Pre	Intra	Post	Clashes	Displacement	RMSD	Interpolation
to_49	0.0	0.0	0.0	10	1.0	3.0	n
to_50	0.0	0.0	0.0	25	1.0	3.0	n
to_51	0.0	0.0	0.0	50	1.0	3.0	n
to_52	0.0	0.0	0.0	150	1.0	3.0	n
to_53	0.0	0.0	0.0	10	1.5	4.5	n
to_54	0.0	0.0	0.0	25	1.5	4.5	n
to_55	0.0	0.0	0.0	50	1.5	4.5	n
to_56	0.0	0.0	0.0	150	1.5	4.5	n
to_57	0.0	0.0	0.0	10	2.5	7.5	n
to_58	0.0	0.0	0.0	25	2.5	7.5	n
to_59	0.0	0.0	0.0	50	2.5	7.5	n
to_60	0.0	0.0	0.0	150	2.5	7.5	n
to_61	1.0	0.0	0.0	10	1.0	3.0	n
to_62	1.0	0.0	0.0	25	1.0	3.0	n
to_63	1.0	0.0	0.0	50	1.0	3.0	n
to_64	1.0	0.0	0.0	150	1.0	3.0	n
to_65	1.0	0.0	0.0	10	1.5	4.5	n
to_66	1.0	0.0	0.0	25	1.5	4.5	n
to_67	1.0	0.0	0.0	50	1.5	4.5	n
to_68	1.0	0.0	0.0	150	1.5	4.5	n
to_69	1.0	0.0	0.0	10	2.5	7.5	n
to_70	1.0	0.0	0.0	25	2.5	7.5	n
to_71	1.0	0.0	0.0	50	2.5	7.5	n
to_72	1.0	0.0	0.0	150	2.5	7.5	n
to_73	0.0	0.0	0.0	10	1.0	3.0	y
to_74	0.0	0.0	0.0	25	1.0	3.0	y
to_75	0.0	0.0	0.0	50	1.0	3.0	y
to_76	0.0	0.0	0.0	150	1.0	3.0	y
to_77	0.0	0.0	0.0	10	1.5	4.5	y
to_78	0.0	0.0	0.0	25	1.5	4.5	y
to_79	0.0	0.0	0.0	50	1.5	4.5	y
to_80	0.0	0.0	0.0	150	1.5	4.5	y
to_81	0.0	0.0	0.0	10	2.5	7.5	y
to_82	0.0	0.0	0.0	25	2.5	7.5	y
to_83	0.0	0.0	0.0	50	2.5	7.5	y
to_84	0.0	0.0	0.0	150	2.5	7.5	y
to_85	1.0	0.0	0.0	10	1.0	3.0	y
to_86	1.0	0.0	0.0	25	1.0	3.0	y
to_87	1.0	0.0	0.0	50	1.0	3.0	y
to_88	1.0	0.0	0.0	150	1.0	3.0	y
to_89	1.0	0.0	0.0	10	1.5	4.5	y
to_90	1.0	0.0	0.0	25	1.5	4.5	y
to_91	1.0	0.0	0.0	50	1.5	4.5	y
to_92	1.0	0.0	0.0	150	1.5	4.5	y
to_93	1.0	0.0	0.0	10	2.5	7.5	y
to_94	1.0	0.0	0.0	25	2.5	7.5	y
to_95	1.0	0.0	0.0	50	2.5	7.5	y
to_96	1.0	0.0	0.0	150	2.5	7.5	y

Table C.17: Parameters for runs with enabled S^{rmsd} transformation matching but disabled intra- and post-clustering.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	≤ 10		≤ 5		≤ 3		≤ 2.5		≤ 2.0		≤ 1.0	
to_49	308	168	(3.70)	116	(6.45)	56	(6.25)	162	(4.71)	146	(6.73)	74	(13.43)
to_50	308	174	(2.70)	128	(3.51)	97	(5.09)	169	(3.37)	154	(3.08)	104	(6.20)
to_51	308	172	(2.67)	142	(3.37)	110	(8.49)	173	(2.57)	161	(3.18)	117	(7.20)
to_52	308	179	(2.04)	149	(3.60)	116	(8.49)	180	(3.37)	168	(4.49)	123	(7.51)
to_53	308	173	(4.63)	117	(6.50)	69	(8.41)	168	(3.70)	151	(6.65)	79	(9.68)
to_54	308	168	(2.36)	130	(2.94)	96	(4.03)	173	(3.28)	155	(3.70)	102	(3.80)
to_55	308	176	(1.18)	141	(2.99)	110	(3.98)	181	(2.24)	163	(2.44)	117	(4.28)
to_56	308	178	(1.47)	149	(1.85)	116	(3.49)	183	(2.04)	168	(2.58)	126	(4.22)
to_57	307	165	(3.67)	111	(5.10)	67	(6.78)	162	(2.68)	148	(4.16)	82	(9.91)
to_58	308	164	(3.02)	121	(4.55)	87	(4.83)	167	(2.63)	150	(4.11)	93	(5.40)
to_59	308	168	(3.12)	133	(3.47)	103	(3.13)	174	(2.47)	158	(3.28)	109	(2.62)
to_60	307	172	(2.45)	139	(3.18)	107	(2.79)	174	(1.97)	162	(3.75)	113	(3.01)
to_61	308	163	(4.01)	109	(3.47)	61	(7.48)	159	(3.31)	141	(5.80)	72	(10.22)
to_62	308	177	(3.01)	134	(3.16)	101	(5.40)	174	(3.05)	157	(3.13)	110	(6.32)
to_63	308	183	(2.23)	143	(2.77)	117	(7.95)	176	(2.97)	165	(2.84)	119	(6.85)
to_64	308	185	(2.42)	150	(2.67)	123	(7.15)	179	(2.84)	169	(2.55)	129	(6.85)
to_65	308	173	(4.45)	116	(3.37)	73	(5.90)	161	(3.20)	144	(3.17)	87	(9.37)
to_66	308	177	(3.10)	135	(3.01)	96	(4.95)	176	(2.68)	162	(3.52)	102	(3.96)
to_67	308	181	(1.97)	141	(2.35)	113	(6.35)	179	(2.73)	167	(2.57)	115	(3.70)
to_68	308	184	(1.98)	152	(2.57)	119	(4.05)	183	(2.21)	170	(2.34)	121	(4.29)
to_69	308	166	(4.49)	110	(4.35)	70	(5.19)	157	(2.71)	143	(3.57)	86	(7.27)
to_70	308	173	(2.45)	128	(3.52)	93	(3.55)	174	(2.37)	156	(3.26)	103	(3.83)
to_71	308	177	(3.08)	135	(3.00)	106	(4.93)	173	(2.09)	163	(2.10)	113	(4.34)
to_72	308	183	(1.96)	143	(2.19)	111	(2.95)	176	(1.86)	165	(2.30)	121	(4.26)
to_73	308	158	(3.68)	110	(7.50)	49	(12.69)	156	(4.87)	142	(6.34)	64	(11.66)
to_74	308	162	(2.83)	116	(3.71)	77	(5.00)	161	(3.51)	144	(2.50)	88	(7.52)
to_75	308	168	(2.21)	129	(3.09)	94	(5.32)	162	(3.33)	151	(3.23)	97	(6.36)
to_76	308	168	(1.78)	137	(2.08)	98	(7.14)	168	(2.36)	153	(2.22)	105	(7.52)
to_77	308	171	(4.30)	114	(3.86)	59	(9.69)	165	(3.64)	148	(4.24)	74	(10.03)
to_78	308	170	(2.49)	122	(3.42)	82	(4.65)	169	(2.02)	149	(2.74)	90	(5.03)
to_79	308	173	(2.23)	133	(2.62)	103	(3.28)	171	(2.93)	159	(3.55)	105	(4.89)
to_80	308	173	(2.22)	138	(2.63)	107	(5.68)	174	(2.57)	160	(3.07)	111	(6.15)
to_81	308	162	(3.92)	106	(2.84)	68	(10.97)	160	(3.76)	141	(2.93)	78	(9.55)
to_82	308	159	(2.23)	115	(4.43)	75	(5.73)	162	(1.75)	144	(3.52)	86	(9.44)
to_83	307	158	(2.80)	119	(2.32)	96	(3.35)	163	(3.12)	143	(2.62)	97	(3.67)
to_84	307	157	(2.18)	125	(1.81)	97	(5.49)	161	(3.25)	147	(2.57)	100	(3.31)
to_85	308	165	(3.49)	110	(7.65)	54	(7.24)	158	(3.82)	141	(5.65)	68	(9.29)
to_86	308	167	(3.51)	118	(5.35)	82	(5.46)	164	(3.44)	148	(3.58)	91	(8.07)
to_87	308	172	(2.89)	129	(3.10)	95	(7.02)	167	(2.95)	152	(3.36)	97	(5.10)
to_88	308	177	(2.43)	139	(2.61)	103	(6.28)	172	(2.24)	159	(2.18)	109	(4.65)
to_89	308	171	(3.13)	114	(5.24)	63	(6.29)	162	(3.26)	145	(3.55)	79	(9.16)
to_90	308	168	(3.26)	121	(2.81)	84	(4.30)	168	(3.51)	149	(3.61)	93	(3.96)
to_91	308	171	(1.57)	129	(2.91)	97	(5.65)	166	(1.86)	155	(1.66)	101	(4.23)
to_92	308	174	(2.20)	139	(2.40)	100	(5.01)	171	(1.49)	161	(2.23)	107	(4.41)
to_93	308	162	(4.57)	114	(4.00)	67	(5.12)	156	(3.04)	140	(2.56)	87	(6.78)
to_94	308	161	(3.54)	118	(3.32)	86	(2.98)	162	(2.81)	149	(2.98)	97	(3.97)
to_95	308	157	(2.60)	121	(2.49)	94	(4.95)	161	(2.60)	147	(2.09)	97	(2.24)
to_96	308	159	(2.01)	126	(1.98)	95	(4.17)	161	(2.16)	147	(2.94)	101	(3.46)
div2	308	202	(12.54)	162	(15.59)	122	(25.54)	200	(4.96)	187	(4.83)	130	(9.85)
div3	308	206	(16.51)	161	(16.58)	127	(26.39)	209	(8.30)	192	(7.76)	136	(11.91)
all	308	225	(247.19)	188	(216.98)	154	(410.51)	231	(92.50)	216	(39.47)	166	(136.83)

Table C.18: Performance of models with enabled S^{rmsd} transformation match score and disabled intra- and post-clustering. First (second) half corresponds to disabled (enabled) ligand interpolation.

	Clustering			Allowed
Code	Pre	Intra	Post	Clashes
co_01	1.0	0.0	1.0	10
co_02	1.0	0.0	1.0	25
co_03	1.0	0.0	1.0	50
co_04	1.0	0.0	1.0	150
co_05	1.0	1.0	2.0	10
co_06	1.0	1.0	2.0	25
co_07	1.0	1.0	2.0	50
co_08	1.0	1.0	2.0	150
co_09	1.0	2.0	3.0	10
co_10	1.0	2.0	3.0	25
co_11	1.0	2.0	3.0	50
co_12	1.0	2.0	3.0	150
co_13	1.0	3.0	5.0	10
co_14	1.0	3.0	5.0	25
co_15	1.0	3.0	5.0	50
co_16	1.0	3.0	5.0	150

Table C.19: Parameters for runs with disabled transformation match score (and thus ligand interpolation) but enabled clustering.

	Clustering			Allowed
Code	Pre	Intra	Post	Clashes
base_01	0.0	0.0	0.0	10
base_02	0.0	0.0	0.0	25
base_03	0.0	0.0	0.0	50
base_04	0.0	0.0	0.0	150
base_05	1.0	0.0	0.0	10
base_06	1.0	0.0	0.0	25
base_07	1.0	0.0	0.0	50
base_08	1.0	0.0	0.0	150

Table C.20: Parameters for the baseline runs. Only the number of clashes and pre-clustering differ. Intra- and post-clustering as well as transformation matching and hence ligand interpolation are disabled.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	$\leq$ 10		$\leq$ 5		$\leq$ 3		$\leq$ 2.5		$\leq$ 2.0		$\leq$ 1.0	
co_01	307	60	(6.95)	23	(6.35)	9	(4.00)	63	(8.68)	46	(6.52)	19	(15.68)
co_02	307	65	(8.02)	25	(5.12)	11	(1.36)	62	(4.79)	54	(7.52)	21	(4.10)
co_03	307	67	(3.93)	31	(2.39)	18	(9.50)	68	(4.85)	53	(3.06)	26	(6.65)
co_04	307	63	(6.17)	36	(2.14)	21	(16.95)	65	(5.12)	55	(2.25)	28	(3.04)
co_05	307	77	(10.12)	31	(5.42)	15	(4.40)	79	(11.97)	60	(13.52)	26	(9.23)
co_06	307	86	(7.44)	40	(4.00)	24	(7.00)	87	(9.36)	67	(6.69)	34	(10.53)
co_07	307	94	(7.12)	45	(4.18)	32	(5.44)	87	(5.18)	71	(6.72)	40	(5.60)
co_08	307	89	(5.04)	47	(1.96)	34	(11.76)	89	(3.97)	74	(2.68)	41	(4.10)
co_09	307	84	(10.18)	36	(10.92)	17	(8.65)	86	(9.87)	64	(10.92)	25	(3.80)
co_10	307	95	(8.44)	47	(3.66)	27	(6.26)	91	(8.67)	71	(5.34)	34	(4.09)
co_11	307	101	(5.27)	53	(3.43)	37	(5.65)	97	(8.56)	79	(5.09)	42	(3.07)
co_12	307	99	(5.98)	58	(2.59)	39	(6.92)	101	(6.21)	82	(3.07)	44	(3.93)
co_13	307	96	(11.86)	35	(10.51)	11	(4.18)	88	(10.16)	64	(10.11)	25	(11.80)
co_14	307	106	(8.52)	50	(5.62)	24	(3.83)	103	(8.98)	78	(7.99)	32	(5.00)
co_15	307	113	(5.39)	61	(3.98)	33	(1.85)	110	(6.76)	92	(6.12)	42	(4.40)
co_16	307	111	(4.15)	62	(1.58)	38	(2.74)	109	(4.54)	89	(2.90)	45	(4.69)
div2	307	123	(30.04)	73	(34.74)	46	(33.72)	125	(12.98)	103	(13.09)	52	(5.90)
div3	307	125	(28.66)	76	(47.41)	45	(36.89)	128	(12.35)	107	(13.87)	52	(6.88)
all	307	130	(197.23)	82	(303.63)	54	(291.13)	135	(75.37)	113	(69.90)	60	(44.70)

Table C.21: Performance of models with disabled transformation match score and enabled intra- and post-clustering. Models in one block use the same clustering parameters.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	$\leq$ 10		$\leq$ 5		$\leq$ 3		$\leq$ 2.5		$\leq$ 2.0		$\leq$ 1.0	
base_01	307	45	(8.31)	13	(7.00)	8	(4.25)	44	(7.61)	32	(7.84)	13	(17.62)
base_02	307	45	(5.84)	16	(7.00)	9	(3.22)	51	(6.80)	36	(5.08)	14	(4.50)
base_03	307	49	(3.53)	24	(4.29)	13	(5.00)	54	(5.22)	40	(1.70)	20	(8.30)
base_04	307	46	(1.37)	28	(5.82)	11	(5.36)	55	(7.07)	44	(3.27)	20	(7.65)
base_05	307	46	(5.09)	16	(5.50)	8	(4.50)	50	(6.26)	35	(5.57)	12	(11.83)
base_06	307	51	(7.12)	19	(10.89)	9	(3.22)	56	(7.36)	41	(6.80)	15	(4.27)
base_07	307	52	(4.44)	24	(3.50)	13	(3.15)	60	(8.63)	46	(4.85)	21	(6.29)
base_08	307	48	(3.35)	31	(4.06)	14	(3.00)	56	(7.38)	47	(5.53)	23	(3.30)
div2	307	60	(22.67)	32	(17.53)	14	(24.14)	69	(22.07)	56	(18.54)	26	(16.00)
div3	307	67	(39.81)	34	(32.06)	14	(25.29)	73	(29.16)	59	(20.39)	26	(16.00)
all	307	72	(104.22)	35	(54.91)	15	(47.67)	76	(65.64)	63	(39.38)	29	(26.83)

Table C.22: Performance of baseline models with disabled transformation match score and disabled intra- and post-clustering. First (second) half with disabled (enabled) pre-clustering (pre-cluster RMSD 1.0Å)

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	≤ 10		≤ 5		≤ 3		≤ 2.5		≤ 2.0		≤ 1.0	
com_01	17	13	(3.38)	9	(1.33)	7	(1.29)	12	(3.50)	10	(1.10)	7	(1.29)
com_02	17	13	(1.00)	11	(1.09)	10	(1.80)	13	(2.77)	11	(1.27)	10	(1.40)
com_03	17	14	(1.00)	13	(1.15)	11	(1.91)	15	(4.87)	12	(1.33)	11	(1.64)
com_04	17	15	(1.00)	15	(1.20)	12	(2.08)	16	(2.00)	14	(1.43)	11	(1.73)
com_05	17	11	(2.09)	8	(3.62)	7	(1.14)	11	(4.27)	9	(1.11)	7	(3.57)
com_06	17	11	(1.00)	10	(1.40)	8	(2.62)	12	(2.92)	10	(1.30)	9	(1.89)
com_07	17	12	(1.00)	12	(1.25)	10	(2.90)	13	(2.15)	11	(1.09)	10	(1.80)
com_08	17	12	(1.00)	12	(1.17)	10	(2.70)	13	(2.08)	12	(5.17)	10	(2.30)
com_09	17	11	(2.00)	9	(1.67)	8	(1.50)	11	(3.36)	9	(1.00)	9	(3.00)
com_10	17	12	(1.08)	10	(1.30)	9	(2.89)	12	(2.92)	10	(1.30)	9	(2.00)
com_11	17	12	(1.08)	11	(1.18)	9	(2.22)	12	(2.17)	10	(1.00)	9	(1.22)
com_12	17	12	(1.25)	11	(1.18)	9	(1.67)	12	(2.17)	11	(5.73)	9	(1.22)
com_13	17	12	(2.33)	6	(1.00)	4	(1.25)	12	(3.33)	9	(9.56)	2	(1.00)
com_14	16	13	(1.00)	13	(1.38)	7	(1.43)	14	(2.43)	11	(1.00)	6	(1.00)
com_15	17	13	(1.00)	13	(1.77)	8	(1.50)	15	(3.40)	12	(1.00)	10	(2.90)
com_16	17	14	(1.00)	13	(1.77)	9	(1.89)	14	(2.21)	12	(1.00)	8	(2.25)
com_17	17	12	(2.58)	8	(1.00)	5	(1.00)	11	(3.45)	10	(8.00)	4	(1.00)
com_18	17	12	(1.42)	11	(1.64)	8	(1.25)	12	(2.75)	10	(1.20)	7	(1.29)
com_19	17	11	(1.00)	11	(2.36)	8	(1.75)	12	(2.58)	10	(1.10)	8	(2.75)
com_20	17	11	(1.00)	11	(2.36)	9	(1.56)	12	(2.67)	10	(1.30)	9	(2.22)
com_21	17	12	(2.67)	9	(1.67)	6	(5.00)	11	(3.36)	10	(8.60)	4	(8.25)
com_22	17	12	(1.50)	11	(1.73)	8	(1.88)	12	(2.75)	10	(1.20)	8	(1.75)
com_23	17	11	(1.00)	11	(2.36)	9	(3.56)	12	(2.50)	10	(1.00)	9	(3.78)
com_24	17	11	(1.00)	11	(2.45)	9	(2.44)	12	(2.58)	10	(1.00)	9	(2.44)
com_25	17	14	(3.14)	9	(1.11)	6	(1.33)	13	(3.23)	11	(1.00)	6	(3.83)
com_26	17	13	(1.00)	11	(1.09)	10	(1.70)	13	(2.77)	11	(1.27)	10	(1.90)
com_27	17	13	(1.00)	12	(1.17)	10	(1.40)	13	(2.08)	11	(1.00)	10	(1.10)
com_28	17	13	(1.00)	13	(1.15)	10	(1.60)	14	(2.00)	12	(1.00)	10	(1.30)
com_29	17	11	(2.09)	8	(3.62)	7	(1.86)	11	(4.27)	9	(1.11)	6	(3.67)
com_30	17	11	(1.18)	10	(1.40)	8	(2.50)	12	(3.00)	10	(1.40)	8	(1.50)
com_31	17	11	(1.00)	11	(1.27)	9	(3.33)	12	(2.25)	10	(1.10)	9	(1.89)
com_32	17	11	(1.00)	11	(1.18)	9	(3.22)	12	(2.17)	11	(8.27)	9	(1.89)
com_33	17	11	(2.45)	9	(1.67)	8	(1.75)	11	(3.45)	9	(1.33)	9	(4.89)
com_34	17	12	(1.08)	10	(1.40)	9	(2.33)	12	(2.92)	10	(1.30)	9	(2.22)
com_35	17	12	(1.08)	11	(1.27)	9	(2.89)	12	(2.17)	10	(1.10)	9	(2.11)
com_36	17	12	(1.25)	11	(1.18)	9	(1.67)	12	(2.17)	11	(8.00)	9	(1.44)
com_37	17	12	(2.42)	7	(2.14)	3	(1.00)	12	(3.33)	9	(8.89)	3	(3.00)
com_38	16	13	(1.38)	13	(2.23)	8	(2.00)	14	(3.21)	12	(2.00)	8	(3.50)
com_39	17	14	(1.00)	14	(1.64)	8	(1.50)	16	(4.25)	13	(1.00)	9	(3.00)
com_40	17	14	(1.00)	13	(1.77)	9	(2.22)	14	(2.21)	12	(1.08)	9	(1.44)
com_41	17	11	(2.64)	7	(1.14)	5	(2.40)	10	(3.70)	9	(9.11)	2	(1.00)
com_42	17	12	(1.42)	11	(1.64)	7	(1.43)	12	(2.75)	10	(1.20)	8	(6.50)
com_43	17	12	(1.17)	11	(2.27)	8	(1.38)	12	(2.58)	10	(1.30)	8	(3.88)
com_44	17	11	(1.00)	11	(2.09)	8	(2.75)	12	(2.58)	10	(1.10)	9	(2.78)
com_45	17	12	(2.83)	9	(2.33)	4	(7.50)	11	(3.18)	10	(8.70)	3	(2.67)
com_46	17	12	(1.50)	11	(2.00)	9	(3.22)	12	(2.83)	10	(1.40)	9	(2.56)
com_47	17	12	(1.25)	11	(2.36)	9	(3.56)	12	(2.50)	10	(1.20)	9	(5.00)
com_48	17	11	(1.00)	11	(2.09)	9	(3.22)	12	(2.58)	10	(1.10)	9	(3.00)

Table C.23: Performance of benchmark models for S^{da} transformation match score on Comeau's data set.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	≤ 10		≤ 5		≤ 3		≤ 2.5		≤ 2.0		≤ 1.0	
com_49	17	11	(2.00)	10	(1.00)	8	(1.50)	11	(2.00)	9	(1.11)	9	(2.33)
com_50	17	11	(1.09)	10	(1.60)	8	(2.25)	12	(3.17)	10	(1.40)	8	(2.12)
com_51	17	11	(1.00)	11	(1.36)	9	(1.67)	12	(2.25)	11	(5.91)	9	(1.44)
com_52	17	11	(1.00)	11	(1.36)	9	(1.67)	12	(2.25)	11	(6.27)	9	(1.56)
com_53	17	11	(2.36)	9	(1.44)	7	(1.71)	11	(2.36)	9	(1.44)	8	(2.88)
com_54	17	12	(1.08)	10	(1.50)	9	(2.89)	12	(3.08)	11	(3.64)	9	(1.78)
com_55	17	12	(1.08)	11	(1.36)	9	(2.33)	12	(2.25)	11	(6.18)	9	(1.22)
com_56	17	12	(1.25)	11	(1.36)	9	(1.67)	12	(2.25)	11	(6.45)	9	(1.33)
com_57	17	12	(6.08)	9	(1.22)	7	(1.29)	12	(6.50)	9	(1.22)	8	(2.62)
com_58	17	12	(1.08)	10	(1.40)	9	(2.00)	13	(8.46)	11	(3.64)	9	(1.89)
com_59	17	12	(1.00)	12	(1.25)	9	(2.00)	13	(2.15)	12	(5.75)	9	(1.56)
com_60	17	12	(1.00)	12	(1.25)	9	(1.78)	13	(2.15)	12	(6.00)	9	(1.22)
com_61	17	12	(2.58)	8	(1.38)	5	(4.20)	11	(3.27)	10	(9.60)	3	(12.67)
com_62	17	12	(1.33)	11	(2.09)	9	(1.56)	12	(3.17)	10	(1.00)	8	(2.12)
com_63	17	12	(1.33)	11	(1.82)	8	(1.50)	13	(2.15)	10	(1.10)	9	(3.56)
com_64	17	11	(1.09)	11	(1.73)	9	(1.78)	12	(2.25)	10	(1.10)	9	(3.00)
com_65	17	12	(2.83)	8	(1.75)	5	(6.80)	11	(2.73)	10	(8.50)	4	(10.75)
com_66	17	12	(1.42)	11	(1.55)	9	(1.67)	12	(2.50)	10	(1.20)	8	(1.75)
com_67	17	12	(1.33)	11	(1.73)	8	(1.75)	13	(1.92)	10	(1.10)	8	(3.25)
com_68	17	11	(1.09)	11	(1.55)	9	(1.56)	12	(2.08)	10	(1.00)	9	(4.89)
com_69	17	12	(2.83)	9	(1.89)	6	(1.50)	12	(5.50)	10	(8.50)	4	(7.25)
com_70	17	12	(1.00)	12	(1.42)	9	(3.00)	13	(2.38)	12	(4.92)	8	(3.25)
com_71	17	12	(1.00)	12	(1.67)	9	(2.11)	13	(1.92)	11	(1.09)	9	(4.22)
com_72	17	12	(1.08)	12	(1.50)	9	(1.56)	13	(2.00)	11	(1.00)	9	(3.11)
com_73	17	11	(2.00)	10	(3.00)	8	(2.00)	11	(2.00)	9	(1.33)	8	(7.38)
com_74	17	11	(1.00)	10	(1.40)	9	(1.89)	12	(3.08)	10	(1.30)	9	(2.11)
com_75	17	11	(1.00)	11	(1.45)	9	(1.89)	12	(2.33)	11	(5.82)	9	(1.67)
com_76	17	11	(1.00)	11	(1.36)	9	(2.11)	12	(2.25)	11	(6.00)	9	(2.11)
com_77	17	11	(2.36)	9	(1.56)	7	(1.29)	11	(2.36)	9	(1.56)	8	(1.62)
com_78	17	11	(1.00)	10	(1.50)	9	(2.11)	12	(3.08)	11	(3.82)	9	(2.22)
com_79	17	12	(1.08)	11	(1.36)	10	(10.40)	12	(2.33)	11	(9.18)	9	(1.33)
com_80	17	12	(1.25)	11	(1.36)	9	(2.00)	12	(2.33)	11	(7.36)	9	(1.78)
com_81	17	12	(5.92)	9	(1.44)	8	(3.38)	12	(6.33)	9	(1.22)	8	(2.38)
com_82	17	12	(1.00)	11	(2.00)	9	(2.56)	13	(2.92)	12	(3.58)	9	(2.56)
com_83	17	12	(1.00)	12	(1.42)	9	(1.56)	13	(2.23)	11	(1.09)	9	(1.33)
com_84	17	12	(1.00)	12	(1.42)	9	(1.67)	13	(2.23)	12	(7.33)	9	(1.56)
com_85	17	12	(2.50)	8	(1.38)	4	(2.25)	11	(3.18)	10	(8.40)	2	(3.50)
com_86	17	12	(1.33)	11	(2.27)	9	(2.11)	12	(3.25)	10	(1.40)	9	(3.11)
com_87	17	12	(1.33)	11	(2.27)	8	(2.75)	12	(2.25)	10	(1.30)	9	(3.00)
com_88	17	11	(1.09)	11	(1.73)	9	(2.00)	12	(2.25)	10	(1.20)	9	(1.78)
com_89	17	12	(2.83)	8	(1.62)	5	(5.80)	11	(2.82)	10	(9.00)	3	(11.00)
com_90	17	12	(1.42)	11	(1.45)	9	(1.56)	12	(2.50)	10	(1.30)	8	(1.88)
com_91	17	12	(1.25)	11	(1.91)	9	(2.00)	12	(2.00)	10	(1.00)	9	(2.11)
com_92	17	11	(1.09)	11	(1.55)	9	(1.44)	12	(2.08)	10	(1.10)	9	(1.44)
com_93	17	12	(2.67)	10	(1.70)	5	(4.80)	12	(5.17)	10	(8.40)	3	(7.00)
com_94	17	12	(1.00)	12	(1.58)	9	(1.67)	13	(2.38)	12	(5.25)	9	(2.00)
com_95	17	12	(1.00)	12	(1.75)	9	(2.67)	13	(1.92)	11	(1.00)	9	(2.89)
com_96	17	12	(1.08)	12	(1.50)	9	(1.44)	13	(2.00)	11	(1.00)	9	(1.44)

Table C.24: Performance of benchmark models for S^{rmsd} transformation match score on Comeau's data set.

	Clustering			Allowed	Transformation Matching		Ligand	Match	
Code	Pre	Intra	Post	Clashes	Displacement	Angle/RMSD	Interpolation	Score	N_{symm}
srpic_01	1.0	3.0	5.0	50	2.5	15.0	n	S^{da}	0
srpic_02	1.0	3.0	5.0	150	2.5	15.0	n	S^{da}	0
srpic_03	1.0	3.0	5.0	50	3.0	20.0	n	S^{da}	0
srpic_04	1.0	3.0	5.0	150	3.0	20.0	n	S^{da}	0
srpic_05	1.0	3.0	5.0	50	2.5	15.0	y	S^{da}	0
srpic_06	1.0	3.0	5.0	150	2.5	15.0	y	S^{da}	0
srpic_07	1.0	3.0	5.0	50	3.0	20.0	y	S^{da}	0
srpic_08	1.0	3.0	5.0	150	3.0	20.0	y	S^{da}	0
srpic_09	1.0	3.0	5.0	50	2.5	7.5	n	S^{rmsd}	0
srpic_10	1.0	3.0	5.0	150	2.5	7.5	n	S^{rmsd}	0
srpic_11	1.0	3.0	5.0	50	3.0	9.0	n	S^{rmsd}	0
srpic_12	1.0	3.0	5.0	150	3.0	9.0	n	S^{rmsd}	0
srpic_13	1.0	3.0	5.0	50	2.5	7.5	y	S^{rmsd}	0
srpic_14	1.0	3.0	5.0	150	2.5	7.5	y	S^{rmsd}	0
srpic_15	1.0	3.0	5.0	50	3.0	9.0	y	S^{rmsd}	0
srpic_16	1.0	3.0	5.0	150	3.0	9.0	y	S^{rmsd}	0
srpic_17	1.0	3.0	5.0	50	2.5	15.0	n	S^{da}	6
srpic_18	1.0	3.0	5.0	150	2.5	15.0	n	S^{da}	6
srpic_19	1.0	3.0	5.0	50	3.0	20.0	n	S^{da}	6
srpic_20	1.0	3.0	5.0	150	3.0	20.0	n	S^{da}	6
srpic_21	1.0	3.0	5.0	50	2.5	15.0	y	S^{da}	6
srpic_22	1.0	3.0	5.0	150	2.5	15.0	y	S^{da}	6
srpic_23	1.0	3.0	5.0	50	3.0	20.0	y	S^{da}	6
srpic_24	1.0	3.0	5.0	150	3.0	20.0	y	S^{da}	6
srpic_25	1.0	3.0	5.0	50	2.5	7.5	n	S^{rmsd}	6
srpic_26	1.0	3.0	5.0	150	2.5	7.5	n	S^{rmsd}	6
srpic_27	1.0	3.0	5.0	50	3.0	9.0	n	S^{rmsd}	6
srpic_28	1.0	3.0	5.0	150	3.0	9.0	n	S^{rmsd}	6
srpic_29	1.0	3.0	5.0	50	2.5	7.5	y	S^{rmsd}	6
srpic_30	1.0	3.0	5.0	150	2.5	7.5	y	S^{rmsd}	6
srpic_31	1.0	3.0	5.0	50	3.0	9.0	y	S^{rmsd}	6
srpic_32	1.0	3.0	5.0	150	3.0	9.0	y	S^{rmsd}	6

Table C.25: Parameters for SRPIC experiments. N_{symm} denotes the maximum complex size until which pre-ranking by the number of symmetry-optimizations is to be performed. Compared to benchmark runs, symmetric interface detection is additionally disabled and the lower bound on the number of solutions per level set to 250.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	$\leq$ 10		$\leq$ 5		$\leq$ 3		$\leq$ 2.5		$\leq$ 2.0		$\leq$ 1.0	
srpic_01	40	5	(1.20)	3	(1.00)	3	(1.00)	4	(18.75)	3	(1.00)	3	(1.00)
srpic_02	40	7	(32.86)	3	(1.00)	3	(1.00)	6	(42.50)	3	(1.00)	3	(1.00)
srpic_03	40	5	(1.40)	3	(1.33)	3	(1.33)	5	(5.80)	3	(1.33)	3	(2.33)
srpic_04	40	7	(9.57)	3	(1.67)	3	(1.67)	6	(12.00)	3	(1.67)	3	(2.67)
srpic_05	40	5	(1.00)	4	(1.00)	2	(1.00)	4	(1.00)	4	(1.00)	3	(1.00)
srpic_06	40	6	(1.33)	3	(1.00)	3	(2.00)	6	(1.33)	6	(1.33)	2	(1.00)
srpic_07	40	4	(1.00)	3	(1.00)	3	(1.33)	6	(25.50)	3	(1.00)	3	(3.33)
srpic_08	40	5	(1.20)	4	(1.25)	4	(2.50)	7	(2.71)	5	(1.20)	3	(3.00)
srpic_09	40	5	(1.00)	5	(1.00)	4	(1.00)	5	(1.00)	5	(1.00)	4	(1.00)
srpic_10	40	5	(1.00)	5	(1.00)	4	(1.00)	5	(1.00)	5	(1.00)	4	(1.00)
srpic_11	40	4	(1.00)	4	(1.00)	3	(2.00)	4	(1.00)	4	(1.00)	3	(2.00)
srpic_12	40	5	(4.60)	4	(1.00)	3	(2.67)	4	(1.00)	4	(1.00)	3	(2.67)
srpic_13	40	5	(1.00)	5	(1.20)	4	(2.50)	5	(1.00)	5	(1.00)	3	(1.67)
srpic_14	40	6	(2.00)	5	(1.20)	4	(2.25)	6	(12.50)	5	(1.00)	3	(2.67)
srpic_15	40	6	(13.67)	4	(1.00)	4	(2.25)	5	(1.00)	5	(1.00)	3	(1.33)
srpic_16	40	5	(2.20)	4	(1.00)	4	(2.00)	6	(24.50)	4	(1.00)	3	(2.33)
srpic_17	40	8	(1.00)	7	(1.29)	4	(1.25)	9	(28.78)	7	(1.00)	4	(2.00)
srpic_18	40	8	(1.00)	7	(1.43)	6	(2.00)	8	(1.12)	7	(1.14)	6	(2.17)
srpic_19	40	8	(1.25)	7	(1.43)	4	(2.50)	8	(27.12)	7	(1.14)	4	(1.75)
srpic_20	40	8	(1.25)	7	(1.29)	5	(2.20)	8	(1.38)	8	(1.50)	4	(2.25)
srpic_21	40	8	(1.00)	7	(1.43)	5	(3.60)	7	(1.43)	6	(1.00)	5	(1.00)
srpic_22	40	9	(1.00)	9	(1.11)	6	(2.00)	9	(1.11)	9	(1.11)	8	(1.50)
srpic_23	40	8	(1.00)	7	(1.00)	6	(1.83)	7	(1.00)	7	(1.00)	7	(9.00)
srpic_24	40	10	(3.00)	8	(1.00)	6	(4.50)	8	(1.00)	8	(1.00)	7	(1.00)
srpic_25	40	11	(2.09)	10	(1.00)	10	(1.70)	11	(1.64)	10	(1.00)	10	(1.40)
srpic_26	40	12	(16.17)	10	(1.00)	9	(2.00)	12	(16.17)	10	(1.00)	9	(1.67)
srpic_27	40	6	(1.00)	6	(1.00)	6	(1.50)	7	(2.71)	6	(1.00)	6	(1.00)
srpic_28	40	7	(23.00)	6	(1.00)	6	(1.67)	8	(36.12)	6	(1.00)	6	(1.00)
srpic_29	40	9	(1.00)	9	(1.00)	8	(1.62)	9	(1.00)	9	(1.00)	6	(1.50)
srpic_30	40	10	(1.00)	9	(1.00)	9	(1.44)	9	(1.00)	9	(1.00)	6	(1.17)
srpic_31	40	10	(1.00)	10	(1.00)	9	(1.33)	10	(1.00)	10	(1.00)	6	(1.83)
srpic_32	40	11	(5.64)	9	(1.00)	8	(1.38)	9	(1.00)	9	(1.00)	8	(1.62)
div2	40	14	(40.93)	13	(38.54)	12	(43.42)	14	(17.29)	13	(1.31)	11	(2.00)
div3	40	14	(33.86)	12	(18.92)	11	(31.27)	15	(52.00)	12	(1.08)	10	(1.30)
all	40	15	(190.87)	13	(86.46)	13	(241.92)	15	(180.87)	14	(21.14)	12	(2.17)

Table C.26: Performance of models for the SRPIC experiments for assemblies using docking poses with a constraint score of $\leq$ 1.0. First (second) half with disabled (enabled) symmetry pre-ranking.

	Clustering			Allowed	Transformation Matching		Ligand	Match	
Code	Pre	Intra	Post	Clashes	Displacement	Angle/RMSD	Interpolation	Score	N_{symm}
comb_01	1.0	3.0	5.0	50	2.5	15.0	n	S^{da}	0
comb_02	1.0	3.0	5.0	150	2.5	15.0	n	S^{da}	0
comb_03	1.0	3.0	5.0	50	3.0	20.0	n	S^{da}	0
comb_04	1.0	3.0	5.0	150	3.0	20.0	n	S^{da}	0
comb_05	1.0	3.0	5.0	50	2.5	15.0	y	S^{da}	0
comb_06	1.0	3.0	5.0	150	2.5	15.0	y	S^{da}	0
comb_07	1.0	3.0	5.0	50	3.0	20.0	y	S^{da}	0
comb_08	1.0	3.0	5.0	150	3.0	20.0	y	S^{da}	0
comb_09	1.0	3.0	5.0	50	2.5	7.5	n	S^{rmsd}	0
comb_10	1.0	3.0	5.0	150	2.5	7.5	n	S^{rmsd}	0
comb_11	1.0	3.0	5.0	50	3.0	9.0	n	S^{rmsd}	0
comb_12	1.0	3.0	5.0	150	3.0	9.0	n	S^{rmsd}	0
comb_13	1.0	3.0	5.0	50	2.5	7.5	y	S^{rmsd}	0
comb_14	1.0	3.0	5.0	150	2.5	7.5	y	S^{rmsd}	0
comb_15	1.0	3.0	5.0	50	3.0	9.0	y	S^{rmsd}	0
comb_16	1.0	3.0	5.0	150	3.0	9.0	y	S^{rmsd}	0
comb_17	1.0	3.0	5.0	50	2.5	15.0	n	S^{da}	6
comb_18	1.0	3.0	5.0	150	2.5	15.0	n	S^{da}	6
comb_19	1.0	3.0	5.0	50	3.0	20.0	n	S^{da}	6
comb_20	1.0	3.0	5.0	150	3.0	20.0	n	S^{da}	6
comb_21	1.0	3.0	5.0	50	2.5	15.0	y	S^{da}	6
comb_22	1.0	3.0	5.0	150	2.5	15.0	y	S^{da}	6
comb_23	1.0	3.0	5.0	50	3.0	20.0	y	S^{da}	6
comb_24	1.0	3.0	5.0	150	3.0	20.0	y	S^{da}	6
comb_25	1.0	3.0	5.0	50	2.5	7.5	n	S^{rmsd}	6
comb_26	1.0	3.0	5.0	150	2.5	7.5	n	S^{rmsd}	6
comb_27	1.0	3.0	5.0	50	3.0	9.0	n	S^{rmsd}	6
comb_28	1.0	3.0	5.0	150	3.0	9.0	n	S^{rmsd}	6
comb_29	1.0	3.0	5.0	50	2.5	7.5	y	S^{rmsd}	6
comb_30	1.0	3.0	5.0	150	2.5	7.5	y	S^{rmsd}	6
comb_31	1.0	3.0	5.0	50	3.0	9.0	y	S^{rmsd}	6
comb_32	1.0	3.0	5.0	150	3.0	9.0	y	S^{rmsd}	6

Table C.27: Parameters for CombDock experiments. N_{symm} denotes the maximum complex size until which pre-ranking by the number of symmetry-optimizations is to be performed. Compared to benchmark runs, symmetric interface detection is additionally disabled and the locking of interfaces is turned off, i.e., the algorithm considers the interfaces to be non-distinct. Furthermore, discarding of non-matching solutions once a matching solution in the current level has been found is disabled.

		C_αRMSD (avg. rank)						tRMSD (avg. rank)					
Code	N_{built}	$\leq$ 10		$\leq$ 5		$\leq$ 3		$\leq$ 2.5		$\leq$ 2.0		$\leq$ 1.0	
comb_01	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_02	727	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_03	727	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_04	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_05	727	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_06	727	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_07	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_08	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_09	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_10	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_11	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_12	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_13	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_14	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_15	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_16	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_17	728	1	(15.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_18	727	1	(4.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_19	727	1	(21.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_20	728	2	(3.50)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_21	727	1	(5.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_22	727	2	(10.50)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_23	728	1	(4.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_24	728	2	(7.50)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_25	728	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_26	728	2	(1.50)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_27	728	1	(27.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_28	728	2	(2.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_29	728	1	(8.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_30	728	2	(1.50)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_31	728	1	(22.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
comb_32	728	2	(1.50)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)
all	728	2	(1419.50)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)	0	(-1.00)

Table C.28: Performance on CombDock-generated docking poses using non-distinct interfaces.

BIBLIOGRAPHY

[1] W. C. Röntgen. Über eine neue Art von Strahlen. *Annalen der Physik*, 300(1):1–11, 1898.

[2] W. Friedrich, P. Knipping, and M. Laue. Interferenzerscheinungen bei Röntgenstrahlen. *Annalen der Physik*, 346(10):971–988, 1913.

[3] W. L. Bragg. The diffraction of short electromagnetic waves by a crystal. *Proceedings of the Cambridge Philosophical Society*, 17:43–57, 1913.

[4] L. Pauling. The principles determining the structure of complex ionic crystals. *Journal of the American Chemical Society*, 51(4):1010–1026, 1929.

[5] L. Pauling and R. B. Corey. Atomic coordinates and structure factors for two helical configurations of polypeptide chains. *Proceedings of the National Academy of Sciences of the United States of America*, 37(5):235, 1951.

[6] E. M. Purcell, H. C. Torrey, and R. V. Pound. Resonance absorption by nuclear magnetic moments in a solid. *Physical Review*, 69(1-2):37, 1946.

[7] F. Bloch. Nuclear induction. *Physical Review*, 70(7-8):460–474, 1946.

[8] J. D. Watson and F. H. C. Crick. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953.

[9] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips. A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature*, 181(4610):662–666, 1958.

[10] M. F. Perutz, H. Muirhead, J. M. Cox, L. C. G . Goaman, F. S. Mathews, E. L. McGandy, and L. E. Webb. Three-dimensional Fourier synthesis of horse oxyhaemoglobin at 2.8 Å resolution:(I) X-ray analysis. *Nature*, 219(5149):29–32, 1968.

[11] D. Ophir, S. Rankowitz, B. J. Shepherd, and R. J. Spinrad. Scientific applications: BRAD: the brookhaven raster display. *Communications of the ACM*, 11(6):415–416, 1968.

[12] H. M. Berman. The protein data bank: a historical perspective. *Acta Crystallographica Section A: Foundations of Crystallography*, 64(1):88–95, 2007.

[13] R. Brown. A brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies. *The Philosophical Magazine and Annals of Philosophy*, 4(21):161–173, 1828.

[14] A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 322(8):549–560, 1905.

[15] H. Hess. Thermochemische Untersuchungen. *Annalen der Physik*, 126(6):385–404, 1840.

[16] E. Fischer. Einfluss der Configuration auf die Wirkung der Enzyme. *Berichte der deutschen chemischen Gesellschaft*, 27(3):2985–2993, 1894.

[17] D. E. Koshland. Application of a theory of enzyme specificity to protein synthesis. *Proceedings of the National Academy of Sciences of the United States of America*, 44(2):98–104, 1958.

[18] B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. I. General method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.

[19] N. Metopolis. Equations of state calculations by fast computational machine. *Journal of Chemical Physics*, 21:1087–1097, 1953.

[20] L. Boltzmann. Über die Beziehung zwischen dem zweiten Hauptsatz der mechanischen Wärmetheorie und der Wahrscheinlichkeitsrechnung resp. den Sätzen über das Wärmegleichgewicht. *Akademie der Wissenschaften in Wien, Mathematisch-Naturwissenschaftliche Klasse, Sitzungsberichte, Abteilung IIa. Mathematik, Astronomie, Physik, Meteorologie und Technik*, 76:373–435, 1877.

[21] M. Planck. Über das Gesetz der Energieverteilung im Normalspectrum. *Annalen der Physik*, 309(3):553–563, 1901.

[22] S. Kirkpatrick, Gelatt C. D., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[23] J. H. Holland. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2):88–105, 1973.

[24] J. H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.

[25] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4(2):187–217, 1983.

[26] R. Nussinov. The significance of the 2013 Nobel Prize in chemistry and the challenges ahead. *PLoS Computational Biology*, 10(1):e1003423, 2014.

[27] W. M. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155(760):279–284, 1967.

[28] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[29] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[30] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin. A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology*, 161(2):269–288, 1982.

[31] M. Rarey, B. Kramer, T. Lengauer, and G. Klebe. A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology*, 261(3):470–89, 1996.

[32] C. Branden and J. Tooze. *Introduction to protein structure*, volume 2. Garland New York, 1991.

[33] J. M. Bujnicki. *Prediction of protein structures, functions, and interactions*. Wiley Online Library, 2009.

[34] D. J. Craik. Seamless proteins tie up their loose ends. *Science*, 311(5767):1563–1564, 2006.

[35] C. M. Dobson. Principles of protein folding, misfolding and aggregation. *Seminars in cell & developmental biology*, 15(1):3–16, 2004.

[36] P. E. Wright and H. J. Dyson. Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm. *Journal of molecular biology*, 293(2):321–331, 1999.

[37] H. J. Dyson and P. E. Wright. Intrinsically unstructured proteins and their functions. *Nature Reviews Molecular Cell Biology*, 6(3):197–208, 2005.

[38] P. Tompa and K.-H. Han. Intrinsically disordered proteins. *Physics Today*, 65(8):64–65, 2012.

[39] A. K. Dunker, J. D. Lawson, C. J. Brown, R. M. Williams, P. Romero, J. S. Oh, C. J. Oldfield, A. M. Campen, C. M. Ratliff, K. W. Hipps, J. Ausio, M. S. Nissen, R. Reeves, C. Kang, C. R. Kissinger, R. W. Bailey, M. D. Griswold, W. Chiu, E. C. Garner, and Z. Obradovic. Intrinsically disordered protein. *Journal of Molecular Graphics and Modelling*, 19(1):26–59, 2001.

[40] S. A. Hollingsworth and P. A. Karplus. A fresh look at the ramachandran plot and the occurrence of standard structures in proteins. *Biomolecular Concepts*, 1(3-4):271–283, 2010.

[41] J. S. Richardson. The anatomy and taxonomy of protein structure. In *Advances in Protein Chemistry*, volume 34, pages 167–339. Academic Press, 1981.

[42] S. Jones, M. Stewart, A. Michie, M. B. Swindells, C. Orengo, and J. M. Thornton. Domain assignment for protein structures using a consensus approach: characterization and analysis. *Protein Science*, 7(2):233–242, 1998.

[43] A. Andreeva, D. Howorth, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin. SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic acids research*, 32(suppl 1):D226–D229, 2004.

[44] C. Hadley and D. T. Jones. A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure*, 7(9):1099–1112, 1999.

[45] J. Monod, J.-P. Changeux, and F. Jacob. Allosteric proteins and cellular control systems. *Journal of molecular biology*, 6(4):306–329, 1963.

[46] A. Kentsis and K. L. B. Borden. Physical mechanisms and biological significance of supramolecular protein self-assembly. *Current Protein and Peptide Science*, 5(2):125–134, 2004.

[47] N. Brooijmans and I. D. Kuntz. Molecular recognition and docking algorithms. *Annual Review of Biophysics and Biomolecular Structure*, 32(1):335–373, 2003.

[48] H. Pearson. Genetics: what is a gene? *Nature*, 441(7092):398–401, 2006.

[49] S. R. Eddy. Non–coding RNA genes and the modern RNA world. *Nature Reviews Genetics*, 2(12):919–929, 2001.

[50] G. Orphanides, T. Lagrange, and D. Reinberg. The general transcription factors of RNA polymerase II. *Genes & Development*, 10(21):2657–2683, 1996.

[51] S. Hahn. Structure and mechanism of the RNA polymerase II transcription machinery. *Nature Structural & Molecular Biology*, 11(5):394–403, 2004.

[52] T. A. Steitz. A structural understanding of the dynamic ribosome machine. *Nature Reviews Molecular Cell Biology*, 9(3):242–253, 2008.

[53] T. M. Schmeing and V. Ramakrishnan. What recent ribosome structures have revealed about the mechanism of translation. *Nature*, 461(7268):1234–1242, 2009.

[54] D. N. Wilson and K. H. Nierhaus. Ribosomal proteins in the spotlight. *Critical Reviews in Biochemistry and Molecular Biology*, 40(5):243–267, 2005.

[55] J. Brosius. tRNAs in the spotlight during protein biosynthesis. *Trends in Biochemical Sciences*, 26(11):653–656, 2001.

[56] W. A. Decatur and M. J. Fournier. rRNA modifications and ribosome function. *Trends in Biochemical Sciences*, 27(7):344–351, 2002.

[57] J. Gumbart, C. Chipot, and K. Schulten. Free-energy cost for translocon-assisted insertion of membrane proteins. *Proceedings of the National Academy of Sciences*, 108(9):3596–3601, 2011.

[58] S. Shao and R. S. Hegde. Membrane protein insertion at the endoplasmic reticulum. *Annual review of cell and developmental biology*, 27:25–56, 2011.

[59] M. J. Gething. Protein folding in the cell. *Nature*, 355:33–45, 1992.

[60] I. Braakman and N. J. Bulleid. Protein folding and modification in the mammalian endoplasmic reticulum. *Annual review of biochemistry*, 80:71–99, 2011.

[61] F. U. Hartl, A. Bracher, and M. Hayer-Hartl. Molecular chaperones in protein folding and proteostasis. *Nature*, 475(7356):324–332, 2011.

[62] F. U. Hartl. Molecular chaperones in cellular protein-folding. *Nature*, 381(6583):571–580, 1996.

[63] B. Wilkinson and H. F. Gilbert. Protein disulfide isomerase. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1699(1):35–44, 2004.

[64] T. Kouzarides. Acetylation: a regulatory modification to rival phosphorylation? *The EMBO journal*, 19(6):1176–1179, 2000.

[65] J. Cieśla, T. Frączyk, and W. Rode. Phosphorylation of basic amino acid residues in proteins: important but easily missed. *Acta Biochimica Polonica*, 58:137–148, 2011.

[66] M. Karin and Y. Ben-Neriah. Phosphorylation meets ubiquitination: the control of NF-κB activity. *Annual Review of Immunology*, 18(1):621–663, 2000.

[67] D. G. Nicholls and S. Ferguson. *Bioenergetics*. Academic Press, 2013.

[68] A. Radzicka and R. Wolfenden. A proficient enzyme. *Science*, 267(5194):90–93, 1995.

[69] W. A. Eaton, E. R. Henry, J.s Hofrichter, and A. Mozzarelli. Is cooperative oxygen binding by hemoglobin really understood? *Nature Structural & Molecular Biology*, 6(4):351–358, 1999.

[70] S. Bleif, F. Hannemann, M. Lisurek, J. P. von Kries, J. Zapp, M. Dietzen, I. Antes, and R. Bernhardt. Identification of CYP106A2 as a regioselective allylic bacterial diterpene hydroxylase. *ChemBioChem*, 12(4):576–582, 2011.

[71] I. Rosenshine, M. S. Donnenberg, J. B. Kaper, and B. B. Finlay. Signal transduction between enteropathogenic escherichia coli (EPEC) and epithelial cells: EPEC induces tyrosine phosphorylation of host cell proteins to initiate cytoskeletal rearrangement and bacterial uptake. *The EMBO journal*, 11(10):3551–3560, 1992.

[72] D. M. Rosenbaum, S. G. F. Rasmussen, and B. K. Kobilka. The structure and function of G-protein-coupled receptors. *Nature*, 459(7245):356–363, 2009.

[73] G. L. Collingridge, R. W. Olsen, J. Peters, and M. Spedding. A nomenclature for ligand-gated ion channels. *Neuropharmacology*, 56(1):2–5, 2009.

[74] S. R. Hubbard and J. H. Till. Protein tyrosine kinase structure and function. *Annual Review of Biochemistry*, 69(1):373–398, 2000.

[75] A. J. Venkatakrishnan, X. Deupi, G. Lebon, C. G. Tate, G. F. Schertler, and M. M. Babu. Molecular signatures of G-protein-coupled receptors. *Nature*, 494(7436):185–194, 2013.

[76] R. Medzhitov. Recognition of microorganisms and activation of the immune response. *Nature*, 449(7164):819–826, 2007.

[77] R. Seger and E. G. Krebs. The MAPK signaling cascade. *The FASEB Journal*, 9(9):726–735, 1995.

[78] A. R. Saltiel and C. R. Kahn. Insulin signalling and the regulation of glucose and lipid metabolism. *Nature*, 414(6865):799–806, 2001.

[79] S. Seisenberger, J. R. Peat, T. A. Hore, F. Santos, W. Dean, and W. Reik. Reprogramming DNA methylation in the mammalian life cycle: building and breaking epigenetic barriers. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 368(1609):20110330, 2013.

[80] P. A. Jones and D. Takai. The role of DNA methylation in mammalian epigenetics. *Science*, 293(5532):1068–1070, 2001.

[81] I. Grummt and C. S. P. Epigenetic silencing of RNA polymerase I transcription. *Nature Reviews Molecular Cell Biology*, 4(8):641–649, 2003.

[82] C. V. Robinson, A. Šali, and W. Baumeister. The molecular sociology of the cell. *Nature*, 450(7172):973–982, 2007.

[83] E. D. Levy, J. B. Pereira-Leal, C. Chothia, and S. A. Teichmann. 3D complex: a structural classification of protein complexes. *PLoS Computational Biology*, 2(11):e155, 2006.

[84] Y. Liu, G. Fiskum, and D. Schubert. Generation of reactive oxygen species by the mitochondrial electron transport chain. *Journal of Neurochemistry*, 80(5):780–787, 2002.

[85] H. Schägger and K. Pfeiffer. Supercomplexes in the respiratory chains of yeast and mammalian mitochondria. *The EMBO journal*, 19(8):1777–1783, 2000.

[86] F. Hannemann, A. Bichet, K. M. Ewen, and R. Bernhardt. Cytochrome P450 systems – biological variations of electron transport chains. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1770(3):330–344, 2007.

[87] H. Eubel, J. Heinemeyer, S. Sunderhaus, and H.-P. Braun. Respiratory chain supercomplexes in plant mitochondria. *Plant Physiology and Biochemistry*, 42(12):937–942, 2004.

[88] H. Schägger. Respiratory chain supercomplexes. *IUBMB Life*, 52(3-5):119–128, 2001.

[89] A. R. Fernie, F. Carrari, and L. J. Sweetlove. Respiratory metabolism: glycolysis, the TCA cycle and mitochondrial electron transport. *Current Opinion in Plant Biology*, 7(3):254–261, 2004.

[90] G. K. Brown, L. J. Otero, M. LeGris, and R. M. Brown. Pyruvate dehydrogenase deficiency. *Journal of Medical Genetics*, 31(11):875, 1994.

[91] C. Marsac, D. François, F. Fouque, and C. Benelli. Pyruvate dehydrogenase deficiencies. In *Mitochondrial Diseases*, pages 173–184. Springer, 1999.

[92] T. Izard, A. Ævarsson, M. D. Allen, A. H. Westphal, R. N. Perham, A. de Kok, and W. G. J. Hol. Principles of quasi-equivalence and euclidean geometry govern the assembly of cubic and dodecahedral cores of pyruvate dehydrogenase complexes. *Proceedings of the National Academy of Sciences*, 96(4):1240–1245, 1999.

[93] J. L. S. Milne, X. Wu, M. J. Borgnia, J. S. Lengyel, B. R. Brooks, D. Shi, R. N. Perham, and S. Subramaniam. Molecular structure of a 9-MDa icosahedral pyruvate dehydrogenase subcomplex containing the E2 and E3 enzymes using cryoelectron microscopy. *Journal of Biological Chemistry*, 281(7):4364–4370, 2006.

[94] M. Smolle, A. E. Prior, A. E. Brown, A. Cooper, O. Byron, and J. G. Lindsay. A new level of architectural complexity in the human pyruvate dehydrogenase complex. *Journal of Biological Chemistry*, 281(28):19772–19780, 2006.

[95] Z. H. Zhou, D. B. McCarthy, C. M. O'Connor, L. J. Reed, and J. K. Stoops. The remarkable structural and functional organization of the eukaryotic pyruvate dehydrogenase complexes. *Proceedings of the National Academy of Sciences*, 98(26):14802–14807, 2001.

[96] Z. H. Zhou, W. Liao, R. H. Cheng, J. E. Lawson, D. B. McCarthy, L. J. Reed, and J. K. Stoops. Direct evidence for the size and conformational variability of the pyruvate dehydrogenase complex revealed by three-dimensional electron microscopy the "breathing" core and its functional relationship to protein dynamics. *Journal of Biological Chemistry*, 276(24):21704–21713, 2001.

[97] S. Fang and A. M. Weissman. Ubiquitin-proteasome system. *Cellular and Molecular Life Sciences*, 61(13):1546–1561, 2004.

[98] K. Haglund and I. Dikic. Ubiquitylation and cell signaling. *The EMBO journal*, 24(19):3353–3359, 2005.

[99] J. Adams. Proteasome inhibition: a novel approach to cancer therapy. *Trends in Molecular Medicine*, 8(4 Suppl):S49–S54, 2002.

[100] N. Yamamoto, H. Sawada, Y. Izumi, T. Kume, H. Katsuki, S. Shimohama, and A. Akaike. Proteasome inhibition induces glutathione synthesis and protects cells from oxidative stress: relevance to Parkinson disease. *Journal of Biological Chemistry*, 282(7):4364–4372, 2007.

[101] J. Adams, V. J. Palombella, E. A. Sausville, J. Johnson, A. Destree, D. D. Lazarus, J. Maas, C. S. Pien, S. Prakash, and P. J. Elliott. Proteasome inhibitors: a novel class of potent and effective antitumor agents. *Cancer Research*, 59(11):2615–2622, 1999.

[102] J. Adams. The development of proteasome inhibitors as anticancer drugs. *Cancer Cell*, 5(5):417–421, 2004.

[103] L. R. Dick and P. E. Fleming. Building on bortezomib: second-generation proteasome inhibitors as anti-cancer therapy. *Drug Discovery Today*, 15(5):243–249, 2010.

[104] J. Löwe, D. Stock, B. Jap, P. Zwickl, W. Baumeister, and R. Huber. Crystal structure of the 20S proteasome from the archaeon T. acidophilum at 3.4Å resolution. *Science*, 268(5210):533–539, 1995.

[105] M. Groll, L. Ditzel, J. Löwe, D. Stock, M. Bochtler, H. D. Bartunik, and R. Huber. Structure of 20S proteasome from yeast at 2.4Å resolution. *Nature*, 386(6624):463, 1997.

[106] A. Förster, E. I. Masters, F. G. Whitby, H. Robinson, and C. P. Hill. The 1.9Å structure of a proteasome-11S activator complex and implications for proteasome-PAN/PA700 interactions. *Molecular Cell*, 18(5):589–599, 2005.

[107] F. Kopp, R. Steiner, B. Dahlmann, L. Kuehn, and H. Reinauer. Size and shape of the multicatalytic proteinase from rat skeletal muscle. *Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular Enzymology*, 872(3):253–260, 1986.

[108] F. Förster, K. Lasker, S. Nickell, A. Šali, and W. Baumeister. Toward an integrated structural model of the 26S proteasome. *Molecular & Cellular Proteomics*, 9(8):1666–1677, 2010.

[109] K. Lasker, F. Förster, S. Bohn, T. Walzthoeni, E. Villa, P. Unverdorben, F. Beck, R. Aebersold, A. Šali, and W. Baumeister. Molecular architecture of the 26S proteasome holocomplex determined by an integrative approach. *Proceedings of the National Academy of Sciences*, 109(5):1380–1387, 2012.

[110] L. Bedford, S. Paine, P. W. Sheppard, R. J. Mayer, and J. Roelofs. Assembly, structure, and function of the 26S proteasome. *Trends in Cell Biology*, 20(7):391–401, 2010.

[111] D. Moreira and P. López-García. Ten reasons to exclude viruses from the tree of life. *Nature Reviews Microbiology*, 7(4):306–311, 2009.

[112] John J. R. and Roger M. B. Spherical viruses. *Current Opinion in Structural Biology*, 8(2):142–149, 1998.

[113] R. V. Mannige and C. L. Brooks III. Periodic table of virus capsids: implications for natural selection and design. *PloS One*, 5(3):e9423, 2010.

[114] J. Watson and F. Crick. Structure of small viruses. *Nature*, 177(4506):473–475, 1956.

[115] R. A. Crowther and L. A. Amos. Three-dimensional image reconstructions of some small spherical viruses. *Cold Spring Harbor Symposia on Quantitative Biology*, 36:489–494, 1972.

[116] D. L. D. Caspar and A. Klug. Physical principles in the construction of regular viruses. *Cold Spring Harbor Symposia on Quantitative Biology*, 27:1–24, 1962.

[117] R. W. Horne and P. Wildy. Symmetry in virus architecture. *Virology*, 15(3):348–373, 1961.

[118] B. V. V. Prasad and M. F. Schmid. Principles of virus structural organization. In *Viral Molecular Machines*, pages 17–47. Springer, 2012.

[119] S. Cheng and C. L. Brooks III. Viral capsid proteins are segregated in structural fold space. *PLoS Computational Biology*, 9(2):e1002905, 2013.

[120] G. Zhao, J. R. Perilla, E. L. Yufenyuy, X. Meng, B. Chen, J. Ning, J. Ahn, A. M. Gronenborn, K. Schulten, C. Aiken, and P. Zhang. Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. *Nature*, 497(7451):643–646, 2013.

[121] L. K. Medina-Kauwe. Development of adenovirus capsid proteins for targeted therapeutic delivery. *Therapeutic Delivery*, 4(2):267–277, 2013.

[122] N. Uchida, M. M. Hsieh, K. N. Washington, and J. F. Tisdale. Efficient transduction of human hematopoietic repopulating cells with a chimeric HIV1-based vector including SIV capsid. *Experimental Hematology*, 41(9):779–788, 2013.

[123] T. L. Blundell and L. N. Johnson. *Crystallization of proteins*, pages 59–82. Academic Press, 1976.

[124] D. E. McRee. *Practical Protein Crystallography*. Academic Press, 1993.

[125] J. Mesters. *Principles of protein X-ray crystallography*. Springer, 2007.

[126] V. Cherezov, D. M. Rosenbaum, M. A. Hanson, S. G. F. Rasmussen, F. S. Thian, T. S. Kobilka, H.-J. Choi, P. Kuhn, W. I. Weis, B. K. Kobilka, and R. J. Stevens. High-resolution crystal structure of an engineered human β2-adrenergic G protein-coupled receptor. *science*, 318(5854):1258–1265, 2007.

[127] T. Warne, M. J. S.-V., J. G. Baker, R. Moukhametzianov, P. C. Edwards, R. Henderson, A. G. W. Leslie, C. G. Tate, and G. F. X. Schertler. Structure of a β_1-adrenergic G-protein-coupled receptor. *Nature*, 454(7203):486–491, 2008.

[128] V.-P. Jaakola, M. T. Griffith, M. A. Hanson, V. Cherezov, E. Y. T. Chien, J. R. Lane, A. P. Ijzerman, and R. C. Stevens. The 2.6 Angstrom crystal structure of a human A2A adenosine receptor bound to an antagonist. *Science*, 322(5905):1211–1217, 2008.

[129] K. Wüthrich. The way to NMR structures of proteins. *Nature structural biology*, 8(11):923–925, 2001.

[130] W. Rieping, M. Habeck, and M. Nilges. Inferential structure determination. *Science*, 309(5732):303–306, 2005.

[131] A. H. Kwan, M. Mobli, P. R. Gooley, G. F. King, and J. P. Mackay. Macromolecular NMR spectroscopy for the non-spectroscopist. *FEBS journal*, 278(5):687–703, 2011.

[132] K. Pervushin, R. Riek, G. Wider, and K. Wüthrich. Attenuated t_2 relaxation by mutual cancellation of dipole–dipole coupling and chemical shift anisotropy indicates an avenue to nmr structures of very large biological macromolecules in solution. *Proceedings of the National Academy of Sciences*, 94(23):12366–12371, 1997.

[133] Y. Xu and S. Matthews. TROSY NMR spectroscopy of large soluble proteins. In *Modern NMR Methodology*, pages 97–119. Springer, 2013.

[134] J. Fiaux, E. B. Bertelsen, A. L. Horwich, and K. Wüthrich. NMR analysis of a 900k GroEL–GroES complex. *Nature*, 418(6894):207–211, 2002.

[135] J. Frank. Single-particle imaging of macromolecules by cryo-electron microscopy. *Annual review of Biophysics and Biomolecular Structure*, 31(1):303–319, 2002.

[136] M. Topf, K. Lasker, B. Webb, H. Wolfson, W. Chiu, and A. Šali. Protein structure fitting and refinement guided by cryo-EM density. *Structure*, 16(2):295–307, 2008.

[137] V. Lucic, F. Förster, and W. Baumeister. Structural studies by electron tomography: from cells to molecules. *Annual Review of Biochemistry*, 74:833–865, 2005.

[138] O. Medalia, I. Weber, A. S. Frangakis, D. Nicastro, G. Gerisch, and W. Baumeister. Macromolecular architecture in eukaryotic cells visualized by cryoelectron tomography. *Science*, 298(5596):1209–1213, 2002.

[139] K. Grünewald and M. Cyrklaff. Structure of complex viruses and virus-infected cells by electron cryo tomography. *Current Opinion in Microbiology*, 9(4):437–442, 2006.

[140] W. Chiu, M. L. Baker, W. Jiang, M. Dougherty, and M. F. Schmid. Electron cryomicroscopy of biological machines at subnanometer resolution. *Structure*, 13(3):363–372, 2005.

[141] F. Fabiola and M. S. Chapman. Fitting of high-resolution structures into electron microscopy reconstruction images. *Structure*, 13(3):389–400, 2005.

[142] F. Alber, N. Eswar, and A. Sali. Structure determination of macromolecular complexes by experiment and computation. In *Practical Bioinformatics*, pages 73–96. Springer, 2004.

[143] W. Wriggers and S. Birmanns. Using situs for flexible and rigid-body fitting of multiresolution single-molecule data. *Journal of structural biology*, 133(2):193–202, 2001.

[144] F. Tama, O. Miyashita, and C. L. Brooks III. Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-EM. *Journal of structural biology*, 147(3):315–326, 2004.

[145] M. Delarue and P. Dumas. On the use of low-frequency normal modes to enforce collective movements in refining macromolecular structural models. *Proceedings of the National Academy of Sciences of the United States of America*, 101(18):6957–6962, 2004.

[146] A. E. Todd, C. A. Orengo, and J. M. Thornton. Evolution of function in protein superfamilies, from a structural perspective. *Journal of molecular biology*, 307(4):1113–1143, 2001.

[147] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247(4):536–540, 1995.

[148] I. Sillitoe, A. L. Cuff, B. H. Dessailly, N. L. Dawson, N. Furnham, D. Lee, J. G. Lees, T. E. Lewis, R. A. Studer, R. Rentzsch, C. Yeats, J. M. Thornton, and C. A. Orengo. New functional families (FunFams) in CATH to improve the mapping of conserved functional sites to 3D structures. *Nucleic Acids Research*, 41(D1):D490–D498, 2013.

[149] T. J. P. Hubbard, B. Ailey, S. E. Brenner, A. G. Murzin, and C. Chothia. Scop, structural classification of proteins database: applications to evaluation of the effectiveness of sequence alignment methods and statistics of protein structural data. *Acta Crystallographica Section D: Biological Crystallography*, 54(6):1147–1154, 1998.

[150] A. Andreeva, D. Howorth, C. Chothia, E. Kulesha, and A. G. Murzin. SCOP2 prototype: a new approach to protein structure mining. *Nucleic acids research*, 42(D1):D310–D314, 2014.

[151] O. C. Redfern, A. Harrison, T. Dallman, F. M. G. Pearl, and C. A. Orengo. CATHEDRAL: a fast and effective algorithm to predict folds and domain boundaries from multidomain protein structures. *PLoS Computational Biology*, 3(11):e232, 2007.

[152] I. Sillitoe, M. Dibley, J. Bray, S. Addou, and C. A. Orengo. Assessing strategies for improved superfamily recognition. *Protein science*, 14(7):1800–1810, 2005.

[153] T. E. Lewis, I. Sillitoe, A. Andreeva, T. L. Blundell, D. W. A. Buchan, C. Chothia, A. Cuff, J. M. Dana, I. Filippis, J. Gough, S. Hunter, D. T. Jones, L. A. Kelley, G. J. Kleywegt, F. Minneci, A. Mitchell, A. G. Murzin, B. Ochoa-Montaño, O. J. Rackham, J. Smith, M. J. Sternberg, S. Velankar, C. Yeats, and C. Orengo. Genome3D: a UK collaborative project to annotate genomic sequences with predicted 3D structures based on SCOP and CATH domains. *Nucleic acids research*, 41(D1):D499–D507, 2013.

[154] D. A. de Lima Morais, H. Fang, O. J. L. Rackham, D. Wilson, R. Pethica, C. Chothia, and J. Gough. SUPERFAMILY 1.75 including a domain-centric gene ontology method. *Nucleic acids research*, 39(suppl 1):D427–D434, 2011.

[155] S. R. Eddy. Hidden Markov Models. *Current Opinion in Structural Biology*, 6(3):361–365, 1996.

[156] L. Holm and C. Sander. The FSSP database: fold classification based on structure-structure alignment of proteins. *Nucleic Acids Research*, 24(1):206–209, 1996.

[157] R. Wang, L. Lai, and S. Wang. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of Computer-Aided Molecular Design*, 16(1):11–26, 2002.

[158] J. Janin. *Docking Predictions of Protein-Protein Interactions and Their Assessment: The CAPRI Experiment*, pages 87–104. Springer, 2013.

[159] S.-Y. Huang and X. Zou. Advances and challenges in protein-ligand docking. *International journal of molecular sciences*, 11(8):3016–3034, 2010.

[160] M. F. Lensink and S. J. Wodak. Docking, scoring, and affinity prediction in CAPRI. *Proteins: Structure, Function, and Bioinformatics*, 81(12):2082–2095, 2013.

[161] A. N. Jain. Scoring functions for protein-ligand docking. *Current Protein and Peptide Science*, 7(5):407–420, 2006.

[162] N. Moitessier, P. Englebienne, D. Lee, J. Lawandi, and C. R. Corbeil. Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. *British Journal of Pharmacology*, 153(S1):S7–S26, 2008.

[163] M. Kontoyianni, L. M. McClellan, and G. S. Sokol. Evaluation of docking performance: comparative data on docking algorithms. *Journal of Medicinal Chemistry*, 47(3):558–565, 2004.

[164] E. Kellenberger, J. Rodrigo, P. Muller, and D. Rognan. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. *Proteins: Structure, Function, and Bioinformatics*, 57(2):225–242, 2004.

[165] E. Perola, W. P. Walters, and P. S. Charifson. A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. *Proteins: Structure, Function, and Bioinformatics*, 56(2):235–249, 2004.

[166] M. D. Cummings, R. L. DesJarlais, A. C. Gibbs, V. Mohan, and E. P. Jaeger. Comparison of automated docking programs as virtual screening tools. *Journal of Medicinal Chemistry*, 48(4):962–976, 2005.

[167] G. L. Warren, C. W. Andrews, A.-M. Capelli, B. Clarke, J. LaLonde, M. H. Lambert, M. Lindvall, N. Nevins, S. F. Semus, S. Senger, G. Tedesco, I. D. Wall, J. M. Woolven, Peishoff C. E., and M. S. Head. A critical assessment of docking programs and scoring functions. *Journal of Medicinal Chemistry*, 49(20):5912–5931, 2006.

[168] J. B. Cross, D. C. Thompson, B. K. Rai, J. C. Baber, K. Y. Fan, Y. Hu, and C. Humblet. Comparison of several molecular docking programs: pose prediction and virtual screening accuracy. *Journal of Chemical Information and Modeling*, 49(6):1455–1474, 2009.

[169] H. Claussen, C. Buning, M. Rarey, and T. Lengauer. FlexE: efficient molecular docking considering protein structure variations. *Journal of Molecular Biology*, 308(2):377–95, 2001.

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

[171] H.-J. Böhm. The computer program LUDI: a new method for the de novo design of enzyme inhibitors. *Journal of computer-aided molecular design*, 6(1):61–78, 1992.

[172] H.-J. Böhm. LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads. *Journal of Computer-Aided Molecular Design*, 6(6):593–606, 1992.

[173] G. Klebe and T. Mietzner. A fast and efficient method to generate biologically relevant conformations. *Journal of Computer-Aided Molecular Design*, 8(5):583–606, 1994.

[174] G. Jones. Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, 267(3):727–748, 1997.

[175] D. S. Goodsell and A. J. Olson. Automated docking of substrates to proteins by simulated annealing. *Proteins: Structure, Function, and Bioinformatics*, 8(3):195–202, 1990.

[176] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry*, 19(14):1639–1662, 1998.

[177] F. Osterberg, G. M. Morris, M. F. Sanner, A. J. Olson, and D. S. Goodsell. Automated docking to multiple target structures: incorporation of protein mobility and structural water heterogeneity in AutoDock. *Proteins*, 46(1):34–40, 2002.

[178] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry*, 30(16):2785–2791, 2009.

[179] J. Fuhrmann, A. Rurainski, H.-P. Lenhof, and D. Neumann. A new Lamarckian genetic algorithm for flexible ligand-receptor docking. *Journal of Computational Chemistry*, 31(9):1911–1918, 2010.

[180] J. J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl, and D. Baker. Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of Molecular Biology*, 331(1):281–299, 2003.

[181] S. Chaudhury, M. Berrondo, B. D. Weitzner, P. Muthu, H. Bergman, and J. J. Gray. Benchmarking and analysis of protein docking performance in Rosetta v3.2. *PLoS One*, 6(8):e22477, 2011.

[182] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 2004.

[183] M. D. Daily, D. Masica, A. Sivasubramanian, S. Somarouthu, and J. J. Gray. CAPRI rounds 3–5 reveal promising successes and future challenges for RosettaDock. *Proteins: Structure, Function, and Bioinformatics*, 60(2):181–186, 2005.

[184] Y. Inbar, H. Benyamini, R. Nussinov, and H. J. Wolfson. Prediction of multimolecular assemblies by multiple docking. *Journal of Molecular Biology*, 349(2):435–447, 2005.

[185] Y. Inbar, H. Benyamini, R. Nussinov, and H. J Wolfson. Combinatorial docking approach for structure prediction of large proteins and multi-molecular assemblies. *Physical biology*, 2(4):S156–S165, 2005.

[186] R. Norel, S. L. Lin, H. J. Wolfson, and R. Nussinov. Molecular surface complementarity at protein-protein interfaces: the critical role played by surface normals at well placed, sparse, points in docking. *Journal of Molecular Biology*, 252(2):263–273, 1995.

[187] V. Polak. Budda: Backbone unbound docking application. Master's thesis, School of Computer Science, Tel–Aviv University, 2003.

[188] D. Duhovny, R. Nussinov, and H. J. Wolfson. Efficient unbound docking of rigid molecules. In *Algorithms in Bioinformatics*, pages 185–200. Springer, 2002.

[189] D. Schneidman-Duhovny, Y. Inbar, Ru. Nussinov, and H. J. Wolfson. PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucleic Acids Research*, 33(Web Server Issue):W363–W367, 2005.

[190] C. Dominguez, R. Boelens, and A. M. J. J. Bonvin. HADDOCK: A protein-protein docking approach based on biochemical or biophysical information. *Journal of the American Chemical Society*, 125(7):1731–1737, 2003.

[191] E. Karaca, A. S. J. Melquiond, S. J. de Vries, P. L. Kastritis, and A. M. J. J. Bonvin. Building macromolecular assemblies by information-driven docking. Introducing the HADDOCK multibody docking server. *Molecular & Cellular Proteomics*, 9(8):1784–1794, 2010.

[192] A. D. McLachlan. Gene duplications in the structural evolution of chymotrypsin. *Journal of Molecular Biology*, 128(1):49–79, 1979.

[193] S. R. Comeau and C. J. Camacho. Predicting oligomeric assemblies: N-mers a primer. *Journal of Structural Biology*, 150(3):233–244, 2005.

[194] J. G. Mandell, M. E. Roberts, V. A.and Pique, V. Kotlovyi, J. C. Mitchell, E. Nelson, I. Tsigelny, and L. F. Ten Eyck. Protein docking using continuum electrostatics and geometric fit. *Protein Engineering*, 14(2):105–113, 2001.

[195] C. J. Camacho, D. W. Gatchell, S. R. Kimura, and S. Vajda. Scoring docked conformations generated by rigid-body protein-protein docking. *Proteins: Structure, Function, and Bioinformatics*, 40(3):525–537, 2000.

[196] S. R. Comeau, D. W. Gatchell, S. Vajda, and C. J. Camacho. ClusPro: an automated docking and discrimination method for the prediction of protein complexes. *Bioinformatics*, 20(1):45–50, 2004.

[197] D. S. Marks, T. A. Hopf, and C. Sander. Protein structure prediction from sequence variation. *Nature Biotechnology*, 30(11):1072–1080, 2012.

[198] T. A. Hopf, L. J. Colwell, R. Sheridan, B. Rost, C. Sander, and D. S. Marks. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, 149(7):1607–1621, 2012.

[199] I. Sandler, O. Medalia, and A. Aharoni. Experimental analysis of co-evolution within protein complexes: the yeast exosome as a model. *Proteins: Structure, Function, and Bioinformatics*, 81(11):1997–2006, 2013.

[200] G. W. Clark, A. Bezginov, J. M. Yang, R. L. Charlebois, and E. R. M. Tillier. *Using Coevolution to Predict Protein-Protein Interactions*, pages 237–256. Springer, 2011.

[201] A. Fiser, R. K. G. Do, and A. Šali. Modeling of loops in protein structures. *Protein Science*, 9(9):1753–1773, 2000.

[202] N. Eswar, B. Webb, M. A. Marti-Renom, M. S. Madhusudhan, D. Eramian, M.-Y. Shen, U. Pieper, and A. Šali. Comparative protein structure modeling using Modeller. *Current Protocols in Bioinformatics*, pages 5–6, 2006.

[203] D. C. Rapaport. *The art of molecular dynamics simulation*. Cambridge University Press, 2004.

[204] G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins*, 77(4):778–795, 2009.

[205] A. A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Science*, 12(9):2001–14, 2003.

[206] R. F. Goldstein. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophysical Journal*, 66(5):1335–1340, 1994.

[207] J. Xu. Rapid protein side-chain packing via tree decomposition. In *Research in computational molecular biology*, pages 423–439. Springer, 2005.

[208] C. Hartmann, I. Antes, and T. Lengauer. IRECS: a new algorithm for the selection of most probable ensembles of side-chain conformations in protein models. *Protein Science*, 16(7):1294–1307, 2007.

[209] B. R. Brooks, C. L. Brooks, A. D. MacKerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus. CHARMM: the biomolecular simulation program. *Journal of Computational Chemistry*, 30(10):1545–1614, 2009.

[210] C. Hartmann, I. Antes, and T. Lengauer. Docking and scoring with alternative side-chain conformations. *Proteins: Structure, Function, and Bioinformatics*, 74(3):712–726, 2009.

[211] H. M. Berman, T. Battistuz, T. N. Bhat, W. F. Bluhm, P. E. Bourne, K. Burkhardt, Z. Feng, G. L. Gilliland, L. Iype, S. Jain, P. Fagan, J. Marvin, D. Padilla, V. Ravichandran, B. Schneider, N. Thanki, H. Weissig, J. D. Westbrook, and C. Zardecki. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

[212] F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer Jr, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, 112(3):535–542, 1977.

[213] http://www.rcsb.org/pdb/statistics/holdings.do, 2014.

[214] http://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=total, 2014.

[215] http://www.3dcomplex.org/Hierarchy.cgi, 2014.

[216] E. D. Levy. Piqsi: protein quaternary structure investigation. *Structure*, 15(11):1364–1367, 2007.

[217] M. Gerstein and W. Krebs. A database of macromolecular motions. *Nucleic Acids Research*, 26(18):4280–4290, 1998.

[218] R. Mosca, A. Céol, and P. Aloy. Interactome3D: adding structural details to protein networks. *Nature Methods*, 10(1):47–53, 2012.

[219] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, L. J. Jensen, and C. von Mering. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39(Database Issue):D561–D568, 2011.

[220] P. Aloy, H. Ceulemans, A. Stark, and R. B. Russell. The relationship between sequence and interaction divergence in proteins. *Journal of Molecular Biology*, 332(5):989–998, 2003.

[221] T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[222] N. M. Goodey and S. J. Benkovic. Allosteric regulation and catalysis emerge via a common route. *Nature Chemical Biology*, 4(8):474–482, 2008.

[223] P. W. Schenk and B. E. Snaar-Jagalska. Signal perception and transduction: the role of protein kinases. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, 1449(1):1–24, 1999.

[224] M. R. Arkin and A. Whitty. The road less traveled: modulating signal transduction enzymes by inhibiting their protein-protein interactions. *Current Opinion in Chemical Biology*, 13(3):284–290, 2009.

[225] E. C. Chang, T. H. Charn, S.-H. Park, W. G. Helferich, B. Komm, J. A. Katzenellenbogen, and B. S. Katzenellenbogen. Estrogen receptors α and β as determinants of gene expression: influence of ligand, dose, and chromatin binding. *Molecular Endocrinology*, 22(5):1032–1043, 2008.

[226] B. K. Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, 2004.

[227] C. Sotriffer, R. Mannhold, H. Kubinyi, and G. Folkers. *Virtual Screening: Principles, Challenges, and Practical Guidelines*. Methods and Principles in Medicinal Chemistry. John Wiley & Sons, 2011.

[228] V. Zoete, A. Grosdidier, and O. Michielin. Docking, virtual high throughput screening and in silico fragment-based drug design. *Journal of Cellular and Molecular Medicine*, 13(2):238–248, 2009.

[229] C.-J. Tsai, B. Ma, and R. Nussinov. Folding and binding cascades: shifts in energy landscapes. *Proceedings of the National Academy of Sciences*, 96(18):9970–9972, 1999.

[230] K.-I. Okazaki and S. Takada. Dynamic energy landscape view of coupled binding and protein conformational change: induced-fit versus population-shift mechanisms. *Proceedings of the National Academy of Sciences*, 105(32):11182–11187, 2008.

[231] H. X. Kondo, N. Okimoto, G. Morimoto, and M. Taiji. Free-energy landscapes of protein domain movements upon ligand binding. *The Journal of Physical Chemistry B*, 115(23):7629–7636, 2011.

[232] R. Abagyan, M. Totrov, and D. Kuznetsov. ICM – a new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation. *Journal of Computational Chemistry*, 15(5):488–506, 1994.

[233] C. M. Venkatachalam, X. Jiang, T. Oldfield, and M. Waldman. LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites. *Journal of Molecular Graphics and Modeling*, 21(4):289–307, 2003.

[234] D. S. Goodsell, G. M. Morris, and A. J. Olson. Automated docking of flexible ligands: applications of AutoDock. *Journal of Molecular Recognition*, 9(1):1–5, 1996.

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

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

[237] A. R. Leach. Ligand docking to proteins with discrete side-chain flexibility. *Journal of Molecular Biology*, 235(1):345–56, 1994.

[238] J. Fernández-Recio, M. Totrov, and R. Abagyan. ICM-DISCO docking by global energy optimization with fully flexible side-chains. *Proteins*, 52(1):113–117, 2003.

[239] J. Meiler and D. Baker. ROSETTALIGAND: protein-small molecule docking with full side-chain flexibility. *Proteins*, 65(3):538–48, 2006.

[240] P. Källblad and P. M. Dean. Efficient conformational sampling of local side-chain flexibility. *Journal of Molecular Biology*, 326(5):1651–65, 2003.

[241] M. I. Zavodszky and L. A. Kuhn. Side-chain flexibility in protein-ligand binding: the minimal rotation hypothesis. *Protein Science*, 14(4):1104–1114, 2005.

[242] T. M. Frimurer, G. H. Peters, L. F. Iversen, H. S. Andersen, N. P. Møller, and O. H. Olsen. Ligand-induced conformational changes: improved predictions of ligand binding conformations and affinities. *Biophysical Journal*, 84(4):2273–2281, 2003.

[243] M. Zacharias. ATTRACT: protein-protein docking in CAPRI using a reduced protein model. *Proteins*, 60(2):252–6, 2005.

[244] A. May and M. Zacharias. Protein-protein docking in CAPRI using ATTRACT to account for global and local flexibility. *Proteins*, 69(4):774–780, 2007.

[245] R. M. Knegtel, I. D. Kuntz, and C. M. Oshiro. Molecular docking to ensembles of protein structures. *Journal of Molecular Biology*, 266(2):424–440, 1997.

[246] B. Q. Wei, L. H. Weaver, A. M. Ferrari, B. W. Matthews, and B. K. Shoichet. Testing a flexible-receptor docking algorithm in a model binding site. *Journal of Molecular Biology*, 337(5):1161–1182, 2004.

[247] G. Bottegoni, I. Kufareva, M. Totrov, and R. Abagyan. Four-dimensional docking: a fast and accurate account of discrete receptor flexibility in ligand docking. *Journal of Medicinal Chemistry*, 52(2):397–406, 2009.

[248] D. M. Lorber and B. K. Shoichet. Flexible ligand docking using conformational ensembles. *Protein Science*, 7(4):938–950, 1998.

[249] Y. Zhao and M. F. Sanner. FLIPDock: docking flexible ligands into flexible receptors. *Proteins*, 68(3):726–737, 2007.

[250] A. M. Ferrari, B. Q. Wei, L. Costantino, and B. K. Shoichet. Soft docking and multiple receptor conformations in virtual screening. *Journal of Medicinal Chemistry*, 47(21):5076–5084, 2004.

[251] V. Hornak, A. Okur, R. C. Rizzo, and C. Simmerling. HIV-1 protease flaps spontaneously open and reclose in molecular dynamics simulations. *Proceedings of the National Academy of Sciences of the United States of America*, 103(4):915–920, 2006.

[252] C. A. Sotriffer, O. Krämer, and G. Klebe. Probing flexibility and "induced-fit" phenomena in aldose reductase by comparative crystal structure analysis and molecular dynamics simulations. *Proteins: Structure, Function, and Bioinformatics*, 56(1):52–66, 2004.

[253] A. Amadei, A. B. M. Linssen, and H. J. C. Berendsen. Essential dynamics of proteins. *Proteins: Structure, Function, and Bioinformatics*, 17(4):412–425, 1993.

[254] M. A. Balsera, W. Wriggers, Y. Oono, and K. Schulten. Principal component analysis and long time protein dynamics. *The Journal of Physical Chemistry*, 100(7):2567–2572, 1996.

[255] M. Zacharias. Rapid protein-ligand docking using soft modes from molecular dynamics simulations to account for protein deformability: binding of FK506 to FKBP. *Proteins*, 54(4):759–767, 2004.

[256] G. Smith, M. Sternberg, and P. Bates. The relationship between the flexibility of proteins and their conformational states on forming protein-protein complexes with an application to protein-protein docking. *Journal of Molecular Biology*, 347(5):1077–1101, 2005.

[257] D. A. Case. Normal mode analysis of protein dynamics. *Current Opinion in Structural Biology*, 4(2):285–290, 1994.

[258] M. M. Tirion. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Physical Review Letters*, 77(9):1905–1908, 1996.

[259] K. Hinsen. Analysis of domain motions by approximate normal mode calculations. *Proteins*, 33(3):417–429, 1998.

[260] I. Bahar, A. R. Atilgan, and B. Erman. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design*, 2(3):173–181, 1997.

[261] A. R. Atilgan, S. R. Durell, R. L. Jernigan, M. C. Demirel, O. Keskin, and I. Bahar. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophysical Journal*, 80(1):505–515, 2001.

[262] F. Tama and Y. H. Sanejouand. Conformational change of proteins arising from normal mode calculations. *Protein Engineering*, 14(1):1–6, 2001.

[263] C. Chennubhotla, A. J. Rader, L. W. Yang, and I. Bahar. Elastic network models for understanding biomolecular machinery: from enzymes to supramolecular assemblies. *Physical Biology*, 2(4):S173, 2005.

[264] A. Ahmed, S. Villinger, and H. Gohlke. Large-scale comparison of protein essential dynamics from molecular dynamics simulations and coarse-grained normal mode analyses. *Proteins*, 78(16):3341–3352, 2010.

[265] L. Yang, G. Song, and R. L. Jernigan. How well can we understand large-scale protein motions using normal modes of elastic network models? *Biophysical Journal*, 93(3):920–929, 2007.

[266] A. Ahmed and H. Gohlke. Multiscale modeling of macromolecular conformational changes combining concepts from rigidity and elastic network theory. *Proteins: Structure, Function, and Bioinformatics*, 63(4):1038–1051, 2006.

[267] S. Kundu and R. L. Jernigan. Molecular mechanism of domain swapping in proteins: an analysis of slower motions. *Biophysical Journal*, 86(6):3846–3854, 2004.

[268] S. Kundu, D. C. Sorensen, and G. N. Phillips. Automatic domain decomposition of proteins by a Gaussian Network Model. *Proteins: Structure, Function, and Bioinformatics*, 57(4):725–733, 2004.

[269] R. Tatsumi, Y. Fukunishi, and H. Nakamura. A hybrid method of molecular dynamics and harmonic dynamics for docking of flexible ligand to flexible receptor. *Journal of Computational Chemistry*, 25(16):1995–2005, 2004.

[270] Z. Zhang, Y. Shi, and H. Liu. Molecular dynamics simulations of peptides and proteins with amplified collective motions. *Biophysical Journal*, 84(6):3583–3593, 2003.

[271] F. Tama, O. Miyashita, and C. L. Brooks. Flexible multi-scale fitting of atomic structures into low-resolution electron density maps with elastic network normal mode analysis. *Journal of Molecular Biology*, 337(4):985–999, 2004.

[272] F. Tama, O. Miyashita, and C. L. Brooks. Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-EM. *Journal of Structural Biology*, 147(3):315–326, 2004.

[273] M. Delarue. Dealing with structural variability in molecular replacement and crystallographic refinement through normal-mode analysis. *Acta Crystallographica Section D: Biological Crystallography*, 64(Pt 1):40–48, 2008.

[274] K. Hinsen, N. Reuter, J. Navaza, D. L. Stokes, and J. J. Lacapère. Normal mode-based fitting of atomic structure into electron density maps: application to sarcoplasmic reticulum Ca-ATPase. *Biophysical Journal*, 88(2):818–27, 2005.

[275] P. Durand, G. Trinquier, and Y.-H. Sanejouand. A new approach for determining low-frequency normal modes in macromolecules. *Biopolymers*, 34(6):759–771, 1994.

[276] F. Tama, F. X. Gadea, O. Marques, and Y. H. Sanejouand. Building-block approach for determining low-frequency normal modes of macromolecules. *Proteins*, 41(1):1–7, 2000.

[277] G. Li and Q. Cui. A coarse-grained normal mode approach for macromolecules: an efficient implementation and application to Ca^{2+}-ATPase. *Biophysical Journal*, 83(5):2457–2474, 2002.

[278] A. D. Schuyler and G. S. Chirikjian. Normal mode analysis of proteins: a comparison of rigid cluster modes with C_α coarse graining. *Journal of Molecular Graphics and Modeling*, 22(3):183–193, 2004.

[279] P. Doruker, R. L. Jernigan, and I. Bahar. Dynamics of large proteins through hierarchical levels of coarse-grained structures. *Journal of Computational Chemistry*, 23(1):119–127, 2002.

[280] W. Zheng and B. R. Brooks. Probing the local dynamics of nucleotide-binding pocket coupled to the global dynamics: myosin versus kinesin. *Biophysical Journal*, 89(1):167–178, 2005.

[281] D. Ming and M. E. Wall. Allostery in a coarse-grained model of protein dynamics. *Physical Review Letters*, 95(19):198103, 2005.

[282] K. Eom, S. C. Baek, J. H. Ahn, and S. Na. Coarse-graining of protein structures for the normal mode studies. *Journal of Computational Chemistry*, 28(8):1400–1410, 2007.

[283] C. Micheletti, P. Carloni, and A. Maritan. Accurate and efficient description of protein vibrational dynamics: comparing molecular dynamics and gaussian models. *Proteins: Structure, Function, and Bioinformatics*, 55(3):635–645, 2004.

[284] O. Kurkcuoglu, R. L. Jernigan, and P. Doruker. Mixed levels of coarse-graining of large proteins using elastic network model succeeds in extracting the slowest motions. *Polymer*, 45(2):649–657, 2004.

[285] S. E. Dobbins, V. I. Lesk, and M. J. E. Sternberg. Insights into protein flexibility: the relationship between normal modes and conformational change upon protein-protein docking. *Proceedings of the National Academy of Sciences*, 105(30):10390–10395, 2008.

[286] A. May and M. Zacharias. Accounting for global protein deformability during protein-protein and protein-ligand docking. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1754(1-2):225–231, 2005.

[287] K. Bastard, C. Prévost, and M. Zacharias. Accounting for loop flexibility during protein-protein docking. *Proteins*, 62(4):956–69, 2006.

[288] M. Zacharias. Accounting for conformational changes during protein-protein docking. *Current Opinion in Structural Biology*, 20(2):180–6, 2010.

[289] A. Ahmed, F. Rippmann, G. Barnickel, and H. Gohlke. A normal mode-based geometric simulation approach for exploring biologically relevant conformational transitions in proteins. *Journal of Chemical Information and Modeling*, 51(7):1604–1622, 2011.

[290] M. Rueda, G. Bottegoni, and R. Abagyan. Consistent improvement of cross-docking results using binding site ensembles generated with elastic network normal modes. *Journal of Chemical Information and Modeling*, 49(3):716–725, 2009.

[291] P. Petrone and V. S. Pande. Can conformational change be described by only a few normal modes? *Biophysical Journal*, 90(5):1583–1593, 2006.

[292] A. May and M. Zacharias. Protein-ligand docking accounting for receptor side chain and global flexibility in normal modes: evaluation on kinase inhibitor cross docking. *Journal of Medicinal Chemistry*, 51(12):3499–3506, 2008.

[293] C. N. Cavasotto, J. A. Kovacs, and R. A. Abagyan. Representing receptor flexibility in ligand docking through relevant normal modes. *Journal of the American Chemical Society*, 127(26):9632–9640, 2005.

[294] M. J. Hartshorn, M. L. Verdonk, G. Chessari, S. C. Brewerton, W. T. M. Mooij, P. N. Mortenson, and C. W. Murray. Diverse, high-quality test set for the validation of protein-ligand docking performance. *Journal of Medicinal Chemistry*, 50(4):726–741, 2007.

[295] M. L. Verdonk, P. N. Mortenson, R. J. Hall, M. J. Hartshorn, and C. W. Murray. Protein-ligand docking against non-native protein conformers. *Journal of Chemical Information and Modeling*, 48(11):2214–2225, 2008.

[296] M. Levitt, C. Sander, and P. S. Stern. The normal modes of a protein: native bovine pancreatic trypsin inhibitor. *International Journal of Quantum Chemistry*, 24(S10):181–199, 1983.

[297] B. Brooks and M. Karplus. Harmonic dynamics of proteins: normal modes and fluctuations in bovine pancreatic trypsin inhibitor. *Proceedings of the National Academy of Sciences*, 80(21):6571–6575, 1983.

[298] A. Hildebrandt, A. Dehof, A. Rurainski, A. Bertsch, M. Schumann, N. Toussaint, A. Moll, D. Stöckel, S. Nickels, S. C. Mueller, H.-P. Lenhof, and O. Kohlbacher. BALL - biochemical algorithms library 1.3. *BMC Bioinformatics*, 11(1):531, 2010.

[299] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison. Open babel: an open chemical toolbox. *Journal of Cheminformatics*, 3(1):1–14, 2011.

[300] P. A. Kollman, R. Dixon, W. Cornell, T. Fox, C. Chipot, and A. Pohorille. The development/application of a "minimalist" organic/biochemical molecular mechanic force field using a combination of ab initio calculations and experimental data. *Computer Simulation of Biomolecular Systems*, 3:83–96, 1997.

[301] J. Nocedal. Updating Quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[302] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[303] I. K. McDonald and J. M. Thornton. Satisfying hydrogen bonding potential in proteins. *Journal of Molecular Biology*, 238:777–793, 1994.

[304] A. C. Wallace, R. A. Laskowski, and J. M. Thornton. LIGPLOT: a program to generate schematic diagrams of protein-ligand interactions. *Protein Engineering*, 8(2):127–134, 1995.

[305] J. A. Kovacs, P. Chacón, and R. Abagyan. Predictions of protein flexibility: first-order measures. *Proteins*, 56(4):661–668, 2004.

[306] A. Shrake and J. A. Rupley. Environment and exposure to solvent of protein atoms. lysozyme and insulin. *Journal of molecular biology*, 79(2):351–371, 1973.

[307] K. Hinsen, A. Thomas, and M. J. Field. Analysis of domain motions in large proteins. *Proteins: Structure, Function, and Bioinformatics*, 34(3):369–382, 1999.

[308] L. Yang, G. Song, and R. L. Jernigan. Protein elastic network models and the ranges of cooperativity. *Proceedings of the National Academy of Sciences*, 106(30):12347–12352, 2009.

[309] C. Oostenbrink, A. Villa, A. E. Mark, and W. F. Van Gunsteren. A biomolecular force field based on the free enthalpy of hydration and solvation: The GROMOS force-field parameter sets 53A5 and 53A6. *Journal of Computational Chemistry*, 25(13):1656–1676, 2004.

[310] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl. GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. *Journal of Chemical Theory and Computation*, 4(3):435–447, 2008.

[311] A. W. Schuttelkopf and D. M. F. van Aalten. PRODRG: a tool for high-throughput crystallography of protein-ligand complexes. *Acta Crystallographica Section D: Biological Crystallography*, 60(8):1355–1363, 2004.

[312] M. Vandonselaar, R. A. Hickie, W. Quail, and L. T. J. Delbaere. Trifluoperazine-induced conformational change in Ca^{2+}-calmodulin. *Nature Structural & Molecular Biology*, 1(11):795–801, 1994.

[313] C. Chothia. Hydrophobic bonding and accessible surface area in proteins. *Nature*, 248(5446):338–339, 1974.

[314] C. Chothia. The nature of the accessible and buried surfaces in proteins. *Journal of molecular biology*, 105(1):1–12, 1976.

[315] C. Chothia. Principles that determine the structure of proteins. *Annual Review of Biochemistry*, 53(1):537–572, 1984.

[316] C. Chothia and J. Janin. Principles of protein-protein recognition. *Nature*, 256(5520):705–708, 1975.

[317] C. Chothia, S. Wodak, and J. Janin. Role of subunit interfaces in the allosteric mechanism of hemoglobin. *Proceedings of the National Academy of Sciences*, 73(11):3793–3797, 1976.

[318] A. Ben-Shem, N. G. de Loubresse, S. Melnikov, L. Jenner, G. Yusupova, and M. Yusupov. The structure of the eukaryotic ribosome at 3.0Å resolution. *Science*, 334(6062):1524–1529, 2011.

[319] A. N. Cronshaw, J. M .and Krutchinsky, W. Zhang, B. T. Chait, and M. J. Matunis. Proteomic analysis of the mammalian nuclear pore complex. *The Journal of cell biology*, 158(5):915–927, 2002.

[320] F. Alber, S. Dokudovskaya, L. M. Veenhoff, W. Zhang, J. Kipper, D. Devos, A. Suprapto, O. Karni-Schmidt, R. Williams, B. T. Chait, M. P. Rout, and A. Sali. Determining the architectures of macromolecular assemblies. *Nature*, 450(7170):683–694, 2007.

[321] D. Rajamani, S. Thiel, S. Vajda, and C. J. Camacho. Anchor residues in protein–protein interactions. *Proceedings of the National Academy of Sciences of the United States of America*, 101(31):11287–11292, 2004.

[322] F. Pazos, M. Helmer-Citterich, G. Ausiello, and A. Valencia. Correlated mutations contain information about protein-protein interaction. *Journal of Molecular Biology*, 271(4):511–523, 1997.

[323] E. Sprinzak and H. Margalit. Correlated sequence-signatures as markers of protein-protein interaction. *Journal of Molecular Biology*, 311(4):681–692, 2001.

[324] D. T. Jones, D. W. A. Buchan, D. Cozzetto, and M. Pontil. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

[325] J. W. Back, L. de Jong, A. O. Muijsers, and C. G. de Koster. Chemical cross-linking and mass spectrometry for protein structural modeling. *Journal of molecular biology*, 331(2):303–313, 2003.

[326] A. Sinz. Chemical cross-linking and mass spectrometry to map three-dimensional protein structures and protein-protein interactions. *Mass Spectrometry Reviews*, 25(4):663–682, 2006.

[327] A. Leitner, T. Walzthoeni, A. Kahraman, F. Herzog, O. Rinner, M. Beck, and R. Aebersold. Probing native protein structures by chemical cross-linking, mass spectrometry, and bioinformatics. *Molecular & Cellular Proteomics*, 9(8):1634–1649, 2010.

[328] M. T. Degiacomi and M. Dal Peraro. Macromolecular symmetric assembly prediction using swarm intelligence dynamic modeling. *Structure*, 21(7):1097–1106, 2013.

[329] I. André, P. Bradley, C. Wang, and D. Baker. Prediction of the structure of symmetrical protein assemblies. *Proceedings of the National Academy of Sciences*, 104(45):17656–17661, 2007.

[330] P. Popov, D. W. Ritchie, and S. Grudinin. DockTrina: Docking triangular protein trimers. *Proteins: Structure, Function, and Bioinformatics*, 2013.

[331] V. Venkatraman and D. W. Ritchie. Predicting multi-component protein assemblies using an ant colony approach. *International Journal of Swarm Intelligence Research*, 3(3):19–31, 2012.

[332] S.-Y. Huang. Search strategies and evaluation in protein–protein docking: principles, advances and challenges. *Drug discovery today*, 2014.

[333] I. André, C. E. M. Strauss, D. B. Kaplan, P. Bradley, and D. Baker. Emergence of symmetry in homooligomeric biological assemblies. *Proceedings of the National Academy of Sciences*, 105(42):16148–16152, 2008.

[334] D. S. Goodsell and A. J. Olson. Structural symmetry and protein function. *Annual Review of Biophysics and Biomolecular Structure*, 29(1):105–153, 2000.

[335] F. Alber, F. Förster, D. Korkin, M. Topf, and A. Sali. Integrating diverse data for structure determination of macromolecular assemblies. *Annu. Rev. Biochem.*, 77:443–477, 2008.

[336] J. H. Challis. A procedure for determining rigid body transformation parameters. *Journal of Biomechanics*, 28(6):733–737, 1995.

[337] J. Vince. Mathematics. *Mathematics for Computer Graphics*, 2010.

[338] R. Hammack, W. Imrich, and S. Klavžar. *Handbook of product graphs*. Taylor & Francis US, 2011.

[339] H. C. H. Weyl. *Symmetry*. Princeton University Press, 1952.

[340] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*, 27(3):Article 43, 1–11, 2008.

[341] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan. Symmetry in 3D geometry: Extraction and applications. In *Computer Graphics Forum*. Wiley Online Library, 2013.

[342] B. Kojić-Prodić and Z. Štefanić. Symmetry versus asymmetry in the molecules of life: Homomeric protein assemblies. *Symmetry*, 2(2):884–906, 2010.

[343] D. F. Holt, B. Eick, and E. A. O'Brien. *Handbook of computational group theory*. CRC Press, 2005.

[344] D. Kozakov, K. H. Clodfelter, S. Vajda, and C. J. Camacho. Optimal clustering for detecting near-native conformations in protein docking. *Biophysical Journal*, 89(2):867–875, 2005.

[345] S. Lorenzen and Y. Zhang. Identification of near-native structures by clustering protein docking conformations. *Proteins: Structure, Function, and Bioinformatics*, 68(1):187–194, 2007.

[346] B. Pierce and Z. Weng. A combination of rescoring and refinement significantly improves protein docking performance. *Proteins: Structure, Function, and Bioinformatics*, 72(1):270–279, 2008.

[347] S. Vajda and D. Kozakov. Convergence and combination of methods in protein-protein docking. *Current Opinion in Structural Biology*, 19(2):164–170, 2009.

[348] Peter G Wolynes. Symmetry and the energy landscapes of biomolecules. *Proceedings of the National Academy of Sciences of the United States of America*, 93(25):14249, 1996.

[349] D. Q. Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[350] T. Bajd, M. Mihelj, and M. Munih. *Introduction to Robotics*. Springer, 2013.

[351] A. K. Hildebrandt, M. Dietzen, T. Lengauer, H.-P. Lenhof, E. Althaus, and A. Hildebrandt. Efficient computation of root mean square deviations under rigid transformations. *Journal of Computational Chemistry*, 35(10):765–771, 2014.

[352] C. J. Albers, F. Critchley, and J. C. Gower. Quadratic minimisation problems in statistics. *Journal of Multivariate Analysis*, 102(3):698–713, 2011.

[353] E. Krissinel and K. Henrick. Inference of macromolecular assemblies from crystalline state. *Journal of Molecular Biology*, 372(3):774–797, 2007.

[354] K. Henrick and J. M. Thornton. PQS: a protein quaternary structure file server. *Trends in Biochemical Sciences*, 23(9):358–361, 1998.

[355] A. Andreeva, D. Howorth, J.-M. Chandonia, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin. Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Research*, 36(Database issue):D419–D425, 2008.

[356] http://www.wwpdb.org/documentation/format32/v3.2.html, 2012.

[357] J. D. Fischer, G. L. Holliday, and J. M. Thornton. The CoFactor database: organic cofactors in enzyme catalysis. *Bioinformatics*, 26(19):2496–2497, 2010.

[358] http://www.rcsb.org/pdb/statistics/clusterStatistics.do, 2012.

[359] P. D. Gabanyi, M. J.and Adams, K. Arnold, L. Bordoli, L. G. Carter, J. Flippen-Andersen, L. Gifford, J. Haas, A. Kouranov, W. A. McLaughlin, D. I. Micallef, W. Minor, R. Shah, T. Schwede, Y. P. Tao, J. D. Westbrook, M. Zimmerman, and H. M. Berman. The Structural Biology Knowledgebase: a portal to protein structures, sequences, functions, and methods. *Journal of Structural and Functional Genomics*, 12(2):45–54, 2011.

[360] Schrödinger, LLC. Pymol. The PyMOL Molecular Graphics System, Version 1.5.0.1, Schrödinger, LLC., 2012.

[361] R. B. Russell, F. Alber, P. Aloy, F. P. Davis, D. Korkin, M. Pichaud, M. Topf, and A. Šali. A structural perspective on protein-protein interactions. *Current Opinion in Structural Biology*, 14(3):313–324, 2004.

[362] N. Zaki. Protein-protein interaction prediction using homology and inter-domain linker region information. In *Advances in Electrical Engineering and Computational Science*, pages 635–645. Springer, 2009.

[363] M. M. F. Bugalho and A. L. Oliveira. Constant time clash detection in protein folding. *Journal of Bioinformatics and Computational Biology*, 7(1):55–74, 2009.

[364] K. Yamada, T. Miyata, D. Tsuchiya, T. Oyama, Y. Fujiwara, T. Ohnishi, H. Iwasaki, H. Shinagawa, M. Ariyoshi, K. Mayanagi, and K. Morikawa. Crystal structure of the RuvA-RuvB complex: a structural basis for the Holliday junction migrating motor machinery. *Molecular Cell*, 10(3):671–681, 2002.

[365] J. G. S. Ho, P. I. Kitov, E. Paszkiewicz, J. Sadowska, D. R. Bundle, and K. K.-S. Ng. Ligand-assisted aggregation of proteins. Dimerization of serum amyloid P component by bivalent ligands. *Journal of Biological Chemistry*, 280(36):31999–32008, 2005.

[366] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1(6):80–83, 1945.

[367] A. R. Crofts. The cytochrome *bc1* complex: function in the context of structure. *Annual Review of Physiology*, 66:689–733, 2004.

[368] P. D. Boyer. The ATP synthase – a splendid molecular machine. *Annual Review of Biochemistry*, 66(1):717–749, 1997.

[369] G. Oster and H. Wang. ATP synthase: two motors, two fuels. *Structure*, 7(4):R67–R72, 1999.

[370] Y. Liu, L. Xu, N. Opalka, J. Kappler, H.-B. Shu, and G. Zhang. Crystal structure of sTALL-1 reveals a virus-like assembly of TNF family ligands. *Cell*, 108(3):383–394, 2002.

[371] U. Ogmen, O. Keskin, A. S. Aytuna, R. Nussinov, and A. Gursoy. PRISM: protein interactions by structural matching. *Nucleic acids research*, 33(suppl 2):W331–W336, 2005.

[372] E. D. Lowe, N. Hasan, J.-F. Trempe, L. Fonso, M. E. M. Noble, J. A. Endicott, L. N. Johnson, and N. R. Brown. Structures of the Dsk2 UBL and UBA domains and their complex. *Acta Crystallographica Section D: Biological Crystallography*, 62(2):177–188, 2006.

[373] N. T. Doncheva, K. Klein, F. S. Domingues, and M. Albrecht. Analyzing and visualizing residue networks of protein structures. *Trends in Biochemical Sciences*, 36(4):179–182, 2011.

[374] C. Ericson. *Real-Time Collision Detection*. Taylor & Francis US, 2005.

[375] B. Gärtner. Fast and robust smallest enclosing balls. In *Algorithms-ESA'99*, pages 325–338. Springer, 1999.

[376] K. Fischer, B. Gärtner, and M. Kutz. Fast smallest-enclosing-ball computation in high dimensions. In *Algorithms-ESA 2003*, pages 630–641. Springer, 2003.

[377] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[378] B. Gärtner. A subexponential algorithm for abstract optimization problems. *SIAM Journal on Computing*, 24(5):1018–1035, 1995.

[379] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[380] I. Schur. Ein Satz über quadratische Formen mit komplexen Koeffizienten. *American Journal of Mathematics*, pages 472–480, 1945.

[381] J. Arvo. A simple method for box-sphere intersection testing. In *Graphics Gems*, pages 335–339. Academic Press Professional, Inc., 1990.

[382] I. Sharf, A. Wolf, and M. B. Rubin. Arithmetic and geometric solutions for average rigid-body rotation. *Mechanism and Machine Theory*, 45(9):1239–1251, 2010.

[383] W. D. Curtis, A. L. Janin, and K. Zikan. A note on averaging rotations. In *IEEE Virtual Reality Annual International Symposium*, pages 377–385. IEEE, 1993.

[384] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003.

[385] M. F. Lensink, R. Méndez, and S. J. Wodak. Docking and scoring protein complexes: CAPRI 3rd edition. *Proteins: Structure, Function, and Bioinformatics*, 69(4):704–718, 2007.

[386] J. Janin. Protein-protein docking tested in blind predictions: the CAPRI experiment. *Molecular BioSystems*, 6(12):2351–2362, 2010.