Martin-Luther-Universität Halle-Wittenberg
Institute for Biochemistry and Biotechnology
Faculty I of Natural Science – Biological Science

Diploma Thesis

Modeling the Phytochelatinsynthase of *Chlamydomonas reinhardtii*

Patrice Peterson

October 27, 2013

Advisors: PD Dr. Iris Thondorf and Dr. Dirk Dobritzsch

Abstract

Especially since the advent of human civilization, heavy metal detoxification has become a vital and necessary stress response mechanism in plants. The enzyme phytochelatin synthase plays a key role in heavy metal detoxification and related metabolic processes. In this diploma thesis, N- and C-terminally truncated monomeric and dimeric models of phytochelatin synthase of *Chlamydomonas reinhardtii* (CrPCS) have been created using Modeller and a custom modeling procedure—the performance of which compares favorably to several high-ranking automatic modeling servers—and submitted to molecular dynamics simulations using the Amber and Gromacs packages. Explicit solvent simulations can be considered stable, implicit solvent simulations did not reach stable conformations in the given timeframe. Additionally, the possibility of crosslinking surface-facing lysine residues was explored in order to verify the obtained models experimentally in the future. Four suitable lysine residues were identified around the active site, two more were in the unmodeled N-terminus of the protein. As no lysines exist in the unmodeled, 40-residue long C-terminus of CrPCS, crosslinking would not contribute to determining the structure of the protein's C-terminus. The code implementing the modeling procedure developed for this project can be found at `https://bitbucket.org/runiq/modeling-clustering`.

> “Hofstadter’s Law: It always takes longer than you expect, even when you take into account Hofstadter’s Law.”
>
> *—Douglas R. Hofstadter*

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

Phytochelatin Synthases

AtPCS	Phytochelatin synthase of *Arabidopsis thaliana*
CePCS	Phytochelatin synthase of *Cænorhabditis elegans*
CrPCS	Phytochelatin synthase of *Chlamydomonas reinhardtii*
NsPCS	Phytochelatin synthase of *Nostoc spec.*
PCS	Phytochelatin synthase
SpPCS	Phytochelatin synthase of *Saccharomyces pombe*
TaPCS	Phytochelatin synthase of *Tuber æstivum*

Chemicals

γ-EC	γ-glutamylcysteine
GSH	Glutathione
MT	Metallothioneïn
NA	Nicotianamine
PC	Phytochelatin

Other

CASP9	Critical Assessment of Methods of Protein Structure Prediction 9
DBI	Davies-Bouldin index
EM	Energy minimization
GB	Generalized Born
HMM	Hidden Markov model

HMW	High molecular weight
LMW	Low molecular weight
MD	Molecular dynamics
MSA	Multiple-sequence alignment
PAM	Probabilistic alignment matrix
PDF	Probability density function
PME	Particle-mesh Ewald
pSF	Pseudo F-statistic
PSSM	Position-specific scoring matrix
QTA	Query–template alignment
$R_{\mathbf{gyr}}$	Radius of gyration
RMSD	Root mean square deviation
RMSF	Root mean square fluctuation
ROS	Reactive oxygen species
SCR	Structurally conserved region
SSR/SST	Ratio of the sum of squares regression to the total sum of squares
SVR	Structurally variable region
TTA	Target–template alignment

1. Introduction

The ability to handle stress is a necessity in any organism's life. It can come in various forms, and any and all constituents of an organism's environment can enact stressful influences on it. Consequently, even a normally benign environmental factor can have a toxic effect (Figure 1.1): Despite Zn^{2+}'s function as a cofactor for various enzymes, such as alcohol dehydrogenase, carboxy- and aminopeptidases, and transcription factors with the zinc finger motif, ingestion of very high concentrations can lead to nausea, vomiting, epigastric pain, lethargy, and fatigue in humans [46]. Similarly, non-essential elements can only be tolerated up to a certain threshold. The toxic effects of a non-essential element such as Cd^{2+} on humans include emphysema, osteoporosis, and irreversible damage to lungs, kidneys, and bones in humans [152]. When left untreated, Cd^{2+}-polluted water can accumulate in the food chain and potentially affect a large number of people [86].

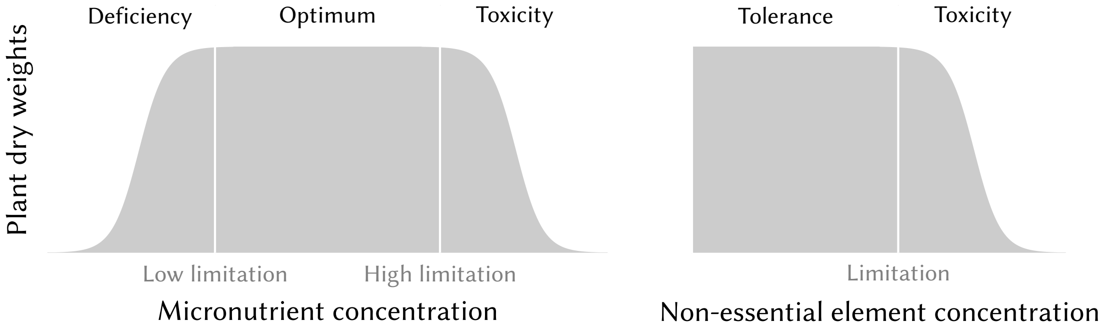


Figure 1.1. Dose-response curves of plants to micronutrients (Zn^{2+}) and non-essential elements (Cd^{2+}), adapted from Lin and Aarts [79]. The Zn^{2+} ion Zn^{2+} is a micronutrient, so Zn^{2+}-deficient plants do not grow as well as those supplied with sufficient amounts of the ion. On the contrary, the Cd^{2+} ion is detrimental to plant growth at high enough concentrations and is not known to be used as a cofactor, resulting in a dose-response curve without a low limitation.

Among the cell biological consequences of prolonged exposure to toxic concentrations of heavy metals are membrane disintegration, ion leakage, lipid peroxidation, DNA/RNA degradation, and eventually cell death [79]. Heavy metals also cause the production of hydroxyl radicals, which in turn disturb the electron transport chain and have a detrimental effect on the antioxidant defence system [6].

Most of the effects of heavy metals on plants are assumed to be general stress responses as exposure to different heavy metals often results in similar symptoms. Those symptoms manifest in the *heavy metal toxicity syndrome*, which includes leaf chlorosis, disturbed water balance, and reduced stomatal opening [6].

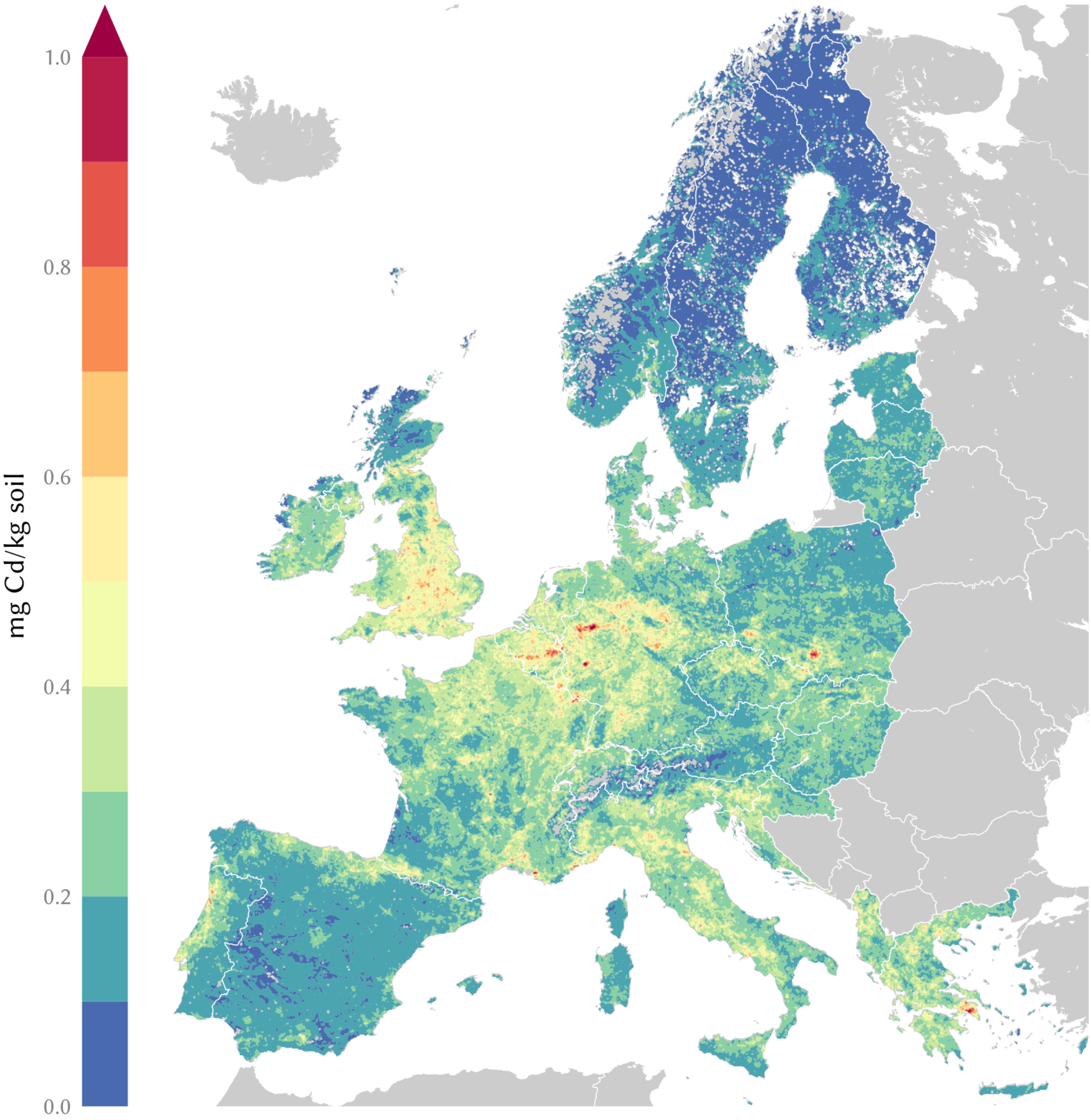


Figure 1.2. Soil contamination with cadmium in central and west Europe in 2007, taken from 1588 samples from the Forum of European Geological Surveys Geochemical database. Adapted from Lado, Hengl, and Reuter [75].

Toxic soil concentrations of heavy metals pose a threat that previously was present only in rare and small areas, such as calamine outcrops where heavy metal-rich minerals reach the surface, or nonmetalliferous soils followed by later contamination by metalliferous soils [74]. However, due to the ascent of human civilization and especially the advent of mining, heavy metal concentrations were rising especially in industrial centers (Figure 1.2). It is therefore not entirely clear how tolerance to high zinc and cadmium concentrations—and, by extension, the existence of so-called hyperaccumulators of these metals—has evolved. In particular, high Cd^{2+} concentrations in soil and water are a relatively recent phenomenon, highly correlated with the expansion of human habitats and the advent of mining.

Due to the inability to escape their heavy metal-plagued habitat, plants were pressed to develop different strategies in order to deal with both nutrient deficiency and toxic substances. This eventually lead to the evolution of so-called *hyperaccumulators*, plants which can grow and thrive in heavy metal-contaminated soils. These plants are able to absorb metal at doses 50 to 500 times greater than average plants without exhibiting toxicity symptoms [77]. Table 1.1 provides an overview over the tolerance levels of land plants against several trace metals; the last column shows the concentration level needed for a species to be considered a hyperaccumulator. The ability for hyperaccumulation was first described in the family *Brassicaceæ* in 1865 [134]; since then, hyperaccumulators from different families have been identified, though members of *Brassicaceæ* are still the most numerous.

Table 1.1. Tolerance levels for trace elements in land plants [74]

Element	Critical deficiency level ($\mu g\,g^{-1}$)	Critical toxicity level ($\mu g\,g^{-1}$)	Hyperaccumulation concentration criterion ($\mu g\,g^{-1}$)
Antimony	n. r.	< 2	> 1000
Arsenic	n. r.	$< 2 - 80$	> 1000
Cadmium	n. r.	$6 - 10$	> 100
Cobalt	n. r.	0.4—several	> 1000
Copper	1 to 5	20 to 30	> 1000
Lead	n. r.	0.6 to 28	> 1000
Manganese	10 to 20	200 to 3500	$>10\,000$
Nickel	0.002 to 0.004	10 to 50	> 1000
Selenium	n. r.	3 to 100	> 1000
Thallium	n. r.	20	> 1000
Zinc	15 to 20	100 to 300	$>10\,000$

Hyperaccumulators can be used for *phytoremediation*, such as lowering concentrations of heavy metals in water [171] or soil [136]. The process of wastewater detoxification by micro- and macroalgæ is called *phycoremediation*. In contrast to more invasive methods like soil excavation, which disrupts the soil structure, the local ecosystem, and reduces soil productivity, phytoremediation is an environmentally friendly process. Its success depends on a number of factors, among them the biomass of the employed hyperaccumulator, its bioconcentration factor, and its growth rate. For example, Willow (*Salix viminalis*) is used for cadmium phytoremediation over aquiferous soil since it can accumulate a significant amount of cadmium in its shoots and leaves ($27\,\mathrm{mg\,kg^{-1}_{dry\ weight}}$) [52].

Significant efforts have also been made to better understand the mechanism of heavy metal detoxification. Despite earlier research stating the contrary [7], a study by Chaney et al. [25] concludes that heavy metal detoxification and hyperaccumulation are correlated. Elucidating the mechanism of heavy metal detoxification and solving the structure of involved enzymes could therefore provide insight into hyperaccumulation as well.

To that end, the green alga *Chlamydomonas reinhardtii* is often employed as a model organism for research on heavy metal tolerance in plants. *C. reinhardtii* is a cheap, viable recombinant protein expression system [128] and retains more than half of its accumulated Cd^{2+} content in chloroplasts, which makes it easily extractable [100]. Therefore, providing insight into the heavy metal detoxification process in that organism can prove a viable next step on the way to understanding hyperaccumulation and improve its applications.

1.1. Heavy Metal Detoxification

There is some controversy as to what constitutes a “heavy metal” in the biological sense as its effects are largely independent of its density and instead rely on its chemical properties. A feasible classification of metal ions based on equilibrium constants of metal ion–ligand complex formation was proposed by Nieboer and Richardson [102]. While the classification of metals into classes A (nitrogen-seeking) and B (oxygen-/sulfur-seeking) generally correlates with the metal ion species’ toxicities, the situation is less clear-cut for the “borderline” class of metal ions: Zn^{2+} and Cd^{2+} are both borderline elements; yet, the former is an essential micronutrient while the latter is inherently toxic.

The manifold defense mechanisms employed by plants must therefore be both varied and specific: They have to keep the micronutrient concentration in a narrow optimal range while at the same time keeping toxic heavy metals out of enzymes’ reaction centers. Based on how they achieve this, plants can be divided into three categories: Metal-resistant excluders, metal-tolerant non-hyperaccumulators, and hyperaccumulators [79]. Excluders aim to deter heavy metals from entering their roots, either by reducing their bioavailability or by reducing the expression of metal uptake transporters. Accumulators rely on confinement and detoxification of metals in a controlled way, either by sequestration into vacuoles, confinement to the apoplast, or chelation in the cytosol. These methods are not exclusive; for instance, the chelation of a heavy metal ion is often also a signal for sequestration of the chelate into the vacuole, or for inhibiting its uptake into the symplast by transport proteins.

In general, the defense mechanism of land plants consists of several parts [79]:

Reducing Bioavailability Bioavailability determines whether the metal is in a form that can be taken up by a plant or not. By secreting metal chelators and altering the pH of the local environment, plant shoots can alter the effective concentration of bioavailable metal ions [47]. Such exudates can be organic and amino acids, precipitants, sugars, polysaccharides, and proteins. Additionally, mycorrhizal symbionts, which are present in most land plants, and bacterial microbes can further reduce bioavailability by taking up metal ions from the environment and secreting chelating agents themselves. By altering the composition and thickness of their cell wall, plant cells can reduce the amount of metal ions which are able to get into the cell.

Controlling Metal Flux Metal influx and efflux are regulated by specific transporter proteins in the cell membrane. These have to be both specific and regulable in order to carry out their intended function. In the case of toxic metals such as Cd^{2+}, influx is not known to happen due to dedicated Cd^{2+} transporters; instead, Cd^{2+} is assumed to have low affinity for Zn^{2+} transporters [73, 115] and potentially iron transporters as well [83]. In *C. reinhardtii*, for example, Cd^{2+} is known to enter through a metal transporter of the Nramp family [127]. Interestingly, no dedicated Cd^{2+} *efflux* transporters have been identified either, which means cadmium efflux is conferred by nonspecific transporters as well.

Chelation The process of chelation aims to incapacitate metal ions which have overcome the plasma membrane. Hyperaccumulators often show higher concentrations of chelating agents than non-hyperaccumulators [79]. Common chelators include nicotianamine (NA), metallothioneïns (MTs), glutathione (GSH), and phytochelatins (PCs). NA is an ubiquitous, non-proteinogenic amino acid which plays an important role in cellular and systemic iron acquisition as well as intracellular metal transfer [155]. It is produced in cases of iron and zinc deficiency stress. MTs are genetically encoded polypeptides with a high number of Cys residues; these residues are placed so as to maximize the molecule's chelate effect. They are vital in tolerance and accumulation of Cu and tolerance of Zn/Cd and can scavenge reactive oxygen species (ROS) as well [33, 57]. GSH is a γ–Glu−Cys−Gly tripeptide which chelates metal ions by the thiol group of its Cys residue. Its roles in the cell are diverse, but mostly oxidation-stress related, ranging from keeping the cellular oxidation status, over acting as a signal for the occurrence of ROS, to being a precursor for PCs by means of phytochelatin synthase (PCS). PCs have the general formula $[\gamma{-}\mathrm{Glu}{-}\mathrm{Cys}]_n{-}\mathrm{Gly}$ and are usually between $n = 5 \cdots 11$ in length. While PCs contribute to heavy metal tolerance, they do not play a substantial role in hyperaccumulators [72, 137].

Sequestration The next stage is to move toxic ions (or chelates) from places where they can do damage to safer ones, including vacuole(s), the cell wall, or different tissues which are better equipped to deal with them. How sequestration is carried out depends on the plant's capability to tolerate high heavy metal concentrations: Non-tolerent species sequester primarily into root vacuoles since high concentrations in the photosynthesis apparatus are potentially more dangerous; this sequestration is mediated primarily by ATP binding cassette-type transporters [111]. Tolerant species and (hyper-)accumulators favor sequestration into shoot tissues by means of the xylem network, using the leaves as storage [79]. *Chlamydomonas reinhardtii* and *Euglena gracilis* are special in that they sequester $\mathrm{PC} \cdot \mathrm{Cd}^{2+}$ complexes into plastids instead [93, 94, 100]. Often, metal ions chelated by MTs or PCs can initiate the sequestration process.

ROS Detoxification Introducing toxic heavy metals into cells leads to oxidative stress in the form of superoxide $\mathrm{O_2}^{-\bullet}$, hydroxyl radicals $\mathrm{OH}^{\bullet}$, hydrogen peroxide $\mathrm{H_2O_2}$, or singlet oxygen $^1\mathrm{O_2}$. These so-called reactive oxygen species (ROS) are highly reactive and can damage a broad range of molecules and cellular structures. However, since oxidative stress can have various sources, the cellular response is fairly general in nature, mainly consisting of an increase in GSH production and antioxidant enzymes like superoxide dismutase, catalase, and GSH reductase. Metal tolerant and (hyper-)accumulator plants generally have a higher level of antioxidant enzymes than metal sensitive ones. As metal-induced stress leads to oxidative stress, keeping the right ratio of GSH to PCs is a balancing act: Both have to be kept at a sufficient level to perform their duties. Of special note is the Zn^{2+} ion and its dual role in ROS detoxification: While an overabundance of the ion leads to induction of oxidative stress, it is also a cofactor in superoxide dismutase [22].

1.2. Phytochelatins

Historically, it was assumed that chelation of heavy metal ions is carried out by PCs in plants [53] and MTs in animals [71]. In fact, plants make use of all three—PCs, MTs, and

GSH—while most animals deal with heavy metals only by using GSH and MTs [71]. It has even been found that PCs are not necessarily the major ligands of Cd^{2+} [89]. Until the discovery by Lane, Kajioka, and Kennedy in 1987, MTs were thought to exist only in animals, with PCs being the plant equivalent [76]. However, shortly after the discovery of PCs' protective function against cadmium, an MT protein was identified in wheat [76] and it was found that PCs, GSH, and MTs work in tandem with each other to convey the ability to protect against heavy metals. Vice versa, PCS genes have also been identified in animals. In particular, the PCS genes of the nematodes *Cænorhabditis elegans*, *Cænorhabditis briggsæ*, and the slime mould *Dictyostelium discoideum* deserve special mention as the proteins they encode actually confer PC synthase activity.

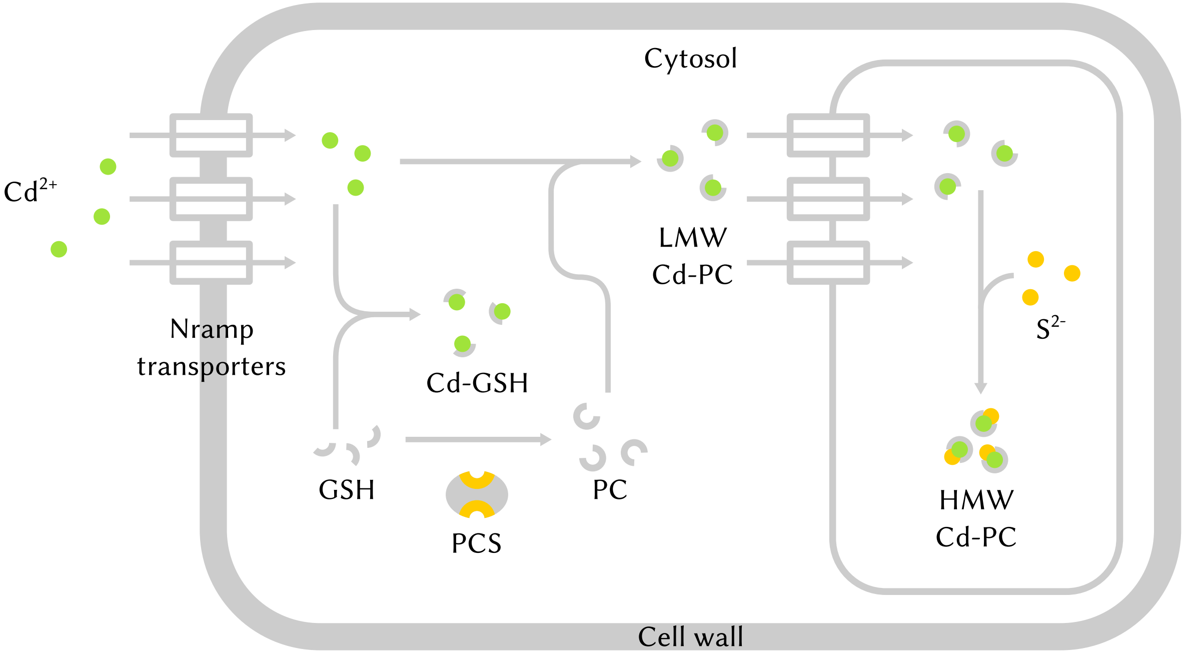
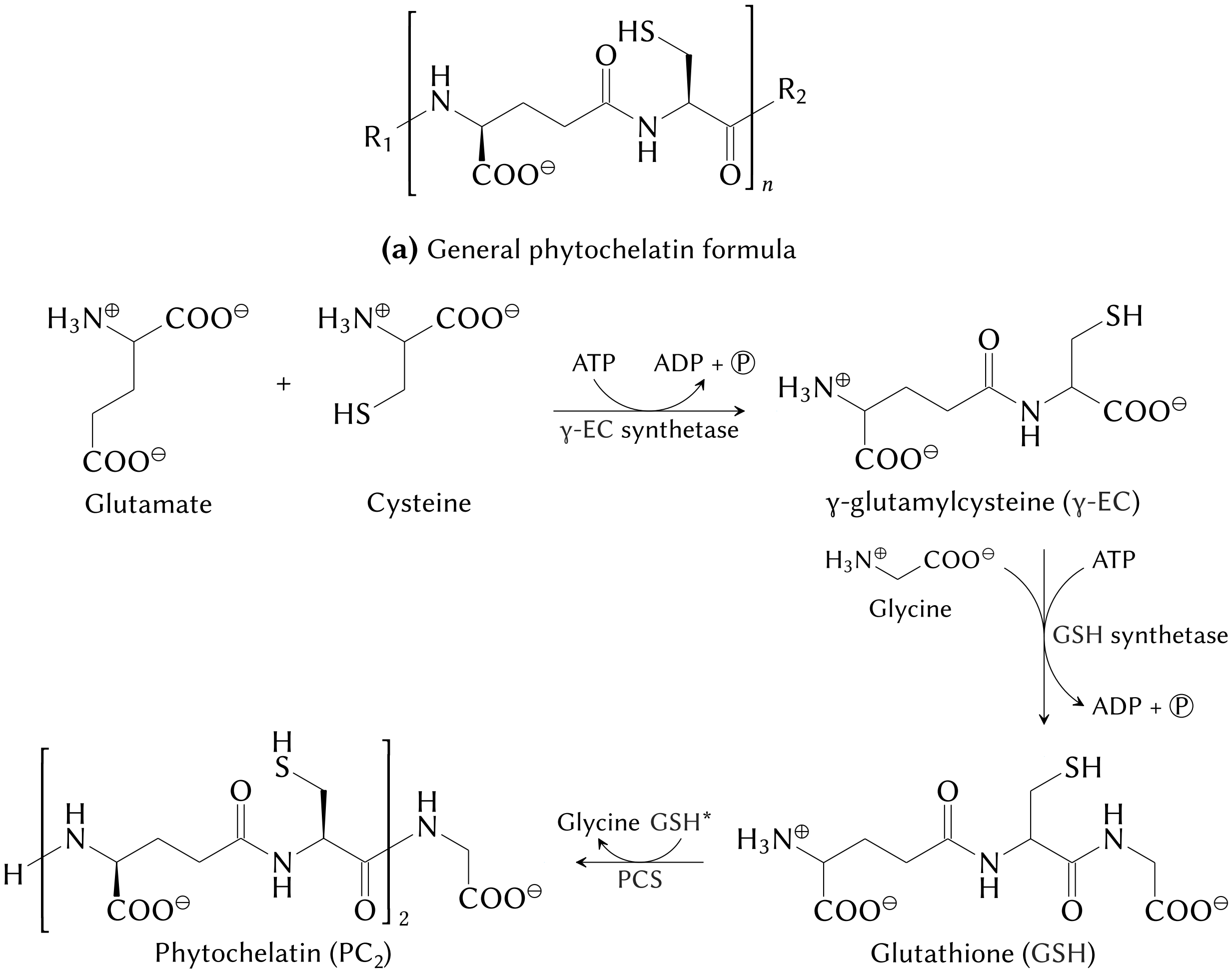


Figure 1.3. Detoxification mechanism by phytochelatins.

The role of PCs in cellular heavy metal detoxification is closely coupled to that of GSH. After entering the cytosol, Cd^{2+} ions are first chelated by GSH and PCs. These initial complexes serve to activate PCS to produce more and longer PCs. Those that are not bound by PCS are sequestered to vacuoles where they are rendered harmless by being incorporated into high molecular weight (HMW) complexes. Once the influx of toxic cadmium ions is lessened and the ration of free PC to $PC \cdot Cd^{2+}$ complexes reaches a certain threshold, the free PC molecules serve as a feedback inhibitor to PCS, indicating that the immediate threat is over. For an overview on the detoxification mechanism by PCs, see Figure 1.3.

PCs are synthesized either from two molecules of GSH, resulting in PC_2, or a molecule



(a) General phytochelatin formula

(b) Phytochelatin biosynthesis pathway

Figure 1.4. (a) General formula of PCs. R_1 = H, Cys; R_2 = OH, Gly, Ser, Glu, Gln, Ala. $n = 2 \cdots 11$, usually between 2 and 5. (b) Phytochelatin (PC_2) biosynthesis pathway, starting from commonly available amino acids. Longer PCs can be synthesized by exchanging one or both GSH molecules in the last, PCS-synthesized step by PCs; for example, $PC_2 + PC_2 \xrightarrow{PCS} PC_4$. The GSH* in the last step is a GSH molecule with its thiol group blocked [164], most commonly by a heavy metal such as Cd^{2+}.

of GSH and a molecule of PC_n, resulting in PC_{n+1}. Both reactions are catalyzed by the same enzyme, phytochelatin synthase, which will be examined further in the next section. The general structure of a PC can be seen in Figure 1.4a. It consists of n γ–Glu−Cys subunits, where $n = 2 \cdots 11$, and a terminal Gly residue. The Cys residues' thiol groups are responsible for the chelation of metal ions, as mentioned above (Section 1.1). Because a single Cd^{2+} ion can bind several chelating thiol groups, $PC \cdot Cd^{2+}$ complexes do not have a defined stoichiometry.

Additionally, $PC \cdot Cd^{2+}$ complexes exist in two distinct forms with different functions, depending on their cellular localization [63]: Low molecular weight (LMW) complexes occur in the cytosol and are ready to be sequestered into vacuoles or out of the cell. Their binding capacity for Cd^{2+} is comparatively low, yet they are crucial as they activate PCS and lead to the production of more PCs. The HMW complexes located in the vacuole—or in plastids for *C. reinhardtii*—are more stable than LMW complexes, and have an increased Cd^{2+} to PC ratio [121]. They are created from sequestered LMW complexes and sulfides, which stem from cysteine sulfinate and are supplied by enzymes from the purine biosynthesis pathway [69, 150]. The HMW complexes confer the actual detoxification ability [98, 110]. Structurally, they are assumed to take the form of a $Cd^{2+} \cdot S^{2-}$ crystal encased in PCs of different lengths. The ratio of Cd^{2+} to PC also differs at different values of ionic strength [53].

1.3. Phytochelatinsynthase

The last reaction in Figure 1.3, the formation of PC_{n+1} catalyzed by phytochelatin synthase (PCS), is main topic of interest in the present thesis. PCS is constitutively expressed in the cytosol [30, 33] and has been demonstrated to occur in three different kingdoms [54]. While PCs have been found in several higher plants, PCS or PCS-like genes have been found in a surprisingly large range of organisms and are almost ubiquitous in higher plants [30]. This widespread availability of PCS genes in nature is assumed to occur due to both horizontal and vertical gene transfer [31].

1.3.1. Structure

Structurally, PCS (EC 2.3.2.15) is a member of the papain superfamily [170] whose members in turn are a group of cysteine proteases [124]. It usually occurs as a dimer or tetramer [54, 170] although monomers have also been found [112]. Eukaryotic PCSs have an N-terminal catalytic and a C-terminal regulatory subunit, while the prokaryotic "half-size" PCSs lack the C-terminal subunit ([56], see Figure 1.5 for an example). A 3D model of the only available

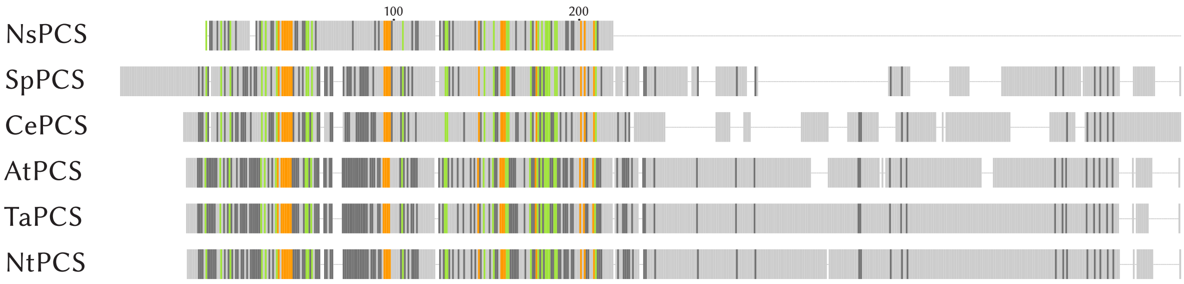


Figure 1.5. Alignment between several eukaryotic and prokaryotic PCSs. Positions where all residues match are indicated in green, positions which are identical for at least 50 % of sequences are dark gray, sequences with lower matching ratio are light gray, gaps are black lines. Active site residues are highlighted in orange (all active site residues are conserved between all displayed sequences). The only prokaryotic PCS, NsPCS, lacks the C-terminal regulatory domain. CePCS is the only animal PCS in this alignment and has relatively low sequence identity to the other PCSs. NsPCS, phytochelatin synthase of *Nostoc spec.*; SpPCS, phytochelatin synthase of *Saccharomyces pombe*; CePCS, phytochelatin synthase of *Cænorhabditis elegans*; AtPCS, phytochelatin synthase of *Arabidopsis thaliana*; TaPCS, phytochelatin synthase of *Tuber æstivum*; NtPCS, phytochelatin synthase of *Nicotiana tabacum*. For the full-length alignment, see Appendix A.2.

structure of a PCS at the time of writing, the prokaryotic NsPCS, can be found in Figure 1.6.

The active site is in the N-terminal domain depicted in Figure 1.6 and Figure 1.7. As its most vital residues—those of the catalytic triad and the oxyanion hole—are widely conserved among all known PCSs (see Figure 1.5 and Appendix A.2), and the reaction mechanism of PCSs has been elucidated, it can be assumed that the active site of eukaryotic PCSs looks similar to that of NsPCS. The active site of 2BTW consists of the catalytic triad, its associated residues, and two substrate binding sites. Both substrate binding sites are necessary for the full reaction to be carried out; however, the first substrate binding site suffices for γ-glutamylcysteine (γ-EC) cleavage (see next section).

At the time of writing, no structural data on the C-terminal domain has been published, but it is assumed to play a role in metal sensing and activation (see Section, [168]).

Ruotolo et al. propose that the C-terminal domain is not as compact and stable as the N-terminal domain and does not fold autonomously [133]. The most upstream part of the C-terminal domain is required for PCS stabilization. The downstream part is responsible for interaction with other species of heavy metal ions. According to Ruotolo et al., it is conserved in a lineage-dependent fashion, thus providing a selective advantage by defining the range of heavy metals to which the enzyme is responsive [133]. Several cysteine residues have been identified in the C-terminal domain which could confer that ability (see Figure 1.5),

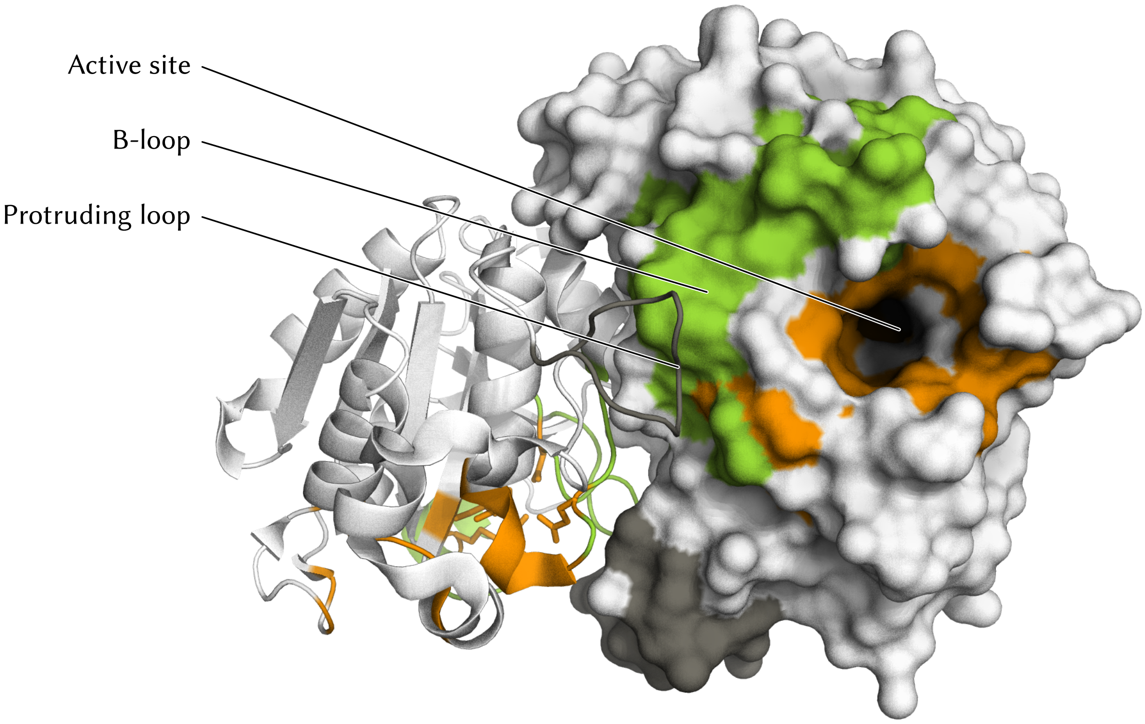


Figure 1.6. 3D structure model of phytochelatin synthase of *Nostoc spec.* (2BTW as found by Vivares, Arnoux, and Pignol [170]. The two chains are shown as different representations; chain A is represented as a cartoon, chain B as a surface. Orange colored residues are part of the active site, green residues are part of B-loops, and the protruding loop is in dark gray. The tunnel on the surface to the right is the substrate binding site.

and homology to thioredoxin and MTs has been established [88, 90].

1.3.2. Functions and Mechanism

The apparent main function of PCS in eukaryotes is the synthesis of PCs from GSH and PCs [54]. Longer PCs can bind heavy metals at higher affinity and $\mathrm{PC} \cdot \mathrm{Cd}^{2+}$ complexes can be sequestered to “safer” locations, as mentioned above. However, the synthesis of PCs, while an integral part of heavy metal detoxification, is not the “original” main function of PCS [30, 124]. Two facts hint at one or more auxiliary functions: Firstly, prokaryotic “half-size” PCSs like NsPCS hardly catalyze the formation of longer PCs [49]. Secondly, PCS is a constitutively expressed, widely available enzyme despite the recent anthropogenic occurrence of large heavy metal concentrations. Among others, PCS enzymes have been identified to play roles in zinc homeostasis [31], GSH-conjugate catabolism [16, 40, 123], plant innate immunity [170], and xenobiotic resistance [28].

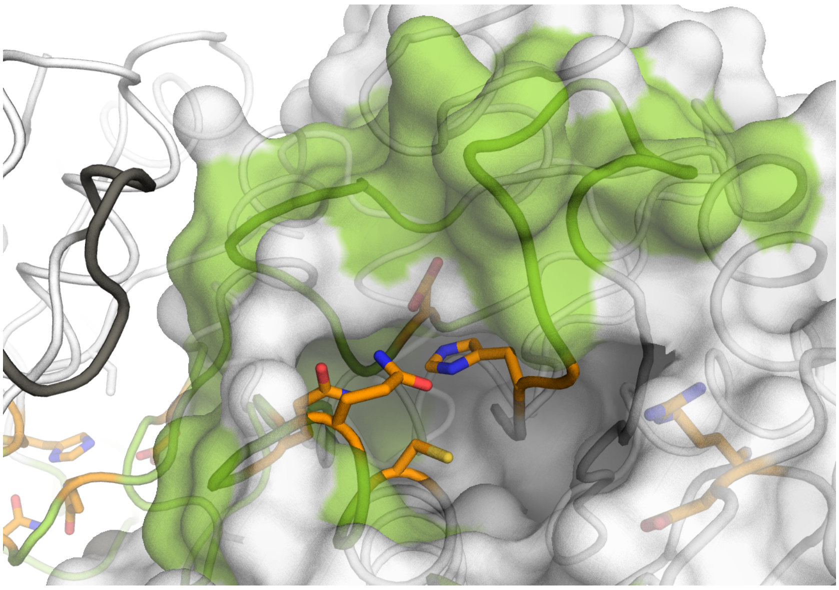


Figure 1.7. Close-up view of the NsPCS active site. Active site residues, which are those residues involved in the binding of the substrate, are colored orange, b-loops are green, and the protruding loop is in dark gray.

The general reaction catalyzed by PCS is the last one depicted in Figure 1.3: The transfer of a γ-EC moiety onto a GSH or PC molecule via an acyl-enzyme intermediate [54]. The reaction consists of two steps. First, the enzyme acts as a *peptidase* by cleavage of the Cys-Gly bond and formation of the acyl-enzyme intermediate:

$$\gamma\text{–Glu–Cys–Gly} + \text{PCS} \longrightarrow \gamma\text{–Glu–Cys–PCS} + \text{Gly}$$

Second, a *transpeptidase* reaction with a thiol-blocked GSH/PC molecule results in the release of PC_{n+1} and the reconstitution of the active site:

$$\gamma\text{–Glu–Cys–PCS} + (\gamma\text{–Glu–Cys}^{*})_{n}\text{–Gly} \longrightarrow \text{PCS} + (\gamma\text{–Glu–Cys}^{*})_{n+1}\text{–Gly}$$

Three residues forming a *catalytic triad* are involved in both reaction steps, namely cysteine, histidine, and aspartate. In 2BTW/2BU3, those are Cys^{70}, His^{183}, and Asp^{201} [165, 170]. This catalytic triad is present in all serine and cysteine proteases and is conserved among all known prokaryotic and eukaryotic PCSs [126].

The binding of the donor GSH molecule leads to conformational changes in the protein which, in turn, result in the formation of a low-barrier hydrogen bond between the imidazole ring of His^{183} and the carboxyl group of Asp^{201} (step 1, [29]). This hydrogen bond increases the basicity of the His^{183} δ-nitrogen, which can now abstract a proton from the Cys^{70} thiol side chain, turning it strongly nucleophilic (step 2). The thiolate ion attacks the substrate carbonyl carbon in an S_N2 reaction, leading to the formation of the first tetrahedral intermediate (step 3). This intermediate is stabilized by backbone carbonyl oxygens of adjacent residues and forms an oxyanion hole, like in serine and cysteine proteases. Due to the conformational changes induced by substrate binding, the glycyl amide nitrogen is a better leaving group than the thiol sulfur and is cleaved off upon abstracting a proton from His^{183} (step 4). Thus, the chemically stable acyl-enzyme complex is formed (Figure 1.9). As the newly formed glycine leaves the active site, the acceptor GSH/PC takes its place. His^{183} again abstracts a proton from the acceptor N-terminal amine group, making it nucleophilic. This leads to an S_N2 attack of the acceptor amine on the thioester carbonyl carbon and to the formation of the second tetrahedral intermedate (step 5). Subsequently, the thioester bond is cleaved and Cys^{70} regains its proton from His^{183}, leading to the protein's original state (step 6).

Despite its primary function in heavy metal detoxification, no metal ions are actually required for PCS to perform its function. Albeit the "canonical" GSH/PC and their metal thiolates are required for maximum catalytic efficiency, the necessary criterion for PC formation is the presence of a blocked thiol group in the acceptor molecule [164]. With this acceptor, the enzyme forms another acyl-enzyme intermediate at a second acylation site, the exact position of which has not yet been conclusively determined [122]: Experiments on truncated phytochelatin synthase of *Arabidopsis thaliana* (AtPCS) and prokaryotic PCSs—which lack the C-terminal domain and only weakly catalyze the PC formation step [126, 159]—indicate that the second acylation site is located in the C-terminal domain. However, Vivares, Arnoux, and Pignol show evidence of a second acylation site in the N-terminal domain of NsPCS [170].

1.3.3. Regulation and Role of the C-Terminus

PCS can be regulated both at the transcriptional and post-translational level. It is generally constitutively expressed and almost exclusively regulated posttranslationally via interaction with free metal ions and $PC \cdot Cd^{2+}/GSH \cdot Cd^{2+}$ metal complexes [56, 164], although transcriptional regulation has been found in phytochelatin synthase of *Tuber æstivum* (TaPCS) and phytochelatin synthase of *Chlamydomonas reinhardtii* (CrPCS) [18, 32]. The highest enzyme activity is achieved in $0.5\,\mu\text{M}$ Cd^{2+} [164]. However, the protein doesn't usually bind free metal ions; instead, it is activated by thiol-blocked PC or GSH molecules [165, 166] which *in vivo* occur as $PC \cdot Cd^{2+}/GSH \cdot Cd^{2+}$ complexes during heavy metal-induced stress [164, 165]. As mentioned in the previous Section, thiol-blocked PC and GSH molecules also act as substrates. Due to this enzyme-substrate positive feedback loop, PC biosynthesis generally occurs within minutes of Cd^{2+} exposure. In addition to the enzyme-substrate positive feedback loop, phosphorylation at a specific conserved threonin residue in the C-terminus has been found to influence PCS activity [172].

Direct binding of free heavy metal ions can occur at exceptionally high heavy metal concentrations. The binding site for such a direct interaction is assumed to be at a site distinct from the active site [164]. However, the N-terminal binding of free Cd^{2+} can actually inhibit enzyme activity in both prokaryotic and C-terminally truncated eukaryotic PCSs [126]. A direct metal binding site is therefore proposed to exist in the C-terminal domain.

The observed differences between prokaryotic PCSs, which only catalyze the first step of the PCS reaction and which are not affected by heavy metal ions, and eukaryotic PCSs, which catalyze the full PCS reaction and which are affected by complexed metal ions, are a direct result of the presence or absence of the C-terminal domain. It is the binding of complexed heavy metal ions by the C-terminal domain which leads to higher activity in eukaryotic PCSs. According to some authors, the C-terminus is also the location of the second acylation site responsible for PC biosynthesis [126]. An activation mechanism proposed by Tsuji et al. suggests that PCS is initially in an inactive, unfolded state and only folds upon binding of a $PC \cdot Cd^{2+}/GSH \cdot Cd^{2+}$ complex [160]. However, the prokaryotic NsPCS, whose 3D structure was solved by Vivares, Arnoux, and Pignol [170], is able to synthesize PCs. Additionally, truncated AtPCS, i.e. AtPCS without its C-terminus, is also competent in synthesizing PCs [133]. In light of these findings, Vivares, Arnoux, and Pignol propose the location of the second acylation site to be in the N-terminal domain. Accordingly, Tsuji et al. propose that N-terminal conserved sequence regions are associated with the first reaction step while N-terminal regions with low homology interact with heavy metals and/or bind to acceptor molecules, i.e. are associated with the second reaction step [160]. The final location of the second acylation site is therefore not yet conclusively found.

Inhibition of PCS function occurs by the accumulation of free PCs in the cytosol or the depletion of $PC \cdot Cd^{2+}/GSH \cdot Cd^{2+}$ substrate: The fact that free PCs—which are the molecules with the highest Cd^{2+} binding affinity—exist means that no more heavy metal detoxification is necessary; a lack of thiol-blocked PC/GSH molecules interrupts the enzyme-substrate positive feedback loop.

1.4. Motivation and Goal

The goal of this thesis project was to create a viable structure model of CrPCS and verify it using molecular dynamics (MD) simulations in Amber and Gromacs. To that effect, correct and suitable parameters are to be found for those simulations. Additionally, the possibility of crosslinking CrPCS with other enzymes via lysine linkers is to be investigated.

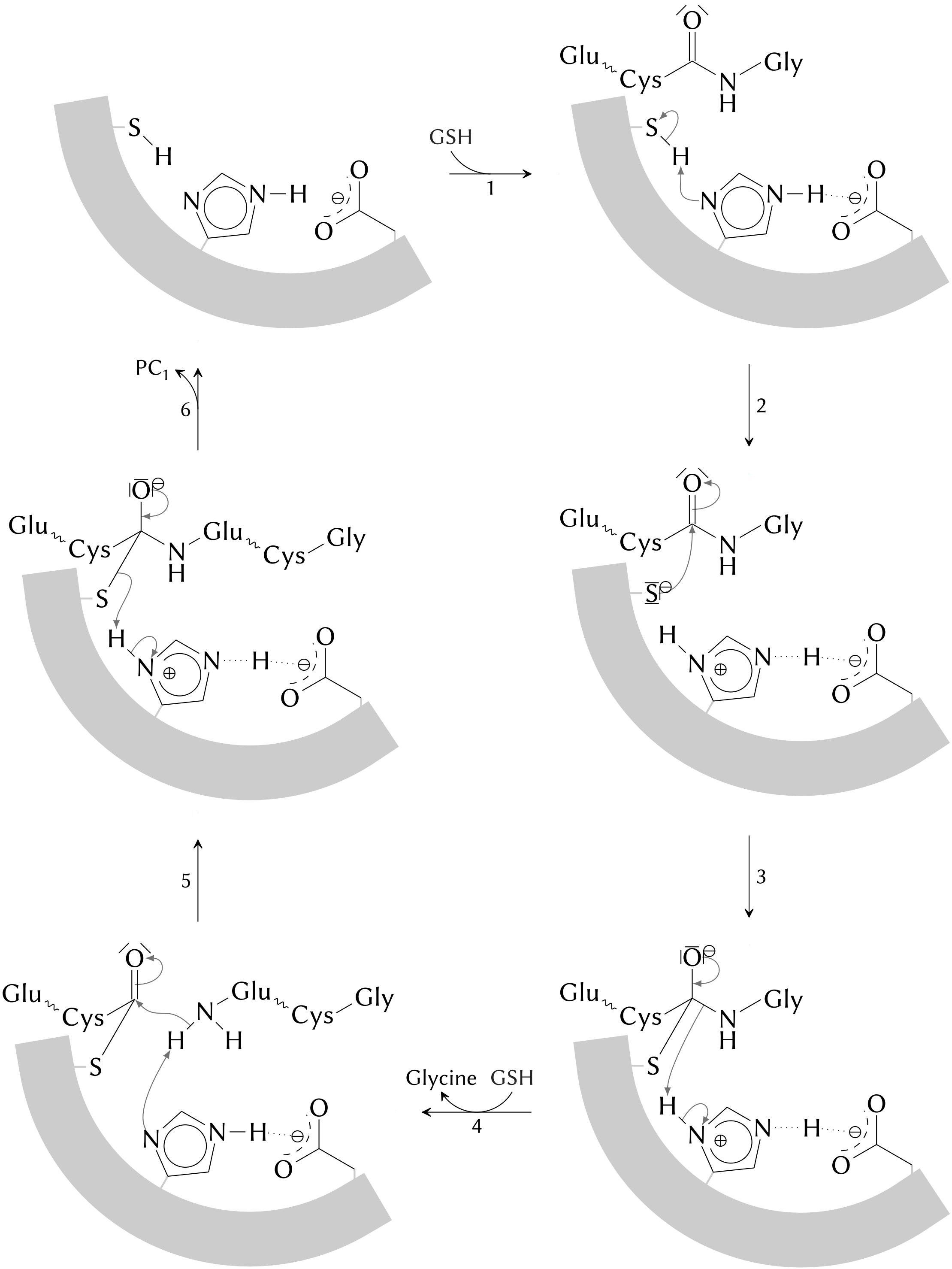


Figure 1.8. Reaction mechanism of phytochelatin synthase. The formation of PC_2 from two molecules of GSH is shown as an example; by replacement of the acceptor GSH by PC_n, the formation of longer PCs is possible. Refer to the text for a detailed explanation of the mechanism.

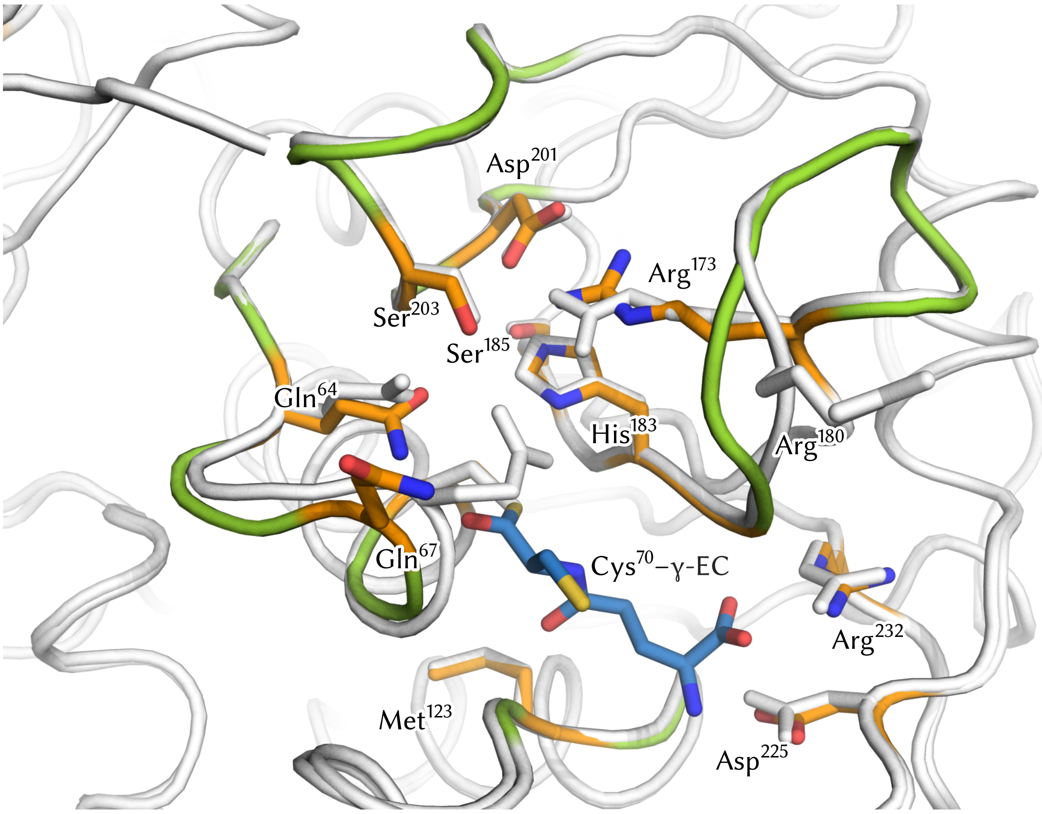


Figure 1.9. Conformational changes in the active site of NsPCS chain B upon formation of the acyl-enzyme complex. Active site residues are orange, the bound γ-EC substrate is in light blue, b-loops are green, and the rest of the molecules are white. The most significant changes are the change in position of Gln^{67}, changing the protein from a "closed" to an "open" conformation; the increased mobility of Arg^{180} in the acyl-enzyme complex, which results in Arg^{180} having a low resolution; and the change of orientation in the Gln^{64} side chain to form the oxyanion hole together with Cys^{70}. Also, the protruding loop (dark gray in Figure 1.7) is disordered in the acyl-enzyme intermediate.

2. Fundamentals

There are several different ways to go about obtaining a valid model of a molecular structure. This chapter outlines the necessary basics to understand the methods used in this project.

2.1. Molecular Modeling

Though the exact ways to obtain a model of a molecular structure by means of homology modeling differ in details, the basic steps are generally the same:

1. Identify templates.
2. Create an alignment between the template(s) and the target sequence.
3. Create model(s) from the alignment.
4. Evaluate and validate model(s) by different means.

Depending on the method used, these steps can partially blend into each other, or additional steps might have to be introduced, or steps could be omitted entirely.

2.1.1. Alignments

The creation of target–template alignments (TTAs) lies at the very heart of the comparative modeling process since errors produced in this step can only very rarely be compensated for later on. Therefore it is imperative that the target amino acid sequence and template structure (or structures) are as well-aligned as possible.

There are two general approaches to create a sequence alignment: Dynamic programming as used in the Needleman-Wunsch [101] and Smith-Waterman [147] algorithms, and heuristic approaches as employed, for example, by the T-Coffee [106] and HHsearch [148] suite of programs. The heuristic approach is favored in template identification due to its superior speed, while the dynamic programming approach is often used when generating sequence alignments between known templates and the target. I will start this introduction with the dynamic programming approaches, however; they came first historically and are the simplest to explain and expand upon.

Dynamic Programming Algorithms

Dynamic programming always leads to the best possible alignment with a given scoring matrix. The most common methods used today are the Needleman-Wunsch algorithm for global alignments and the Smith-Waterman one for local alignments. Both are essentially the same except for a few differences. The Needleman-Wunsch algorithm works as follows:

1. Create an empty array H with the residues of sequence A as rows and sequence B as columns, so that the array is of size $N_A \times N_B$ where N is the number of amino acids in the sequence

2. For each row, determine the score for each cell:

$$H_{i,j} = \max \left\{ \begin{array}{l} H_{i-1,j-1} + w_{A_i,B_j} \\ H_{i-1,j} + w_{A_i,\Delta} \\ H_{i,j-1} + w_{\Delta,B_j} \end{array} \right\} \tag{2.1}$$

3. Starting from the bottom-right cell, trace the highest-scoring path back to the beginning

In Equation (2.1), $H_{i,j}$ is the value of the cell in row i and column j, w is the score for the current substitution, and a Δ index indicates a gap in the sequence alignment.

The score w depends on the chosen scoring matrix. The simplest matrix one could use is the *identity matrix* which assigns a score of 1 for identical residues and 0 otherwise. However, other, more elaborate matrices are often used as well, the most common ones being the Pam250 and Blosum62 matrices. Their scores are based on the notion of *evolutionary distance*: A negative score means a substitution is unlikely, a positive score means it is more likely. The Blosum family of matrices is considered somewhat better than Pam since they were assembled from a more diverse set of sequences. Note that a higher Pam index indicates higher evolutionary distance while a higher Blosum index indicates lower evolutionary distance.

The gap score is usually negative since gaps in an alignment are undesirable. Nowadays, two different values are used for Δ, a high *gap opening* and a lower *gap extension* cost. More elaborate schemes to calculate the gap costs are not uncommon, however. Modeller's salign() algorithm can make use of secondary structure information to increate the gap penalty in helices and sheets, where a gap could lead to the disruption of the entire tertiary structure [87].

The Needleman-Wunsch algorithm always produces global alignments and is therefore not well-suited to aligning e.g. an entire protein with a single homologous domain. For that, the Smith-Waterman algorithm should be used. It has the same steps as the Needleman-Wunsch algorithm but alters the scoring function (Equation (2.1)) so that no negative values can occur. It also changes the traceback mechanism to accommodate for the different scoring pattern.

The Inadequacy of the Dynamic Programming Approach for Template Identification

The number of protein sequences is rising exponentially thanks to recent breakthroughs in sequencing technology. This can be an immense boon to alignment accuracy, as alignments between multiple related sequences can increase the signal-to-noise ratio, leading to higher quality alignments in the process. Bigger databases also lead to better results for template identification in molecular modeling. It would therefore be desirable to be able to sift through these ever-growing databases as quickly as possible.

While the dynamic programming algorithms always identify optimal alignments, they have the disadvantage of running in $O(n^3)$ time and $O(n^2)$ space complexity[1] and are therefore not suited for the "needle-in-a-haystack" problem that is template identification. It is also not straightforward to extend them beyond the realm of pairwise alignment. So, while template identification and sequence alignments share a common basis, their requirements are orthogonal.

Where pairwise or multiple-sequence alignment algorithms value accuracy above all else, an algorithm that is to be used for template identification must favor speed and sensitivity due to the necessity of having to search through potentially large databases of sequences (or related data, such as sequence profiles or HMMs). The goal is to find as many possible templates in a short timespan as possible, with as few false positive results as possible. Afterwards, the found templates are usually realigned with a different MSA algorithm to make

[1] To classify the computation time, one is interested in proving upper and lower bounds on the minimum amount of time required by the most efficient algorithm solving a given problem. The "big O notation" is useful for comparing the resource requirements of different algorithms in terms of input data. Say we have an input dataset of size 100 MB. An algorithm with time complexity $O(n^2)$ would take at worst $100 \cdot 100 = 10\,000$ units of time to complete, while an algorithm with complexity $O(n^3)$ would take $100 \cdot 100 \cdot 100 = 1\,000\,000$ units of time to complete.

up for the inaccuracies introduced by the template identification step. Since no algorithm is perfect, manual adjustments are common as well.

To tackle the speed problem, *heuristic algorithms* have been developed. These algorithms can traverse large sequence databases in a fraction of the time by sacrificing a modicum of accuracy.

The Fasta [114] and Blast [3] programs were the first heuristic search algorithms to be widely used in the field. Blast was also the basis for several improvements, among them the popular programPsi-Blast algorithm [4] and its descendants.

However, while these algorithms addressed the problem of time complexity, they could still only create pairwise alignments. True multiple-sequence alignments have a number of advantages: By adding (at least) an additional dimension to the sequence information, they can indicate which residues in an alignment are conserved and which are more likely to fall victim to the mutation and selection process. Using multiple template sequences/structures also improves the signal/noise ratio when doing comparative modeling.

To tackle the multiple-sequence alignment problem, the classic approach is to create iterative pairwise alignments and rank them according to their evolutionary distance. To that effect, Position-specific scoring matrices (PSSMs) have been created, which are used when aligning more than two sequences: A separate scoring matrix is generated for each position in the alignment, leading to much more sensitive and accurate alignments. These PSSMs, which are at the heart of profile-based sequence alignments, do not necessarily have to be derived from the sequences alone: A number of 3D structure-based approaches to calculate PSSMs exist, and the Modeller suite can make use of secondary structure information in its `salign()` algorithm in order to improve its scoring matrix.

By using PSSMs, the problem of sifting through a vast array of sequences can therefore be reduced to the alignment of a comparatively small number of sequence profiles since the number of protein folds—and, therefore, the number of meaningful sequence profiles—is limited.

Hidden Markov Models While PSSMs lead to an increase in search sensitivity and speed, there are still a number of issues associated with them [44]: They are complicated models with a large number of parameters (for a sequence profile of length 10, ten 20×20 substitution matrices have to be constructed), and gap and insert penalties still have to be assigned empirically. By using profile hidden Markov models (HMMs), all of these issues can be addressed.

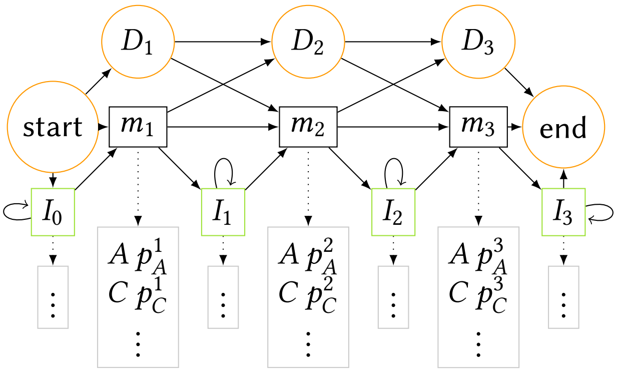
A profile HMM consists of a number of *states* (which, for the purposes of sequence alignment, can be considered sequence matches), *transitions* between these states (each with accompanying probabilities), and values that can be *emitted* at each state (for a PSSM, this would be the residue occurring at a given position in the alignment). The goals when using HMMs are threefold:

1. Given an existing HMM and an observed sequence, how likely is it that the HMM could have generated the sequence?
2. Given an HMM and a target sequence, what combination of HMM states is most likely to result in the target sequence?
3. Given multiple sequences (for example, database of homologous proteins), what is the HMM that will reproduce these sequences' properties most accurately?

An HMM can be considered a nondeterministic finite automaton and as such consists of a number of *states* and *transitions* between them. In the context of sequence alignments, states represent the match at a given position in the alignment: There is either an *insert*, a *deletion*, or a matching residue (which is called an *emission*). A simple HMM for sequence alignment is illustrated in Figure 2.1.

s1	CAF
s2	CGW
s3	CDY
s4	CVF
s5	CKY

(a) Test sequences



(b) Profile HMM for sequences to the left

Figure 2.1. A simple profile HMM, built for the alignment of five sequences of length 3 shown in (a). In (b), m_1, m_2, and m_3 are match states (i.e. positions in the alignment). At each match state, every residue type R occurs with probability p_R, as illustrated for the A and C residues and their respective probabilities p_A and p_C in the black rectangles. The green boxes $I_{0;\ldots;3}$ represent insertions; they also have probabilities for each amino acid attached to them (which are here omitted for brevity). The orange $D_{1;\ldots;3}$ states represent deletions from the sequence. In order to generate a sequence, one has to walk from the "start" to the "end" states.

2.1.2. Comparative Modeling

In contrast to the *ab initio* approach to protein modeling, the comparative modeling approach uses already solved structures as a jumping-off point. The validity of the comparative modeling approach is based on the fact that protein structure is far more conserved than its amino acid sequence: While a large number of protein structures are already known, they adopt a relatively small number of different folds [82]. In fact, Zhang and Skolnick argue that the current contents of the PDB could be enough to solve the structures of all new protein sequences to an accuracy of about 2.25 Å, given algorithms powerful enough to identify the correct templates [180].

Comparative modeling methods can be loosely divided into two different approaches, *homology modeling* and *fold recognition/threading*. For "easier" target sequences with viable templates, the homology modeling methods are usually superior. If no obvious templates can be found (that is, the sequence similarity is at or below the *twilight zone* threshold, which is at about 30 % identity [130]), fold recognition methods might yield better results since they are not solely based on sequence similarity.

Homology Modeling

Homology modeling is based on the correct sequence alignment between the target sequence and the sequence(s) of one or more template structures. (For the sake of clarity, only an alignment of a target sequence with a single template structure is considered.) From this sequence alignment, the target structure is then constructed. Note that the term "homology modeling" is a bit misleading here since the templates used are not necessarily homologous to the target sequence.

For "pure" homology modeling methods, no information besides the sequence alignment is used beyond the initial alignment step. However, modern protein structure prediction methods often use a variety of ways in order to improve alignment quality and cannot be accurately described as either homology modeling or threading methods; most of the time they take bits and pieces of both.

The quality of a sequence alignment is vital to the success of the comparative modeling approach. This is doubly so when using homology modeling, as all subsequent steps directly depend on the quality of the TTA.

There are several ways to go about creating structures for the target sequence. For the sake of brevity, only the *rigid body assembly* and *satisfaction of spatial restraints* methods will be explained in the following paragraphs.

Rigid Body Assembly This historically important method relies on the identification of structurally conserved regions (SCRs) and structurally variable regions (SVRs) in the TTA. When using a sequence profile, these can be easily inferred from the scoring matrices at each position; when HMM-based alignment is used, visualizing the profile HMM using logos [140] can be helpful in identifying conserved regions as well. SCRs can also be assigned from the secondary structure of either the template or the target (using secondary structure prediction methods such as PsiPred [20, 66]).

SCRs in the target sequence can be assumed to adopt largely the same fold and structure as in the template and can therefore modeled as rigid bodies. The modeling process for SVRs is more involved, however, since not enough information is available from the template to create meaningful conformations for them.

Satisfaction of Spatial Restraints A more versatile approach is implemented in the Modeller package [135]. Here, a number of *distance restraints* is used to infer *features* of the target sequence, such as bond lengths/angles, torsion angles, disulfide bonds and more. Then, for each inferred feature, probability density functions (PDFs) are constructed. A PDF is a smooth function which gives the distribution of the feature as a function of the related variables [78, page 541]. The goal is then to find values for these PDFs so that the number of *violations* (i.e. large deviations from expected values) is minimized. This method is inherently flexible since restraints can be inferred from almost any source:

- Target–template alignment
- Basic geometric restraints (bond length, bond angles, torsion angle restraints etc. as used in MD simulations)
- Secondary structure information (limits allowed torsion and bond angles)
- Experimentally determined contacts between residues (e.g. from crosslinking experiments)

To that end, the feature PDFs are assembled to a single molecular PDF:

$$P = \prod_i p^F(f_i) \tag{2.2}$$

Here, P is the molecular PDF and $p^F(f_i)$ are the feature PDFs for features f_i. The best conformation is the one whose PDF violates the restraints as little as possible. To find this

PDF, Modeller minimizes an energy function calculated from the molecular PDF using a conjugate gradients method (Section 2.2.2).

Threading

Not all hope is lost if no good template sequences can be found—the *threading* (or *fold recognition*) approach is still a viable option. It is based on the observation that, even though the number of existing protein structures is vast, these structures adopt only a limited number of folds [82]. Thus there is a high probability that, given a large enough structure database, a protein with a similar fold has already been archived [180]. This limits the usefulness of the threading approach to proteins with known folds but allows creating reliable structures for proteins for which templates only exist deep within the twilight zone.

Threading is at its most basic a very simple approach to protein structure prediction: The target sequence is “threaded” through the backbone of the decoy[2] structure and a scoring function is used to assess the quality of the adopted conformation. After having identified the most promising decoys this way, a more elaborate scoring function can be used to find the best model.

Note that this is just a very basic, intentionally broad description; there is a number of ways in which this very simple method can be improved, both in accuracy and speed.

Better Databases By collecting structures with similar folds into clusters and running the threading search against these clusters instead of individual sequences, a lot of redundant information can be removed from the database. This can be done in a number of ways—the HHsearch method [148], for example, represents both the target sequence and the protein fold database as HMMs (see Section 2.1.1) and matches these against each other. The success of this approach led to the occurrence of a number of “second-generation” databases, among them the Cath [109], Pfam [118], or Scop [97] databases. To pick just an example, the Cath database stores a large number of sequences as HMMs and MSAs.

Faster Alignment Additionally, the actual threading mechanism itself can be sped up by using faster sequence-structure alignment algorithms. A widely used one is the *double dynamic programming* approach [78, page 537]: First of all, a matrix is created which compares the vectors between any two residues ij in protein A to the vectors of any two residues lm

[2] A decoy is a computer-generated protein structure which is designed so to *compete* with the real structure of the protein. Decoys are used to test the validity of a protein model; the model is considered correct only if is able to identify the native state configuration of the protein among the decoys.

in protein B. For each of those vectors, a similarity score is calculated and put into a scoring matrix. The first dynamic programming step is carried out on that matrix in order to obtain the optimal alignment between i and l (i.e. the alignment with the highest similarity score). The similarity scores for each residue are placed in another matrix, upon which the second dynamic programming step is performed. This is the actual “threading” part of the approach.

Another way would be to create a profile HMM to represent the alignment, as is done in the HHsearch approach [148], and use established methods such as the Viterbi algorithm [169] to obtain the correct alignment.

Scoring Function The choice of a good scoring function with high prediction ability is possibly the most vital part of the threading methods. A good scoring function must take into account as much information available as possible while at the same time be small enough to give a result in a reasonable timeframe. This is a hard problem, and so *meta*-threading servers are quite common [12, 116, 179]. They use several different scoring functions and evaluate the combined results. Since different scoring functions have different strengths and weaknesses, combining them can improve the signal/noise ratio.

2.1.3. Clustering

It is postulated that a large number of near-native, partially folded conformations exist near the native state of a protein [145]. Based on this hypothesis, one can argue that, in order to facilitate the folding process, proteins may have evolved native structures within a relatively broad “basin” of low-energy conformations. This can be exploited by *clustering* a large number of potential models into groups of similar structures.

The models created in previous molecular modeling steps are usually already minimized (see Section 2.2.2) and are therefore located in one of the “energy basins” mentioned above. Now, by generating a large number of potential models (*decoys*) and grouping them based on their root mean square deviation (RMSD) to one another, one can observe how the models are split up among the basins. Provided the hypothesis described by Shortle, Simons, and Baker [145] holds, the native structure of the protein is then at the bottom of the basin which is occupied by the highest number of decoys.

The accuracy of this process can be improved by sampling from a larger number of decoys: The more decoys, the better the phase space coverage.

The Average-Linkage Algorithm

The algorithm used in this project for clustering is called average-linkage [142]. It is a *bottom-up* algorithm in which a cluster is created by merging (i.e., *linking*) smaller clusters. In order to choose which two clusters are merged, the algorithms compares the consensus (i.e., *average*) structure The steps are outlined in Algorithm 2.1. For clustering protein decoys, the RMSD is often used as a distance measure.

First, every decoy is placed in its own cluster. Then, the RMSD between all clusters is computed. The two decoys with the lowest RMSD are grouped together to form a new cluster. The last step is to compute the consensus structure of the newly-formed cluster, and to calculate the RMSD of the consensus structure to all other clusters. This process is repeated until either a target cluster number or target RMSD distance threshold is reached.

Algorithm 2.1 Average-linkage algorithm.

```
Require: Target cluster number $n_{\text{max}}$ or RMSD limit $d_{\text{max}}$
  for Decoy $i \leftarrow 1$ to $N$ do
      for Decoy $j \leftarrow 1$ to $N; j \neq i$ do
          Calculate $d_{ij}$
      end for
      Place decoy in its own cluster $c_i$
  end for
  while $n_{\text{clusters}} < n_{\text{max}}$ or $d_{\text{current}} < d_{\text{max}}$ do
      Find and merge two clusters $i$ and $j$ with lowest $d_{ij}$
      Calculate average RMSD of newly merged cluster
  end while
```

Clustering Metrics

There are two basic ways how to choose the granularity of the clustering: Supplying a cutoff, or explicitly specifying a target cluster number for the algorithm to stop.

The cutoff is the maximum distance criterion for which a decoy is still included. For example, if clustering a number of protein decoys with an RMSD cutoff of 1.5 Å, the average-linkage algorithm puts all proteins whose RMSDs are within 1.5 Å of each other in a single cluster. This cluster is removed from the pool and the algorithm starts anew with the remaining decoys. When supplying a target cluster number instead, the algorithm merges decoys into clusters until only the given number of clusters is remaining.

In this project, the cluster number was supplied manually. In order to find the "best" possible cluster number—that is, the number of clusters with the highest *information con-*

tent—a group of similarity and dissimilarity metrics are used: The critical distance, the Davies-Bouldin index (DBI), the pseudo F-statistic (pSF), and the ratio of the sum of squares regression to the total sum of squares (SSR/SST).

Critical Distance The critical distance is the aforementioned cutoff, i.e., the RMSD distance between the consensus structures of clusters to be merged. An "elbow" in the critical distance vs. number of clusters plot indicates an optimal cluster count [142]. For example, if the critical distance resulting from merging two clusters is far larger than that of previous merges, the chosen clusters are disproportionately more dissimilar to each other than the previously merged clusters. Therefore, the best cluster number according to this metric is the one where the "elbow" is.

Davies-Bouldin index This metric quantifies the average distinctiveness of a cluster configuration by comparing the average intra-cluster dispersion to the inter-cluster dispersion [37]. The smaller the DBI, the larger the ratio between intra-cluster similarity to inter-cluster similarity. The larger the DBI, the more similar clusters are to each other. A small DBI is therefore preferrable.

pseudo F-statistic Instead of measuring dispersion, this metric describes the ratio of inter-cluster variance to intra-cluster variance, where variance is measured as the sum of squares of the distances d_{ij} between decoys [23]. It is based on an F-test and measures how well-separated clusters are: The larger the pSF metric, the more tight-knit the clusters resulting from the chosen cluster number. A peak in the pSF vs. number of clusters plot therefore indicates the best cluster number.

SSR/SST This metric compares the sum of squares of decoys within clusters to the sum of squares between all decoys. The sum of squares is calculated separately for each cluster and then summed; this is the SSR. The SST is the sum of squared distances between *all* clusters, not only those within a given cluster. An SSR/SST close to 1 indicates that a lot of the variance between decoys is accounted for by the current clustering configuration. If the SSR/SST is small, only a small amount of the total variance is explained by dividing the decoys into the current clustering configuration. To find the best number of clusters according to this metric, the so-called "elbow criterion" is applied: If merging two clusters results in a steep drop (an "elbow") in the SSR/SST vs. number of clusters plot, this merged cluster configuration does not account for a lot of the variance among the decoys. The best cluster number is therefore on the top of the "elbow".

2.2. Molecular Dynamics

After a model has been created, the next step is to evaluate its validity and compliance with native behavior. Accordingly, molecular dynamics (MD) simulations are often used. They simulate the environment of a protein as accurately as possible. If a protein structure model is stable over a long time in an MD simulation, one can assume it to be of high quality. After all, if only a single atom is placed wrongly, large repulsive forces between nuclei might occur and lead to the blowup of the entire finely-tuned system.

The goal of MD is therefore to simulate a system of particles over time and yield reasonably accurate successive configurations of the system. To that effect, Newton's second law is integrated over time:

$$\boldsymbol{F} = m\boldsymbol{a} \tag{2.3}$$

$$\boldsymbol{F}_i = m_i \frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2} \tag{2.4}$$

$$\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t} = \boldsymbol{v}_i; \quad \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = \frac{\boldsymbol{F}_i}{m_i} \tag{2.5}$$

$$\boldsymbol{r}_i(t) = \int_{t_0}^{t} \boldsymbol{v}_i \,\mathrm{d}t - \boldsymbol{r}_i(t_0); \quad \boldsymbol{v}_i(t) = \int_{t_0}^{t} \frac{\boldsymbol{F}_i}{m_i} \,\mathrm{d}t - \boldsymbol{v}_i(t_0) \tag{2.6}$$

In Equation (2.3), $\boldsymbol{F}$ is the force on a particle, $\boldsymbol{a}$ its acceleration, and m its mass. In Equation (2.4), $\frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2}$ is the second derivative of the particle's position with respect to time, $\boldsymbol{F}_i$ is the force upon the particle, and m_i is again the particle's mass.

Now, in order to obtain the time evolution of a system, the forces $\boldsymbol{F}$ acting upon each of its particles i must be computed. The force that acts upon a single particle is related to the potential energy of the system:

$$\boldsymbol{F}_i = \nabla_i E(\boldsymbol{r}_i) \tag{2.7}$$

Thus the force $\boldsymbol{F}_i$ can be calculated by differentiating the particle's potential energy $E(\boldsymbol{r}_i)$ with respect to its position. How to obtain the potential energy of a particle is subject to the particular *force field* that is chosen for the simulation.

While the actual implementation varies across the popular MD programs, the general procedure stays the same. It is outlined in Algorithm 2.2.

Algorithm 2.2 General force field algorithm.

```
Require: Force field
Require: Coordinates $\boldsymbol{r}_N$
Require: Number of timesteps $n_{\delta t}^{\max}$
  for atom $i \leftarrow 1$ to $N$ do
      Generate velocity $\boldsymbol{v}_i$
  end for
  while $n_{\delta t} < n_{\delta t}^{\max}$ do
      for atom $i \leftarrow 1$ to $N$ do
          Calculate bonded forces $\boldsymbol{F}_i = -\nabla_{r_i}\mathcal{V}(\boldsymbol{r}_i)$
          for atom $j \leftarrow 1$ to $N; j \neq i$ do
              Add nonbonded forces $\sum_j \boldsymbol{F}_{ij}$
          end for
          Calculate kinetic energy $\mathcal{K}(\boldsymbol{v}_i)$
          Calculate acceleration $\boldsymbol{a}_i$
          Update coordinates $\boldsymbol{r}_i$
      end for
      if required then
          Write output coordinates
          Advance timestep
      end if
  end while
```

2.2.1. Force Fields

A force field describes how the potential energy field of a system, $\mathcal{V}$, is calculated. It can be described by a number of summation terms over all particles N in a system:

$$\mathcal{V}(\boldsymbol{r}^N) = \sum_{\text{Bonds}} + \sum_{\text{Angles}} + \sum_{\text{Torsions}} + \sum_{\text{Nonbonded}} \tag{2.8}$$

Besides the ones described here, the introduction of additional terms might be necessary: Improper torsion angle potentials can be used to correct out-of-plane bending motions, while cross terms can account for the fact that bonding terms might influence each other.

Bond Term

The most accurate way to describe the bond energy would be a Morse potential. However, due to its computational complexity and the fact that bonds rarely deviate much from their equilibrium values, a simple harmonic potential is used:

$$E_{\text{Bonds}} = \sum_{\text{Bonds}} \frac{k_i}{2}(l_i - l_{i,0})^2 \tag{2.9}$$

In this potential, modeled after Hooke's law, the bonds behave like springs that oscillate around an equilibrium value. $l_{i,0}$ is the equilibrium bond length, l_i is the actual bond length, and k_i is the force constant. The equilibrium bond length and force constant are different for each bond and for each force field implementation. If higher conformance with the Morse potential is desired, higher-order terms can be incorporated into the equation, turning it into a power series.

Angle Term

The angle potential is a simple Hooke's law harmonic potential, too:

$$E_{\text{Angles}} = \sum_{\text{Angles}} \frac{k_i}{2}(\theta_i - \theta_{i,0})^2 \tag{2.10}$$

The force constants k are, however, considerable smaller than those used in the bond term. As with the bond term, adding higher-order terms improves accuracy at the cost of computational efficiency.

Torsion Angle Term

A torsion angle $\angle ABCD$ is the angle between the planes spanned up by $\angle ABC$ and $\angle BCD$. Its potential is usually defined as a cosine series expansion:

$$v(\omega) = \sum_{n=0}^{N} \frac{V_n}{2} \left[1 + \cos(n\omega - \gamma)\right] \tag{2.11}$$

Here, ω is the torsion angle, V_n is the so-called "barrier height" which is related to the wave's amplitude, n is the multiplicity, and γ is the phase. The higher the potential's order N, the higher the accuracy of the torsion potential.

Besides "normal" torsion angles, it is sometimes necessary to calculate *improper* torsions. Improper torsion terms occur mainly to correct for out-of-plane angle bending motions in aromatic rings and other conjugated systems. They are called "improper" because the involved atoms are not bonded in sequence.

Nonbonded Interactions Term

Nonbonded interactions do not work along specific bonds, but instead through space. Therefore, they are usually modeled as functions of some power of the distance and can be divided into two basic groups, electrostatic interactions and van der Waals interactions [78, page 181]. Generally, nonbonded interactions are calculated between all atoms which are either not in the same molecule or are separated by three or more bonds.

Electrostatic Interactions There are several ways to model the electrostatic contributions to the potential. To ascertain computational efficiency, electrostatics are often modeled as point charges at specific points in the system, usually at the atom coordinates $\boldsymbol{r}_i$. This allows for easy calculation of the Coulomb contribution to the electrostatic potential:

$$\mathcal{V} = \sum_{i=1}^{N_A} \sum_{j=1 \neq i}^{N_A} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \tag{2.12}$$

Here, N_A is the overall number of atoms in the system, q_i and q_j are the point charges of atoms i resp. j, and r_{ij} is the distance between atoms j and j. The simplest model, in which point charges are placed at the coordinates of the atoms in the system, does not consider the contributions of dipole-dipole interactions, or interactions of even higher order such as quadrupoles or octopoles. The accurate incorporation of those would require the *central multipole expansion* method of calculating charges, which comes with its own drawbacks

[78, page 187]. The higher-order terms are instead approximated by using more than one point charge per atom: In order to model the quadrupole in a nitrogen, for example, an additional point charge is placed on the bond between the two nitrogen atoms. Even higher accuracy can be achieved by adding additional charges—however, one must consider that each additional charge increases the computational complexity exponentially since electrostatic interactions are calculated between a comparatively large number of atoms in a system.

The point charge distribution is determined during the parameterization stage of the force field and depends on atom and bond types.

Van der Waals Interactions The nonbonded interactions between some types of atoms cannot be accurately modeled by electrostatic interactions alone. Sufficiently close contact between two atoms can lead to the creation of momentary dipoles through polarization of the atoms' electron clouds. This polarization can in turn lead to weak attractive behavior between those atoms—up to a certain distance at which the repulsive forces between the nuclei become dominant. These dipole-induced interactions are known as *London* or *dispersive* interactions [84].

The dispersive interaction is usually modeled as a 12-6 Lennard-Jones potential:

$$v(r) = 4\epsilon_{ij}\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{2.13}$$

Here, σ is the optimum distance between atoms i and j, and ϵ is the depth of the potential well. The repulsive term is proportional to r^{-12} while the attractive term is proportional to r^{-6}, leading to the characteristic shape of the Lennard-Jones potential.

Final Form of the Force Field Equation

Putting all potential terms into Equation (2.8) yields the final form of the general force field equation:

$$
\begin{aligned}
\mathcal{V}(\boldsymbol{r}^N) = \sum_{\text{Bonds}} \frac{k_i}{2}(l_i - l_{i,0})^2 + \sum_{\text{Angles}} \frac{k_i}{2}(\theta_i - \theta_{i,0})^2 &\quad + \\
\sum_{\text{Torsions}} \frac{V_n}{2}(1 + \cos(n\omega - \gamma)) &\quad + \\
\sum_{i=1}^{N} \sum_{j=1;j\neq i}^{N} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right] &\quad + \\
\sum_{i=1}^{N} \sum_{j=1;j\neq i}^{N} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} &
\end{aligned}
\tag{2.14}
$$

2.2.2. Energy Minimization

Before a force field can be applied to a molecular structure, one or more energy minimization steps must usually be performed to reduce the risk of instability. This is because the criteria required for successful model creation are usually not the same as those for a dynamics simulation.

As an example, consider the situation of a structure created by Modeller that shall be studied in an MD simulation using the `GROMOS96 53A6` force field [108]. The structure obtained by Modeller is created by minimizing Modeller's *objective energy function*, followed by several MD steps in the `CHARMM27` force field. While the `CHARMM27` and `GROMOS96` force fields are similar in their basic form (Equation (2.14)), their actual parameter values differ. Since their parameters are different, their minima probably do not coincide either. Therefore, the minima have to be recalculated for the force field used in the MD simlation, in this case the `GROMOS96 53A6` force field.

Note that, even if a structure was determined by X-ray crystallography, its conformation might deviate from the native structure due to packaging artifacts or inaccuracies introduced during the modeling process. An additional source of error for a homology modeling-derived model is the fact that its conformation is often closer to that of the template (or templates) than to its native structure, which could introduce spots of high energy and increase the risk of simulation blowup.

Therefore, energy minimization is a necessary step prior to all MD simulations.

As already mentioned, the goal of energy minimization is to find the global minimum in the examined system's potential energy surface. A minimum in a derivable function f is a point where its first derivative f' is 0 and its second derivative f'' is > 0. Because the potential energy in the system only depends on the particles' positions, an analytical solution is, in principle, possible. However, due to the fairly complicated nature of modern force fields' energy functions—refer to Equation (2.14) for an illustration—, an analytical approach is not feasible except for very small systems. Instead, several numerical algorithms exist, among them the *steepest descent* and the *conjugate gradients* algorithms, both of which are used in the Amber [24] and Gromacs [14, 58, 80, 163] programs. When applying these algorithms, however, one has to keep in mind that both algorithms can only find the nearest local minimum and are therefore dependent on the starting geometry.

The Steepest Descent Method

The steepest descent method [38] is *robust*, i.e., not susceptible to errors due to a bad starting point. It is therefore useful when a system is far away from its minimum. However, when close to a minimum and the chosen step size λ is too large, the algorithm might not converge but instead oscillate around the minimum due to continuous overcorrection. The following steps are repeated until the energy function converges or until a maximum number of steps is reached:

1. The calculation of the *direction* $\boldsymbol{s}_k$ on the energy hypersurface in which to move the system.
2. Finding the energy minimum along the axis $\boldsymbol{s}_k$ and obtain the new system coordinates $\boldsymbol{x}_{k+1}$.
3. Move the system to the new coordinates $\boldsymbol{x}_{k+1}$.

Step Direction The potential energy $\mathcal{V}$ is a function of the $3N$ coordinates of the system an can therefore be written as $\mathcal{V}(\boldsymbol{x})$, where $\boldsymbol{x}$ is a vector of the $3N$ coordinates. The direction in which to "walk" on the energy hypersurface is then simply the negative gradient of $\mathcal{V}(\boldsymbol{x})$'s first derivative:

$$\boldsymbol{s} = -\frac{\boldsymbol{g}}{|\boldsymbol{g}|} \tag{2.15}$$

Here, $\boldsymbol{s}$ is the direction for the steepest descent algorithm and $\boldsymbol{g}$ is the gradient of the energy function $\mathcal{V}(\boldsymbol{x})$. By dividing $-\boldsymbol{g}$ by its length, a unit vector is produced.

Step Size Once the direction of the next step has been determined, the actual target configuration for the system along that axis has to be found. The target configuration is the point at which the energy is lowest along this coordinate. As usual, there are several ways to solve this problem.

One way is to find three points along the axis so that the middle point is smaller than the end points. Once such a configuration of points has been found, a simple bisection algorithm could be used to locate the actual minimum; or a quadratic function could be fitted against it.

While this proposal is simple in concept, finding three points that satisfy the condition can be computationally demanding. Another, much simpler approach is therefore to take steps of a predetermined length along the vector $\boldsymbol{s}$ and, once the minimum has been crossed, "backtrack" along the axis by a fraction λ of the step length.

The Conjugate Gradients Energy Minimization Method

This algorithm [60] is similar to the previous one in that the basic steps are the same. However, it doesn't exhibit the problem of overcorrection and can find the exact solution (within the given tolerance ϵ) within $3N$ steps at most, where N is the number of atoms in the system.

The difference between the steepest descent and conjugate gradients methods are the ways in which the step direction is found. While the steepest descent algorithm only looks at the gradient, the conjugate gradients method additionally uses information from the previous step vector:

$$\boldsymbol{s}_k = -\boldsymbol{g}_k + \gamma_k \boldsymbol{s}_{k-1} \tag{2.16}$$

Here, $\boldsymbol{s}_k$ is the direction for the current step, $\boldsymbol{g}_k$ is again the gradient at the current step, and γ_k is a scalar that is defined as:

$$\gamma_k = \frac{\boldsymbol{g}_k \cdot \boldsymbol{g}_k}{\boldsymbol{g}_{k-1} \cdot \boldsymbol{g}_{k-1}} \tag{2.17}$$

This places several constraints on the step vectors and gradients:

- The gradients of all steps are orthogonal to each other: $\boldsymbol{g}_i \cdot \boldsymbol{g}_j = 0$.
- The gradients and direction vectors of all step are orthogonal to each other: $\boldsymbol{g}_i \cdot \boldsymbol{s}_j = 0$.
- Each pair of direction vectors is *conjugate* with respect to the Hessian matrix $\mathcal{V}''$.

With this algorithm, it is better to use the line search algorithm to determine the step size. Although either method described in Section 2.2.2 could be used, the high precision of the conjugate gradients step direction pairs itself well with that of the line search mechanism.

Compared to the steepest descent method, the computational complexity for the conjugate gradients mechanism is higher step-wise, however the CG method compensates by taking fewer steps to arrive at the minimum. A common approach is therefore to perform a number of steepest descent minimization steps first and follow up with the conjugate gradients algorithm.

2.2.3. Integrators

The earliest simulations of systems of molecules were done with a hard-sphere model where all collisions were considered elastic and no external forces were applied ($\boldsymbol{F}_i = 0$ in Equation (2.3) [1]). By applying the principle of conservation of linear momentum, the new velocities and positions could be calculated in a fairly simple way. However, demands evolved as high-performance computers allowed modeling more sophisticated systems. The requirements to simulation accuracy increased, and the simple hard-sphere model of interaction was not considered sufficient anymore.

In a real system of molecules, due to mutual interactions between all particles in a system, all manners of forces can act upon a particle. Thus, each particle in the system can be said to exist in a *potential field*, and the total force acting upon a particle is a function of the position of the particle in this field. The coupling of the motions of all particles leads to a many-body problem that cannot be solved analytically. Instead, a *finite difference method* is used to calculate the forces, velocities, and positions of particles at fixed points in time. The accuracy of this approximation increases as the timestep δt decreases. For the general form of the finite difference method, see Figure 2.2.

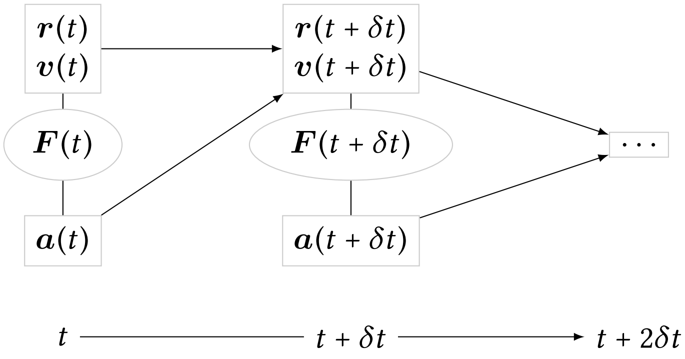

$\boldsymbol{r}(t)$
$\boldsymbol{v}(t)$
$\boldsymbol{F}(t)$
$\boldsymbol{a}(t)$
$\boldsymbol{r}(t + \delta t)$
$\boldsymbol{v}(t + \delta t)$
$\boldsymbol{F}(t + \delta t)$
$\boldsymbol{a}(t + \delta t)$
$\cdots$
t
$t + \delta t$
$t + 2\delta t$

Figure 2.2. General structure of the finite difference algorithm. For each particle, the forces acting upon it are calculated as the vector sum of its interactions with other particles. From the forces, the accelerations at time t can be computed. The positions, velocities of a particles are then used to calculate positions and velocities at $t + \delta t$.
Over the timestep δt, the forces (and, therefore, the accelerations) are assumed to be constant.

Several methods exist to integrate the equations of motion that result from placing the particles in a potential field; some of them shall be described here briefly. All methods require that the positions, velocities, accelerations etc. can be approximated as Taylor series expansions:

$$\boldsymbol{r}(t+\delta t) = \boldsymbol{r}(t) + \delta t \boldsymbol{v}(t) + \frac{1}{2}\delta t^2 \boldsymbol{a}(t) + \frac{1}{6}\delta t^3 \boldsymbol{b}(t) + \frac{1}{24}\delta t^4 \boldsymbol{c}(t) + \cdots \tag{2.18}$$

$$\boldsymbol{v}(t+\delta t) = \boldsymbol{v}(t) + \delta t \boldsymbol{a}(t) + \frac{1}{2}\delta t^2 \boldsymbol{b}(t) + \frac{1}{6}\delta t^3 \boldsymbol{c}(t) + \cdots \tag{2.19}$$

$$\boldsymbol{a}(t+\delta t) = \boldsymbol{a}(t) + \delta t \boldsymbol{b}(t) + \frac{1}{2}\delta t^2 \boldsymbol{c}(t) + \cdots \tag{2.20}$$

In the following paragraphs, only the family of Verlet algorithms [167] will be introduced. More and different algorithms to integrate the equations of motion exist, such as the predictor-corrector methods first employed by Rahman and later refined by Gear [48, 119]. However, the Verlet algorithms are used widely today because of their small computational complexity, their symplecticness (i.e., their ability to preserve the form of the system's Hamiltonian). Because of that, they also lend themselves to the introduction of thermo- and barostats to a system.

The Verlet Algorithm

One of the simplest algorithms is the one proposed by Verlet [167]. It uses a *central difference approximation* of order two: In order to calculate the positions at time $t + \delta t$, the positions and accelerations at t and the positions at $t - \delta t$ are used. This leads to a significantly smaller error complared to the simplistic Euler method (it scales with δt^4 compared to the Euler method's δt) without a large impact on precision, because the third and fourth Taylor term cancel out in the addition step but are still implicitly included in the calculation.

$$\boldsymbol{r}(t+\delta t) = \boldsymbol{r}(t) + \delta t \boldsymbol{v}(t) + \frac{1}{2}\delta t^2 \boldsymbol{a}(t) + \cdots \tag{2.21}$$

$$\boldsymbol{r}(t-\delta t) = \boldsymbol{r}(t) - \delta t \boldsymbol{v}(t) + \frac{1}{2}\delta t^2 \boldsymbol{a}(t) + \cdots \tag{2.22}$$

$$\boldsymbol{r}(t+\delta t) = 2\boldsymbol{r}(t) - \boldsymbol{r}(t-\delta t) + \delta t^2 \boldsymbol{a}(t) \tag{2.23}$$

The "classic" Verlet algorithm has a number of limitations: The positions are obtained by adding a small term $\delta t^2 \boldsymbol{a}(t)$ to a much larger term $2\boldsymbol{r}(t) - \boldsymbol{r}(t - \delta t)$. Due to the limited accuracy of computer floating point calculations, this might lead to a loss of precision.

The precision of the algorithm can decrease further because the Taylor series is usually expanded only up to the acceleration term. However, this can be avoided by choosing a smaller timestep. Last, the algorithm is not self-starting: The velocities for the current step $\boldsymbol{v}(t)$ can only be calculated if the positions for the next step $\boldsymbol{r}(t + \delta t)$ are already known.

All of these issues have been addressed in two variations on the Verlet algorithm: The *leap-frog* algorithm and the *Velocity-Verlet* algorithm.

The Leap-Frog Algorithm

With the leap-frog algorithm, the positions and velocities of a particle system are not computed at the same time. Instead, they are shifted by half a timestep:

$$\boldsymbol{r}(t + \delta t) = \boldsymbol{r}(t) + \delta \boldsymbol{v}\left(t + \frac{1}{2}\delta t\right) \tag{2.24}$$

$$\boldsymbol{v}\left(t + \frac{1}{2}\delta t\right) = \boldsymbol{v}\left(t - \frac{1}{2}\delta t\right) + \delta \boldsymbol{a}(t) \tag{2.25}$$

Knowing $\boldsymbol{r}(t)$ and $\boldsymbol{v}(t + \frac{1}{2}\delta t)$, the positions $\boldsymbol{r}(t + \delta t)$ can be calculated. Thus, the positions and velocities “leap-frog” over each other on the time axis.

The leap-frog algorithm has two advantages over the conventional Verlet algorithm: It doesn’t require the calculation of the difference of large numbers, and it explicitly includes the velocity. However, the positions and velocities are not known at the same time, which means that potential and kinetic energy contributions cannot be accurately calculated at the same time either. Instead, the velocity at a given timestep t is approximated using the following equation:

$$\boldsymbol{v}(t) = \frac{1}{2}\left[\boldsymbol{v}\left(t + \frac{1}{2}\delta t\right) + \boldsymbol{v}\left(t - \frac{1}{2}\delta t\right)\right] \tag{2.26}$$

The leap-frog algorithm is by default used in the Gromacs MD package.

The Velocity-Verlet Algorithm

The Velocity-Verlet algorithm [154] solves all these problems. While the positions at $t + \delta t$ are calculated as in the Taylor series, the velocities $\boldsymbol{v}(t + \delta t)$ are calculated with the average of the accelerations at timesteps t and $t + \delta t$:

$$\boldsymbol{r}(t + \delta t) = \boldsymbol{r}(t) + \delta t\boldsymbol{v}(t) + \frac{1}{2}\delta \boldsymbol{a}(t + \delta t) \tag{2.27}$$

$$\boldsymbol{v}(t + \delta t) = \boldsymbol{v}(t) + \frac{1}{2}\delta t[\boldsymbol{a}(t) + \boldsymbol{a}(t + \delta t)] \tag{2.28}$$

Because the accelerations $\boldsymbol{a}(t + \delta t)$ are not known at the beginning, the Velocity-Verlet algorithm is implemented as a three-stage procedure, which means that the velocity is calculated twice—once at timestep $t + \frac{1}{2}\delta t$, using $\boldsymbol{a}(t)$:

$$\boldsymbol{v}(t + \frac{1}{2}\delta t) = \boldsymbol{v}(t) + \frac{1}{2}\delta t\boldsymbol{a}(t) \tag{2.29}$$

Then, after obtaining the forces and accelerations at timestep $t + \delta t$:

$$\boldsymbol{v}(t + \delta t) = \boldsymbol{v}\left(t + \frac{1}{2}\delta t\right) + \frac{1}{2}\delta t\boldsymbol{a}(t + \delta t) \tag{2.30}$$

Thus the velocities, accelerations and positions are all known at the same time, enabling determination of the kinetic energy contribution to the total energy. The Velocity-Verlet algorithm adds a small overhead as it calculates the velocity twice. This, however, is negligible in the face of the large computational complexity required by the calculation of the forces.

The Velocity-Verlet algorithm is the standard integrator in the Amber MD package.

2.2.4. Constraints

A large timestep is desirable in a simulation as it allows to cover a larger amount of phase space with a given amount of computational effort. However, the timestep must always be chosen carefully to avoid instability in the system: Overlaps between atoms might occur because normally short, localized bond vibrations are sustained over an unnaturally long period of time. The simulation timestep must therefore be significantly smaller than the fastest bond vibration in the system, which—for all-atom systems—is usually that of the C-H bond at 10 fs [78].

In Gromacs and Amber, the fastest bond vibrations—which are usually of little biochemical interest anyway—are replaced with constraints. Thus, the use of a larger timestep is allowed (up to a factor of four [59]). The constraints are implemented as equations which must be satisfied without impacting the overall energy of the system (i.e. the constraint forces may not do any work).

Shake

The Shake algorithm uses N holonomic constraints for a system of N molecules with $3N$ cartesian coordinates. A holonomic bond constraint must satisfy the following equality:

$$\sigma_{ij} = (\boldsymbol{r}_i - \boldsymbol{r}_j)^2 - d_{ij}^2 = 0 \tag{2.31}$$

This ensures that the distance between atoms i and j is always d_{ij}.

The force due to this constraint F_{ck} can be obtained using Lagrange multiplication on the respective cartesian coordinates between the two atoms [78, page 372]:

$$F_{ckx} = \lambda_k \frac{\partial \sigma_k}{\partial x} \tag{2.32}$$

Where x is one of the cartesian coordinates (i.e. it could represent the x, y, or z axis) of one if the atoms i or j and λ_k is the sought-after Lagrange constant for this specific constraint. The force on either atom is opposite in direction to that of the other atom in order to prevent the constraint force from doing any net work.

Incorporating the constraint into the Verlet algorithm yields:

$$\boldsymbol{r}_t(t + \delta t) = 2\boldsymbol{r}_i(t) - \boldsymbol{r}_i(t - \delta t) + \delta t^2 \boldsymbol{a}(t) + \sum_k \frac{\lambda_k \delta t^2}{m_i} \boldsymbol{r}_{ij}(t) \tag{2.33}$$

The highlighted part in Equation (2.33) is identical to the expression in Equation (2.23): The coordinates that would be obtained by “normal” means, i.e. without the use of the Shake algorithm, are perturbed by the constraints’ impositions.

To satisfy the equation, the Lagrange constants λ_k have to be determined so that all constraints k are satisfied. As the number of bonds in a system increases, this becomes progressively more computationally intensive. In order to calculate the constants in a reasonable timeframe, the Shake algorithm iterates over all constraints in turn until all are within a certain tolerance ϵ.

Lincs

The Lincs algorithm [59] attempts to address the following problems that might occur when using Shake:

- Since Shake is iterative and constraints can be coupled to one another, adjusting one constraint might move another bond so far that the method blows up (“Shake errors” in Amber lingo).

- Due to its iterative nature, the algorithm is difficult to parallelize.

By setting the second derivatives of the constraints to 0 and introducing appropriate corrections, Hess et al. managed to reduce the constraint problem to a linear matrix equation. Applying the constraints $g_i(\boldsymbol{r}) = 0$ to Newton's equations of motion yields:

$$-\boldsymbol{M}\frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}t^2} = \frac{\partial}{\partial \boldsymbol{r}}(\boldsymbol{V} - \boldsymbol{\lambda} \cdot \boldsymbol{g}) \tag{2.34}$$

Here, $\boldsymbol{M}$ is the diagonal mass matrix of all atoms in the system, $\boldsymbol{r}$ are the atoms' coordinates, $\boldsymbol{V}$ is the potential field, $\boldsymbol{\lambda}$ are the assorted Langrange multipliers for each constraint, and $\boldsymbol{g}$ are the constraints themselves. Now, the gradients of g in direction of the coordinate axes can be written in matrix form:

$$\mathrm{B}_h i = \frac{\partial g_h}{\partial r_i} \tag{2.35}$$

This simplifies Equation (2.34) to the following matrix equation:

$$-\boldsymbol{M}\frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}t^2} + \boldsymbol{B}^T\boldsymbol{\lambda} + \boldsymbol{f} = 0 \tag{2.36}$$

Solving this equation is considerably faster than iterating multiple times over the Shake constraints while at the same time introducing a negligible error. This also reduces the risk of introducing instability into the system due to large tilts in coupled bonds.

The Lincs algorithm is used in the Gromacs molecular simulation package.

2.2.5. Simulations Under Constant Temperature and Pressure

By default, simulations are run under the *microcanonical* (NVE) ensemble—the number of particles, the volume, and the system's total energy stay constant. However, running simulations under different ensembles can have advantages. By adjusting the temperature, its effects on the system in question can be observed: A protein might unfold, a lipid bilayer might change its fluidity, or a box of water molecules might spontaneously form a crystal lattice. Adjusting the temperature is also a necessary requirement for several methods related to MD, such as simulated annealing or Replica-Exchange MD.

Thermostats

The temperature of a system is a macroscopic property and can only be defined statistically:

$$\langle \mathcal{K} \rangle_{NVT} = \frac{3}{2} N k_B T \tag{2.37}$$

Here, $\langle \mathcal{K} \rangle$ is the average of the kinetic energy of all particles in a system, N is the number of particles, k_B is the Boltzmann constant, and T is the temperature in K.

Velocity Rescaling Thermostats One way to alter the temperature is to adjust the particles' velocities since these directly affect the kinetic energy. The velocities can be adjusted directly [178] or by coupling the system with a heat bath [13]. By simply multiplying the velocities of all particles with a scaling factor, the temperature can be kept constant at each timestep. Using the Berendsen thermostat, the scaling factor λ for the velocities can be obtained like so [78, page 383]:

$$\lambda = \sqrt{1 + \frac{\delta t}{\tau} \left(\frac{T_{\text{bath}}}{T(t)} - 1 \right)} \tag{2.38}$$

Here, τ is a parameter that determines how tightly the system and the heat bath are coupled together, T_{bath} is the target temperature and the temperature of the coupled heat bath, and $T(t)$ is the current temperature. However, this *velocity scaling* approach might lead to trouble when a system with two very different groups of molecules, such as a protein in a water box, is simulated: Over time, the kinetic energy in the system will be distributed unevenly, leading to the problem of "hot solvent, cold solute" [81], also called the "flying ice cube" effect. While the problem can be worked around by scaling the velocities of solvent and solute separately, another issue remains: The Berendsen thermostat doesn't actually sample from a real NVT ensemble, which makes it unfit for some MD applications. It is, however, useful for quickly equilibrating a system to a desired target temperature.

Extended System Thermostats The so-called *extended system method* [62, 104] used in the Nosé-Hoover thermostat works by introducing an additional degree of freedom s to the system, thus modifying its Hamiltonian to conserve temperature instead of total energy [138]. This degree of freedom basically serves as a reservoir of potential and kinetic energy. The magnitude of this reservoir's energy depends on the system's desired target temperature and a coupling constant. Because the additional degree of freedom translates the system into a "virtual" space, the real coordinates and velocities have to be determined by translating the system back into real space, leading to an uneven timestep in the process. The Nosé-Hoover thermostat also samples from the correct NVT ensemble [103].

With the Langevin thermostat [2], Newton's equations of motion are switched out with

Langevin's equations of motion, leading to the following expression for the force acting upon a particle:

$$m_i \boldsymbol{a}_i = \boldsymbol{F}_i - \gamma_i \boldsymbol{\nu}_i + \boldsymbol{R}_i(t) \tag{2.39}$$

Here, $\boldsymbol{F}_i$ is the force from the potential field acting upon the particle, $\gamma_i \boldsymbol{\nu}_i$ is a frictional term that decreases the velocity and, therefore, the temperature, and $\boldsymbol{R}_i(t)$ is a random force that adds to the particle's velocity. The random force $\boldsymbol{R}_i(t)$ is sampled from a Maxwell-Boltzmann distribution at the desired temperature and ensures that the system samples from the correct ensemble.

Barostats

A constant-pressure (NPT) ensemble is obtained by dynamically adjusting a system's volume. This ensemble most closely resembles macroscopic laboratory conditions and is therefore the preferred environment in which to conduct MD simulations.

While the temperature is proportional to the kinetic energy of the system and therefore to its particles' velocities, the pressure is related to the virial $r_{ij}\,\mathrm{d}\mathcal{V}(r_{ij})/\,\mathrm{d}r_{ij}$ [78, page 385]. As the product of the force on an atom and the atom's position, the virial fluctuates much more than the internal energy—it is not uncommon to see pressure fluctuations of 1000 bar and more in a simulation.

Box Vector Rescaling Barostats The methods to keep a constant average pressure are similar to those used to maintain the system's temperature. The Berendsen barostat [13], for example, couples the system to a "pressure bath" and adjusts the volume by a scaling factor λ:

$$\lambda = 1 - \kappa \frac{\delta t}{\tau_P}(P - P_{\text{Bath}}) \tag{2.40}$$

Depending on the chosen scaling behavior (isotropic or anisotropic), either the box vectors are adjusted by $\lambda^{\frac{1}{3}}$ each, or by a scaling factor λ_x where x can stand for each box vector. Again, the Berendsen method of scaling doesn't sample from a known ensemble and is therefore not suited for all MD simulations.

Parrinello-Rahman Barostat While the Berendsen barostat only scales the volume, the Parrinello-Rahman barostat changes the cell's box shape by re-orienting the box vectors [105, 113]. The barostat generates an NPH ensemble, which means the enthalpy $H = E + pV$

stays constant. By also using the Langevin thermostat (or any thermostat which samples from NVT), the generated ensemble is NPT.

2.2.6. Explicit Solvent Simulations

The most straightforward simulation of a protein is the *explicit solvent simulation* in which the molecule is surrounded by a box of solvent molecules.

However, artifacts can occur in such a simple system. The most prominent are due to interactions with the box walls. Nonbonded interactions are especially susceptible to this since they are generally the most long-reaching interactions: They can have noticable effects up to ten molecular diameters or more [78, page 317]. To remove the effect of the walls on the system, *periodic boundary conditions* can be imposed on the simulation. By "wrapping around" the coordinates and interactions around the box walls, the simulated system is effectively infinite.

Periodic Boundary Conditions

Only a small number of possible box shapes lend themselves to periodic boundary conditions: The cube, the truncated octahedron, the hexagonal prism, and the rhombic dodecahedron. While the cube is the simplest to use (no transformations have to be applied when wrapping around the model's cartesian coordinates), it has a relatively high volume and therefore requires a larger number of solvent molecules to simulate. The other shapes have a smaller volume (about 70 % of the cube's volume) and are well-suited to simulating approximately spherical molecules. However, their implementation in software is more difficult, and coordinate transformations have to be applied when wrap-around occurs.

While the use of periodic boundary conditions removes wall interaction artifacts, it introduces another potential problem: If any particle in the system is able to interact with its image in a neighboring cell, the force acting upon it would increase with each time step, leading to the disruption of the entire system. In other words, the *minimum image convention* must be honored: The box size must be large enough that no molecule can interact with its own image. By adhering to this convention, only the unit cell itself and those images that directly surround it have to be considered when determining long-range nonbonded interactions.

Calculation of Long-Range Forces in an Effectively Infinite System

Calculating long-range interactions can result in substantial computational overhead. Especially the Coulomb interactions pose a problem because they decay with r^{-1} (in contrast, van der Waals interactions decay with r^{-6}). There are several ways to solve this problem, the simplest of which is the introduction of cutoffs at a distance where the force of the interactions becomes negligible. However, this can introduce artifacts, such as formation of an ordered lattice of solvent molecules at the cutoff length.

A solution is to split the calculation of nonbonded interactions into a short-range contribution, a long-range contribution, and a constant term [45]:

$$v = v_{\mathrm{dir}} + v_{\mathrm{rec}} + v_{\mathrm{const}} \tag{2.41}$$

v_{dir} is the short-range term and, for the Coulomb potential, is similar to the already established Coulomb term in Equation (2.14). v_{rec} is the long-range term and is modeled as a Fourier sum.

While the original form of the potential converges slowly (with r^{-1} in the case of Coulomb forces), the split terms are made to converge relatively fast in their respective domains. The speed of the Fourier term calculation has been much improved with the invention of the fast Fourier transform algorithm [35].

The Ewald summation method was later extended to the particle-mesh Ewald (PME) method of calculating long-range forces [36], for which a grid of point charges has to be computed over the unit cell. The more points are used, the more accurate the reciprocal space contribution v_{rec} to the long-range potential becomes; however, having to calculate more point charges at each time step is also more computationally expensive.

2.2.7. Implicit Solvent Simulations

Even though state-of-the-art MD programs have developed very efficient code paths for the simulation of bulk solvent, the computational overhead incurred by the addition of a large number of solvent molecules should not be underestimated. Especially the calculation of nonbonded interactions can take a large amount of time. One way to alleviate the computational burden is to forego the addition of (non-catalytic) solvent molecules altogether and instead use an *continuum solvent* method to simulate the solvent environment.

Besides the increased computational efficiency, replacing explicit solvent molecules with a continuum solvent leads to a number of advantages and disadvantages. As no explicit solvent molecules exist, no equilibration of the solvent phase has to be performed prior to a

MD run. Also, by removing solvent viscosity from the simulation, the solute can traverse its phase space more effectively. However, localized contributions by solvent molecules, such as hydrogen bonds to catalytic solvent molecules, cannot be emulated correctly by implicit solvent.

Solvation effects can be written in terms of the solvation free energy ΔG_{sol}, which is the free energy necessary to transform a solute from vacuum into solvent [78, pages 592–601]. Several sources contribute to the overall solvation free energy, which is commonly described as:

$$\Delta G_{\text{sol}} = \Delta G_{\text{elec}} + \Delta G_{\text{vdW}} + \Delta G_{\text{cav}} \tag{2.42}$$

Here, ΔG_{elec} is the electrostatic contribution, ΔG_{vdW} is the contribution by the van der Waals forces, and ΔG_{cav} is the free energy necessary to form the solute cavity. While ΔG_{elec} and ΔG_{vdW} are negative, opening the cavity is work and requires energy, thus making the term ΔG_{cav} positive. The ΔG_{elec} contribution is the most significant one as it is used to model solvent polarization.

The Generalized Born Equation

The *generalized Born* equation is a simple model to obtain the electrostatic contribution to the solvation free energy. In this model, the total electrostatic free energy can be described as:

$$\begin{aligned} G_{\text{elec}} = \sum_{i=1}^{N} \sum_{j=1;j\neq i}^{N} \frac{q_i q_j}{r_{ij}} - \left(1 - \frac{1}{\epsilon}\right) \sum_{i=1}^{N} \sum_{j=1;j\neq i}^{N} \frac{q_i q_j}{r_{ij}} \\ - \frac{1}{2}\left(1 - \frac{1}{\epsilon}\right) \sum_{i=1}^{N} \frac{q_i^2}{a_i} \end{aligned} \tag{2.43}$$

Here, ϵ is the relative permittivity of the implicit solvent, r_{ij} is the distance between the molecules, q_i are the molecules' point charges, and a_i are the molecule's radii. The highlighted part of the equation is the Coulomb potential from Equation (2.14). Note that, since no explicit solvent molecules are present in a GB simulation, the highlighted part is the Coulomb potential of the system *in vacuo*. The non-highlighted terms describe the "solvent" contributions to the electrostatic potential. They comprise the actual GB equation and are usually combined to:

$$\Delta G_{\text{elec}} = -\frac{1}{2}\left(1 - \frac{1}{\epsilon}\right)\sum_{i=1}^{N}\sum_{j=1;j\neq i}^{N}\frac{q_i q_j}{f(r_{ij}, a_{ij})} \tag{2.44}$$

$f(r_{ij}, a_{ij})$ is a function of the distance between atoms i and j and their Born radii a. Using the functional term has a few advantages over the "vanilla" form of the generalized Born equation: It performs well for both charged molecules and dipoles, it is accurate at even long distances r_{ij}, and it is differentiable, allowing to include it in energy minimization methods and MD simulations.

Calculating the Born radii a_i of the atoms works by iteratively adding layers of increasingly thick shells around an atom. The first shell that includes all atoms within the molecule is the effective Born radius.

Nonelectrostatic Contributions

Other contributions to the solvation free energy, ΔG_{vdW} and ΔG_{cav}, are usually combined to a single expression:

$$\Delta G_{\text{vdW}} + \Delta G_{\text{cav}} = \gamma A + b \tag{2.45}$$

A is the solvent accessible area, γ and b are empirically determined constants. The solvent accessible surface area calculation can be sped up by using an approximative algorithm that sums the surface area of all atoms in a molecule and subsequently removes the overlapping surfaces.

3. Methods

The Python programming language [129] was used for scripting and related tasks. The Matplotlib library was used for all plotting endeavours [64]. IPython [117] was used as a Python shell. Pandas [91], SciPy and NumPy [67] were used for data analysis. PyMol [139] was used for structure visualization. Theseus [156–158] and TMalign [181] as well as TMalign's PyMol plugin were used for structural superpositioning.

All modeling steps and MD simulations were carried out on 16-core virtualized workstations with 8 GiB RAM running Ubuntu GNU/Linux.

3.1. Template Identification

To identify potential template structures, the HHsearch [148] algorithm was used. Searches were constructed over the Cath [109], PDB [15], Pfam [118], Scop [97], and Superfamily [51] databases. The initial MSA was generated using the HHblits [125] algorithm. Apart from the alterations described, default options were used; see Table 3.1.

Table 3.1. HHsearch settings for template identification

Setting	Value
Max. MSA iterations	3
Score secondary structure	Yes
Alignment mode	local
Realign with MAC algorithm	Yes
E-value threshold for MSA generation	10^{-3}
Min. coverage of MSA hits	20
Min. sequence identity of MSA hits with query	0
MAC realignment threshold	0.3
Compositional bias correction	Yes

Additionally, manual database searches were conducted in the Pfam, Cath, Superfamily, and Scop databases. However, no useful additional results could be found.

3.2. Sequence Alignment

The initial alignment between PCSs from several species was taken from Vivares, Arnoux, and Pignol [170, fig. 3]. An initial MSA between the target, the template, and other homologous PCS sequences was created using the T-Coffee [39, 106] program. The T-Coffee server uses multiple methods for alignment and combines their output; for this project, all multiple alignment methods (`sap_pair`, `TMalign_pair`, `mustang_pair`, `pcma_msa`, `mafft_msa`, `clustalw_msa`, `dialigntx_msa`, `poa_msa`, `muscale_msa`, `probcons_msa`, `t_coffee_msa`, `amap_msa`, `kalign_msa`, `fsa_msa`, `mus4_msa`, `best_pair4prot`, `fast_pair`, `clustalw_pair`, `lalign_id_pair`, `slow_pair`, `proba_pair`) were used. An initial alignment of the CrPCS sequence to this MSA was generated using T-Coffee. The alignment was subsequently refined in JalView [177] based on the template's secondary structure and the target's predicted secondary structure. For target secondary structure prediction, the PsiPred [20, 65] was used; to analyze the secondary structure of the template, the Dssp algorithm was used [70]. To create the dimeric sequence alignment, the monomeric sequence alignment was expanded to the second chains of both `2BTW` and `2BU3`.

3.3. Modeling

The Modeller program [135] was used to generate the decoys for the target structure. For both monomer and dimer, 2000 decoys were generated and subsequently clustered[1] using the `ptraj` tool of Amber [24]. The nonaligned N- and C-terminal residues (residues 1 to 6 and 217 to 250) were left out of the clustering process as their large RMSD values could lead to an adverse effect on the the clustering results by interfering with the RMSD in the main part of the protein (see Section 4.1.2).

To determine the number of clusters with the highest information content, the guidelines provided by Shao et al. were followed [142]. Refer to Section 2.1.3 and Table 3.2 for guidelines regarding the clustering metrics.

After the number of clusters with the highest information content was determined, the best cluster was selected according to the criteria proposed by Shortle, Simons, and Baker [145]: A large number of decoys in a cluster suggests a broad "valley" in the energy hyperplane which indicates the global minimum and thus the native structure. Therefore, the number of decoys in a cluster was the main criterion. However, if two clusters had a similar number of decoys, the average RMSD of a cluster's decoys to its centroid was used as a secondary

[1] The scripts used for model building and clustering can be found at bitbucket.org/runiq/modeling-clustering and on the accompanying DVD.

Table 3.2. Guidelines for evaluation of clustering metrics

Metric	Criterion for best cluster number
Critical distance	“Elbow” criterion: Abrupt change indicates optimal cluster count
pSF	Maximum where cluster number is manageably small
DBI	Minimum where cluster number is manageably small
SSR/SST	“Elbow” criterion: Abrupt change indicates optimal cluster count

criterion. The selected final structure was the cluster’s representative, which is the decoy with the lowest RMSD to the cluster’s centroid. In order to verify whether the representative structures were viable and in a near-native state, the scores described in the next section were used.

The Apbs program [8, 9, 61] was used to calculate electrostatic potential surfaces in order to assess the dimeric interface. Pdb2pqr [41, 42] was used in conjunction with Amber’s `ff99` [173] to generate initial charges for atoms.

3.4. Structure Assessment

To assess the created decoys and the models generated by the modeling servers, a number of web services were used: The SwissModel structure assessment tool [10, 11, 92], which incorporates the Qmean6 score [11] and the Anolea [92] and Gromos96 [55] scores.

Additionally, models were assessed by Dope score [143], SelectPro score [120], and in Ramachandran plots according to their Φ and Ψ backbone angles using the Rampage server and the backbone angle data from Lovell et al. [85].

Dope is a size-normalized “energy” score. As such, lower values indicate lower-“energy” and therefore better models. Selectpro is size-normalzed as well and can be in the range of 0 to 1, where a score of 1 indicates a native model and a score of 0 a poor one. The Qmean6 Z-score is measured in standard deviations from the average Qmean6 score of all native PDB entries of similar size—a positive score indicates an above-average model, a negative score indicates a model of less-than-average quality. Due to the nature of the modeling process, a positive score generally occurs only with one or more high-quality template(s) and a similarly high-quality alignment.

3.5. Molecular Dynamics

The Amber [24] and Gromacs [14, 58, 80, 163] MD suites were used to perform MD simulations. Explicit solvent simulations were performed with both packages while simulations in implicit solvent were only carried out with Amber; the `GROMOS96 53A6` force field does not supply the parameters necessary for implicit solvent simulations. All simulations were carried out in a 50 mM NaCl solution.

An MD simulation consists of the following general steps:

1. Solvent energy minimization (EM) (explicit solvent simulations only)
2. Whole-system EM
3. Restrained NVT equilibration to the target temperature
4. Restrained NPT equilibration to the target pressure
5. Production run

The basic parameters for these steps can be found in Table 3.4 (Amber) and Table 3.5 (Gromacs).

3.5.1. Model Preparation

After having obtained the models from the clustering process, they had to be prepared for the dynamics simulations. The H++ server version 3.1 [5, 50, 99] was used to set the protonation states of certain amino acids. See Table 3.3 for the settings used.

Table 3.3. H++ settings used for protonation

Setting	Value
Salinity	50 mM
Internal Dielectric	4
External Dielectric	80
pH	7

3.5.2. Amber

For the preparation and analysis of Amber MD runs, AmberTools version 13 was used. The MD runs themselves were carried out using Amber version 12. All simulations were done in the `ff03` force field [43]. The salt concentration in the simulation box was set to 50 mM; for explicit solvent simulations the number of ions to add was determined by using $n = c \cdot V \cdot N_A$ where n is the number of ions to add, c is the target concentration, V is the box volume, and N_A is the Avogadro constant. The Shake algorithm was used to constrain H bonds.

Table 3.4. General settings for an Amber MD simulation

Parameter	Value
Timestep δt	2 fs
Thermostat	Langevin [2] (equilibration)/Berendsen (production)
Barostat (explicit solvent only)	Berendsen
τ_p	$1.0\,\mathrm{ps}^{-1}$
γ_{LN}	2.0
Target temperature	300 K
Coulomb interactions cutoff	8 Å
Van der Waals cutoff	8 Å

Energy Minimization For explicit solvent simulations, the minimization was performed in two steps: First the solvent alone was minimized with the solute being restrained, in order to remove "holes" in the solvent molecule distribution. For the solvent-only minimization, restraints with a force constant of 5 kcal/mol/Å^2 were placed on the solute. All solvent molecules as well as all counterions were minimized. In a second step, a whole-system minimization step was carried out. After 500 steps of steepest descent minimization, a conjugate gradients step was done in order to improve convergence. The convergence criterion was 10^{-4} kcal/mol/Å^2. The actual input files used in the simulation can be found on the accompanying DVD.

For energy minimization in implicit solvent simulations, the solvent minimization step could be omitted since no dedicated solvent molecules were present.

Explicit Solvent Simulations

Due to the concerns raised against simulations in implicit solvent [182], explicit solvent simulations were performed as well, using the TIP3P water model [68]. All explicit solvent simulations were carried out in a system with periodic boundary conditions modeled by the particle-mesh Ewald approach [36] and a nonbonded interactions cutoff of 8.0 Å. The protonated solutes were placed in the middle of a truncated octahedral box, with the smallest distance to a box face being 11.0 Å. The box volume for the monomeric CrPCS structure was 283 621.296 Å, which meant that a total of 9 Na^+ and 9 Cl^- atoms had to be added in order to achieve a salt concentration of 50 mM. The dimer's box volume was 549 156.595 Å, so 17 Na^+ and 17 Cl^- ions had to be added to achieve a 50 mM salt concentration.

***NVT* Equilibration** Equilibration to the desired temperature was carried out in six steps. Starting from 0 K, at each step, the temperature was raised by 50 K and the simulation was performed for 40 ps so that the system would have enough time to equilibrate. A plot of temperature vs. time was used to monitor whether the temperature had converged to its target value. For this equilibration, a 5.0 kcal/mol/$Å^2$ restraint force was placed on all backbone atoms.

***NPT* Equilibration** In a second step, the system's pressure was equilibrated. The *NPT* ensemble is possibly the most "realistic" one as it most closely mirrors lab conditions. The restraint force on all backbone atoms was gradually lowered to 2.0, 0.5, and eventually 0.1 kcal/mol/$Å^2$. Because the pressure generally equilibrates much slower than the temperature (cf. Section 2.2.5), each step was run for 200 ps. The pressure equilibration process was monitored by plotting pressure over time.

Production Run All restraints were then removed from the system and the production run was carried out. The parameters used here are essentially the same as those for the *NPT* equilibration step except for the lack of restraints and the usage of the Berendsen thermostat instead of the Langevin thermostat.

Implicit Solvent Simulations

Simulations using implicit solvent were performed with the GB model by Onufriev, Bashford, and Case [107]. The van der Waals radii were adjusted accordingly to match the `bondi2` parameter set [17], and the cutoffs for nonbonded interactions and Born radius calculation were set to 16 Å.

For EM parameters, see Section 3.5.2.

***NVT* Equilibration** The *NVT* equilibration step is almost identical to the explicit solvent one (cf. Section 3.5.2), except for the use of implicit solvent and a timestep δt of 0.5 fs, as per convention in the lab. A plot of temperature vs. time was used to see if the temperature had converged to its target value.

Restraint Release As implicit solvent simulations do not have a dedicated box around the system, *NPT* equlibration is not necessary. However, the restraints on the system were loosened gradually so as to not introduce instability into the system, in the same fashion as described for explicit solvent simulations (Section 3.5.2).

Production Run As soon as all restraints had been removed from the system, the production simulation started. The parameters are identical to those used for explicit simulations (Section 3.5.2).

3.5.3. Gromacs

Gromacs version 4.6.1 was used to prepare, run, and analyze the MD runs. The force field used in all Gromacs simulations was `GROMOS96 53A6` [108].

The general steps performed were the same as those for the `ff03` force field (cf. Section 3.5.2). However, the values of some parameters (mainly the electrostatic and van der Waals interaction cutoffs) were chosen differently to the Amber parameters, according to best practices described in the Gromacs manual [55] and the `GROMOS96 53A6` force field publication by Oostenbrink et al. [108].

Energy Minimization Gromacs employs steepest descent and conjugate gradients optimization algorithms as well, but it handles them differently: Every 1000 steps, a steepest descent step is included, the rest are conjugate gradients. The minimization stops when either 5000 steps have been performed or when the algorithm converges with a maximum force smaller than $0.01\,\mathrm{kJ\,mol^{-1}\,nm^{-1}}$. The step size for the line search algorithm is 0.1 nm. A grid is constructed for the neighbor list search in order to speed up the simulation. Both the neighbor list and the Coulomb interaction close-range cutoff are set to 0.9 nm.

The Coulomb interactions are modeled using the PME method of Darden, York, and Pedersen [36]. A potential shift is used for the cutoff so as to not introduce artifacts at the border between short- and long-range Coulomb interactions. The van der Waals interactions are modeled using a simple cutoff of 1.4 nm as they drop off quickly with r^{-6}, due to the Lennard-Jones potential used to model them. A similar potential shift algorithm as for

Table 3.5. General settings for a Gromacs MD simulation

Parameter	Value
Timestep δt	2 fs
Thermostat	Velocity-Rescaling (equilibration, [21])/Berendsen (production)
τ	0.8
Target temperature	300 K
Target pressure	1.0 bar
Barostat	Parrinello-Rahman (*NPT* step 2)/Berendsen (*NPT* step 1 and production)
Coupling type	Isotropic
τ_p	1.0 ps^{-1}
Compressibility	4.5×10^{-5} /bar
Neighbor search algorithm	Grid
Neighbor list cutoff	0.9 nm
Coulomb interactions cutoff	0.9 nm
Van der Waals cutoff	1.4 nm
Coulomb interactions model	Particle-mesh Ewald [36]
Potential shift at cutoff for Coulomb interactions	yes
Potential shift at cutoff for van der Waals interactions	yes
Dispersion correction term preserves...	...energy and pressure

the Coulomb cutoff is used in order to avoid artifacts at the cutoff border. Additionally, in order to correct for potential artifacts at the van der Waals cutoff, a dispersion correction term is introduced as suggested by Shirts et al. [144]. As with the Amber explicit solvent simulations, first a solvent-only minimization step is carried out, with the solute restrained.

***NVT* Equilibration** This step is carried out in three distinct parts (in contrast to Amber's six); it has been established that, for presumably stable solutes such as the one modeled in this project, this is sufficient. At each step, the temperature was raised by 100 K. For the first *NVT* step, velocities were generated from a Maxwell-Boltzmann distribution at 1 K. The p-Lincs algorithm [59] was used to place constraints on all bonds which allowed the timestep to be raised to 2 fs. Every step was performed for 25 000 timesteps à 2 fs, which makes the entire *NVT* equilibration take 150 ps. Again, restraints were placed on the solute. A plot of temperature vs. time was used to see if the temperature had converged to its target value. The neighbor list was updated every fifth timestep.

The thermostat used was a velocity-rescaling algorithm. The coupling constant τ was set to 0.8 [78, page 383]. The velocity-rescaling group of thermostats are as stable as the Berendsen thermostat; however, they don't oscillate once arriving at the target temperature. Unlike the Berendsen thermostat, they also sample from the correct NVT ensemble.

NPT **Equilibration** After NVT equilibration was carried out and had successfully converged to the target temperature, the system's pressure had to be equilibrated. Because the pressure generally equilibrates much slower than the temperature (cf. Section 2.2.5), each step was run for an entire nanosecond. Analogously to the NVT equilibration process, a more robust velocity rescaling barostat was first used in order to move close to the target pressure value of 1 bar, with a τ_p constant of $1.0\,\mathrm{ps}^{-1}$. Afterwards, the Parrinello-Rahman barostat [105, 113] was used to make sure the system also samples from the correct NPT ensemble. Apart from the use of a barostat and the length of the simulation, all other settings were the same as in the NVT equilibration step. The pressure equilibration process was again monitored by plotting pressure over time.

Production Run As with Amber, the removal of all restraints signals the start of the actual production run. Additionally, in order to improve performance, the Berendsen thermostat was used instead of the velocity-rescaling algorithm. Apart from these changes, the parameters for the production run were identical to those for the NPT equilibration.

3.6. Investigation of Cd^{2+} Binding Sites

Cd^{2+} binding sites were investigated according to Maier et al. [88] using the alignment from Vivares, Arnoux, and Pignol [170].

The putative binding sites found by Maier et al. were aligned to their corresponding residues in the target protein structure and checked for sequence and structure similarities. It is known that heavy metal ions can have an impact on calcium homeostasis [151] and can block Ca^{2+} channels [153]. Therefore, the Ca^{2+} binding sites present in the templates 2BTW and 2BU3 were checked in the same manner.

4. Results and Discussion

It was already known that the target protein was a phytochelatin synthase of *C. reinhardtii*. A search against the PFAM database [118] resulted in the family PF05023 which lists only phytochelatin synthase of *Nostoc spec.* (PDB IDs 2BTW and 2BU3, UniProtKB ID Q8YY76) as member structures. The two PDB entries represent the same protein: *Nostoc spec.* (UniProtKB ID Q8YY76), the only experimentally solved structure of a phytochelatin synthase in the PDB at the time of writing.

Domain prediction with DomPred [19] results in a single domain for CrPCS with no disorder according to the Disopred server [175, 176].

4.1. Modeling

4.1.1. Template Identification

Manual Template Identification

No motifs have been found in the PROSITE database [146] or in the Minimotif database [95]. However, HHsearch yielded a number of other templates in the 12 % to 15 % range—the best of those was a staphopaïn, a cysteine protease from *Staphylococcus aureus* (PDB ID 1CV8) with 15 % identity and 18 % similarity. Due to their low identity and similarity and the possibility to introduce alignment errors, these structures were not used for model building.

2BTW is a homodimer with a resolution of 2.00 Å, an R value of 0.203, and an R_{free} value of 0.257. 2BU3 is the same macromolecule with a covalently bound γ-EC ligand, a resolution of 1.40 Å, an R value of 0.174, and an R_{free} value of 0.188. The RMSD of 2BTW to 2BU3 is 0.54 Å as reported by TMalign [181].

As 2BTW and 2BU3 are feasible templates, they have been chosen by the majority of modeling servers as well. One has to note, however, that there are a number of caveats with regard to this template:

- NsPCS does not synthesize PCs longer than $n = 2$ [160].

- NsPCS contains only the catalytical, N-terminal domain of the eukaryotic PCS.
- NsPCS is not activated by heavy metals while CrPCS is.
- NsPCS only has one of the five cysteine residues that confer heavy metal-induced enzyme activation according to Maier et al. [88], while CrPCS has four (see Section 4.4).

Template Identification by Modeling Servers

Since the difference in identity between 2BTW/2BU3 and the next best possible template is so large, most of the homology servers also identified them as viable templates. For an overview over the templates chosen, see Table 4.1. The SwissModel server in its automated mode had identified the chain A of template 2BU3 (hereafter called 2BU3a), but the modeling process aborted with an error. Subsequently, the server used 3K8Ua as a template, which resulted in a poor-quality alignment and model (see Figure 4.2).

Table 4.1. Templates chosen by different Modeling Servers

Server	Template
Geno3D	2BTWa
HHpred	2BU3a
I-Tasser	2BTWa (7), 2BU3a (2), 3K8Ua (1)
Loopp	2BU3a, 2BU3b, 2VDCe, 2BTWa, 2BTWb
Multicom	2BU3a
Phyre^2	2BTWa, 2BU3a
RaptorX	2BTWa
SwissModel – automated	3K8Ua

4.1.2. Alignments

Manually Created Alignment

The alignment can be seen in Figure 4.1. The complete alignment between CrPCS and all other PCSs from Vivares, Arnoux, and Pignol [170] can be found in Appendix A.2.

As apparent from the alignment, the N- and C-termini of the target sequence are not aligned to the template. This could lead to problems during the modeling process as a) most template-based modeling methods have to model nonaligned regions using an *ab initio* approach, which is governed by different principles and not always compatible to the rest of the model, and b) termini are difficult to model in general as they are often more or less

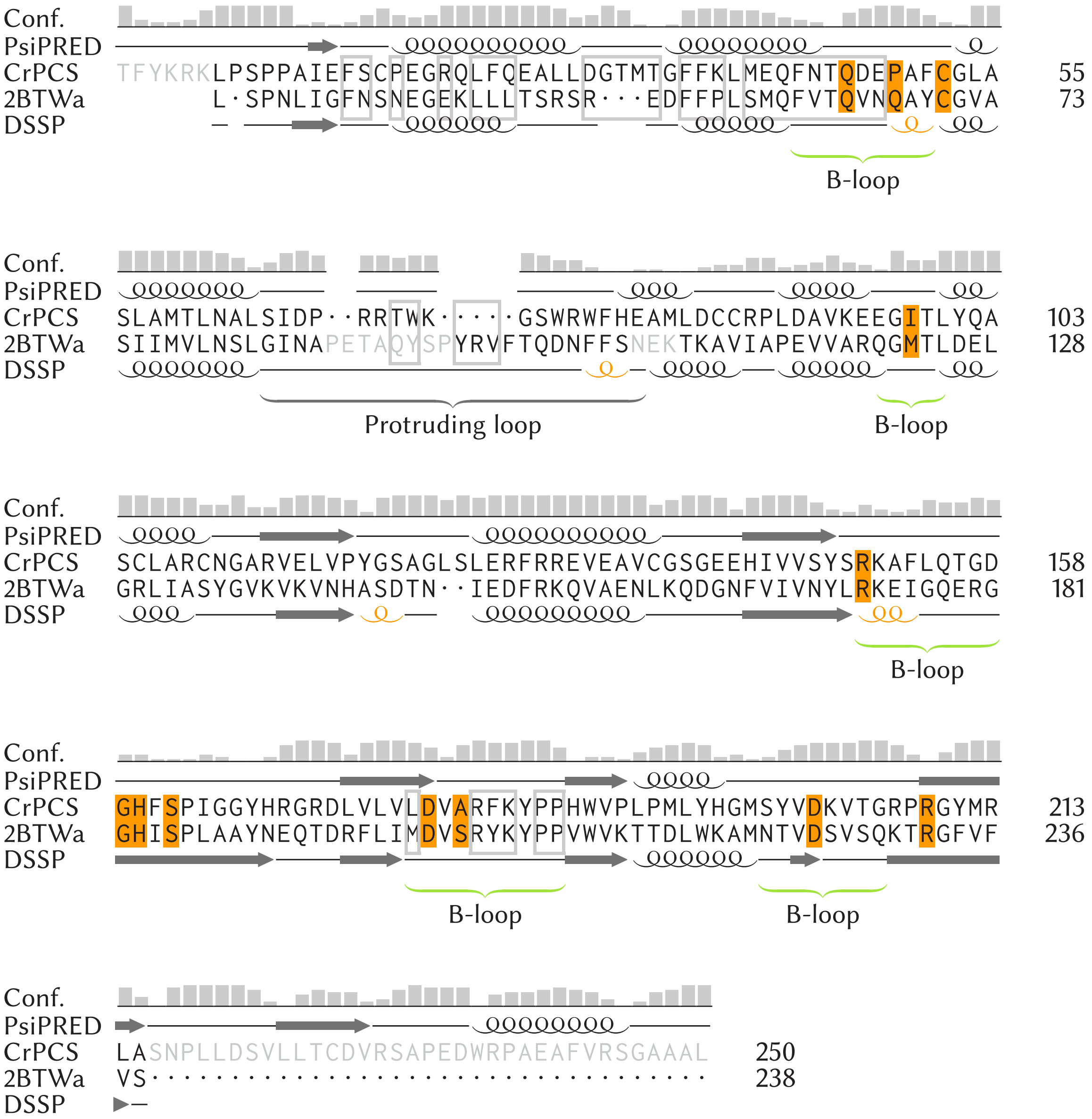

Conf.
PsiPRED
CrPCS TFYKRKLPSPPAIEFSCPEGRQLFQEALLDGTMTGFFKLMEQFNTQDEPAFCGLA 55
2BTWa L·SPNLIGFNSNEGEKLLLTSRSR···EDFFPLSMQFVTQVNQAYCGVA 73
DSSP
B-loop
Conf.
PsiPRED
CrPCS SLAMTLNALSIDP··RRTWK·····GSWRWFHEAMLDCCRPLDAVKEEGITLYQA 103
2BTWa SIIMVLNSLGINAPETAQYSPYRVFTQDNFFSNEKTKAVIAPEVVARQGMTLDEL 128
DSSP
Protruding loop
B-loop
Conf.
PsiPRED
CrPCS SCLARCNGARVELVPYGSAGLSLERFRREVEAVCGSGEEHIVVSYSRKAFLQTGD 158
2BTWa GRLIASYGVKVKVNHASDTN··IEDFRKQVAENLKQDGNFVIVNYLRKEIGQERG 181
DSSP
B-loop
Conf.
PsiPRED
CrPCS GHFSPIGGYHRGRDLVLVLDVARFKYPPHWVPLPMLYHGMSYVDKVTGRPRGYMR 213
2BTWa GHISPLAAYNEQTDRFLIMDVSRYKYPPVWVKTTDLWKAMNTVDSVSQKTRGFVF 236
DSSP
B-loop
B-loop
Conf.
PsiPRED
CrPCS LASNPLLLDSVLLTCDVRSAPEDWRPAEAFVRSGAAAL 250
2BTWa VS································ 238
DSSP

Figure 4.1. Manual alignment of the CrPCS target and the NsPCS template 2BTWA, starting from the alignment presented by Vivares, Arnoux, and Pignol [170]. Residues shaded in orange are part of the active site, those with a gray frame are part of the dimeric interface in 2BTW/2BU3. The major secondary structure elements of CrPCS as predicted by PsiPRED (α-helices, β-sheets, and coils) are shown above the alignment. Above that, PsiPRED's confidence score is displayed. The major secondary structure elements of 2BTWA as identified by the DSSP program (colors as for PsiPRED, 3_{10}-helices in orange) are shown below the alignment. The light gray residues in the CrPCS sequence were omitted from the clustering process and subsequent MD simulations, the light gray residues in the 2BTW template were not present in the 2BU3 structure.

disordered. Nevertheless, no residues were left out of the modeling process in order to find out whether the methods used converge towards a single solution. A visual inspection of several of the created decoys indicated that this was not the case. The residues were then left out of the clustering process as their large RMSD values could potentially lead to an adverse effect on the clustering results by interfering with the RMSD in the main part of the protein. After consulting the advisors, they were also left out of the subsequent MD simulations as they would induce computational overhead without a clear benefit.

However, according to PsiPred, there are secondary structure elements in the C-terminal nonaligned residues (Figure 4.1). Judging from the stability of the model in MD simulations, their influence on protein stability is minor; they do also not have any significant sequence identity to the PCSs of higher plants and therefore do probably not exhibit the functionality of the C-terminal noncatalytical domain found in PCSs of *A. thaliana* or *S. pombe*.

Alignments Created by Modeling Servers

The fingerprints of the alignments generated by modeling servers can be seen in Figure 4.2. The alignment created manually is included for comparison as well. For the sake of brevity, the full alignments have been moved to Appendix A.3. For an overview over the algorithms employed by the different modeling servers, see Appendix A.1. Two algorithms (I-Tasser, Loopp) use multiple alignments between target and template(s) and use them according to the predicted alignment quality. For those algorithms, all alignments are shown.

The most complete models were generated by HHpred, I-Tasser, Multicom and Phyre^2. These servers all modeled the complete sequence, including the N- and C-terminal regions without alignment to the template. The shortest, least complete model was generated by the SwissModel server at 189 residues.

There are two major regions where alignment algorithms disagreed. The first region is the one corresponding to the “protruding loop” in 2BTW/2BU3 and its adjacent residues [170]; the second one is in the range of residues Leu^{116} to Ser^{125}. The protruding loop sequence is well-conserved among eukaryotic PCSs but not so in NsPCS (see alignment in Appendix A.2). It is also notably shorter in eukaryotic PCSs which indicates that it does not play the same role in them as it does in NsPCS. The other region, residues 116 to 125, is not well-conserved among eukaryotic PCSs either, indicating a function different from the one in prokaryotic PCSs.

The active site residues were those which interacted with the γ-EC ligand in 2BU3, and accompanying residues which undergo a conformational change upon γ-EC binding. These residues were—in CrPCS—Glu^{46}, Pro^{49}, Cys^{52}, Ile^{98}, Arg^{150}, Gly^{159}, His^{160}, Ser^{162}, Asp^{178}, Ala^{180},

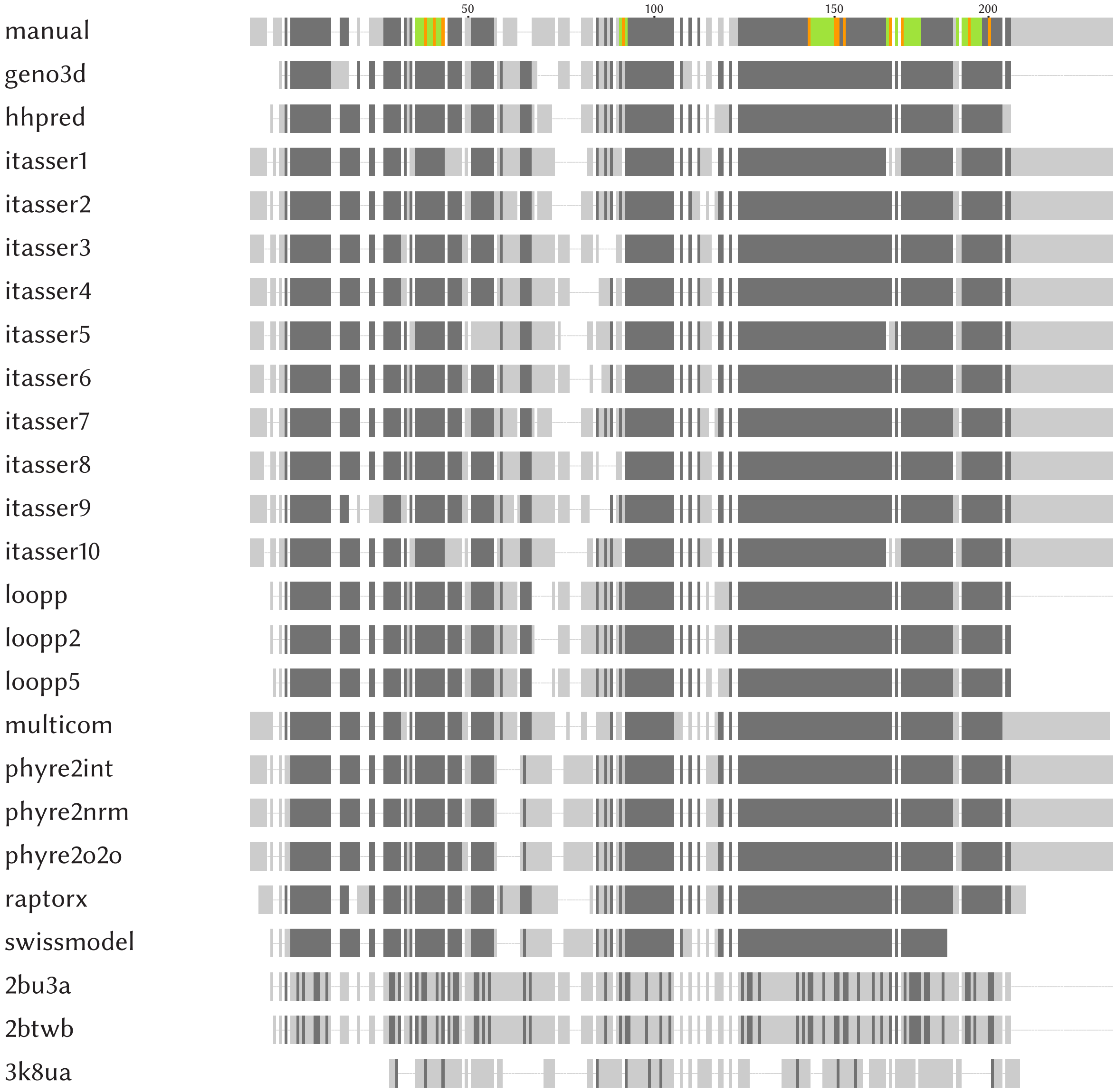


Figure 4.2. Alignments between manual and server-generated models and their templates. The CrPCS sequences were *not* aligned to each other but *only* to their respective templates; this alignment serves as a comparison to identify regions where the alignment algorithms had trouble. Residues which are conserved among more than half of the displayed sequences are dark gray, residues which are conserved in less than 50 % of all displayed sequences are light gray, gaps are indicated by lines. In the "manual" sequence, orange residues are part of the active site, green residues are B-loops.

Asp^{202}, and Arg^{209}.

Almost all active site residues (marked orange in Figure 4.2) were aligned equally among the server-generated alignments. The two exceptions were Ile^{98} and Asp^{178}, which were aligned differently in several I-Tasser models. However, all models which appeared in the "top 10" (Table 4.3) aligned all active site residues in the same manner.

In addition to the active site residues, the regions corresponding to the "B-loops" mentioned by Vivares, Arnoux, and Pignol [170] are assumed to play a role in NsPCS' inability to produce long PCs. In 2BTW, the loops are 61–69 (B-loop 1), 122–124 (B-loop 2), 173–181 (B-loop 3), and 200–209 (B-loop 4). These residues correspond to Phe^{43}–Phe^{51} (B-loop 1), Gly^{97}–Thr^{99} (B-loop 2), Arg^{150}–Asp^{158} (B-loop 3), Leu^{177}–Pro^{186} (B-loop 4), and Ser^{199}–Gly^{206} (B-loop 5). The B-loops were aligned at least similarly, if not identically throughout all models. Aside from several I-Tasser models and the Multicom model, all "top 10" models aligned them in the same position. A single exception must be made for the B-loop residue Ser^{199}, which was aligned differently in the Phyre^2-o2o model.

Assuming the manually generated sequence alignment is correct or mostly correct, this shows that, at 30 % sequence identity, modeling errors resulting from alignment errors should only occur in non-core regions which are not vital for model stability, and that the automatic alignment algorithms are good enough for homology modeling at this quality. The addition of more evolutionally diverse sequences to the alignment could potentially improve the alignment in regions of poor quality.

4.1.3. Model Building

All scores (Qmean6 Z-score, Dope, and Selectpro) were chosen with the ability in mind to compare and assess models of different lengths. Scores for the manual models have been calculated with the non-templated N- and C-termini cut off (see Figure 4.1 and Appendix A.2). As the automated modeling servers also produced models of various lengths (Figure 4.2), truncating the termini in this way was considered to be a part of the manual modeling procedure; which means that the automatically created models were not truncated in the same fashion, but assessed as-is. This procedure was chosen because the Dope score loses its validity when incomplete models are considered.

Manually Created Models

As described in Section 3.3, the generated decoys were clustered according to the guidelines provided by Shao et al. [142]. The clustering metrics used for determining the correct cluster number can be found in Figure 4.3. The critical distance plot indicates that 3 or 5 would be a good cluster number, as there are discernable "shoulders" in the graph at these points. The SSR/SST plot suggests the same cluster numbers. The DBI plot indicates an optimal cluster count of 2 to 4, while the pSF plot is inconclusive. Eventually, a cluster count of 4 was used.

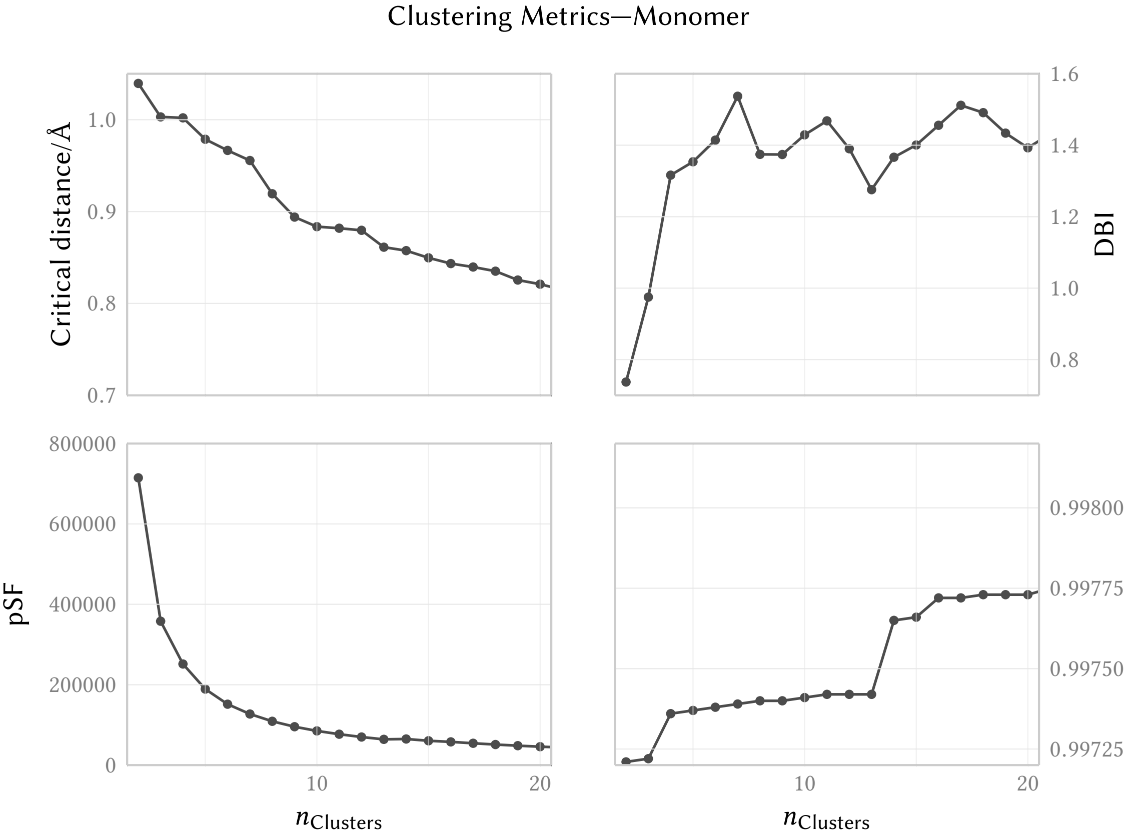


Figure 4.3. Clustering metrics for monomeric model. RMSD, root mean square deviation (Å); DBI, Davies-Bouldin index (dimensionless); pSF, pseudo F-statistic (dimensionless); SSR/SST, ratio of the sum of squares regression to the total sum of squares (dimensionless).

The clusters were evaluated as described in Sections 3.3 and 3.4. Cluster 0 consisted of 1931, cluster 1 of 76, cluster 2 of 2, and cluster 3 of 9. The major criterion is in general the number of decoys in a cluster; therefore, the representative structure of cluster 0 was chosen as the final model (before subjecting it to a C-terminal truncation as discussed in Section 3.3). The Dope, Selectpro, and Qmean6 Z-score of the representative structure of each model can be found in Table 4.2; they indicate that the representative structure is in a near-native state.

The clustering sizes and metrics for the dimer are shown in Figure 4.4. The critical distance plot indicates an ideal cluster count of 3 to 5; the DBI plot suggests one of 2, 3, or 7; and the SSR/SST ration suggests a cluster count of 15. The pSF count is, once again, inconclusive.

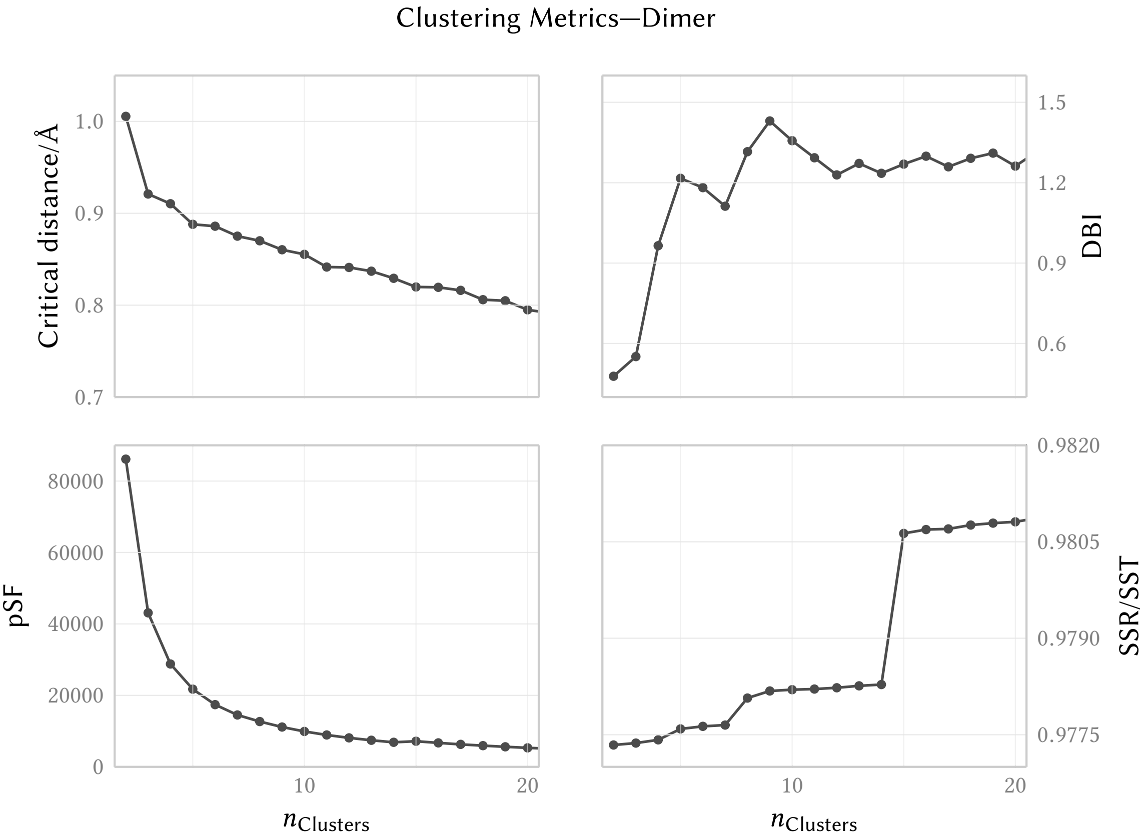


Figure 4.4. Clustering metrics for dimeric model. RMSD, root mean square deviation (Å); DBI, Davies-Bouldin index (dimensionless); pSF, pseudo F-statistic (dimensionless); SSR/SST, ratio of the sum of squares regression to the total sum of squares (dimensionless).

From these metrics, a cluster count of 7 was chosen. The structure assessment scores shown in Table 4.2 confirmed cluster 0 might be close to a near-native structure.

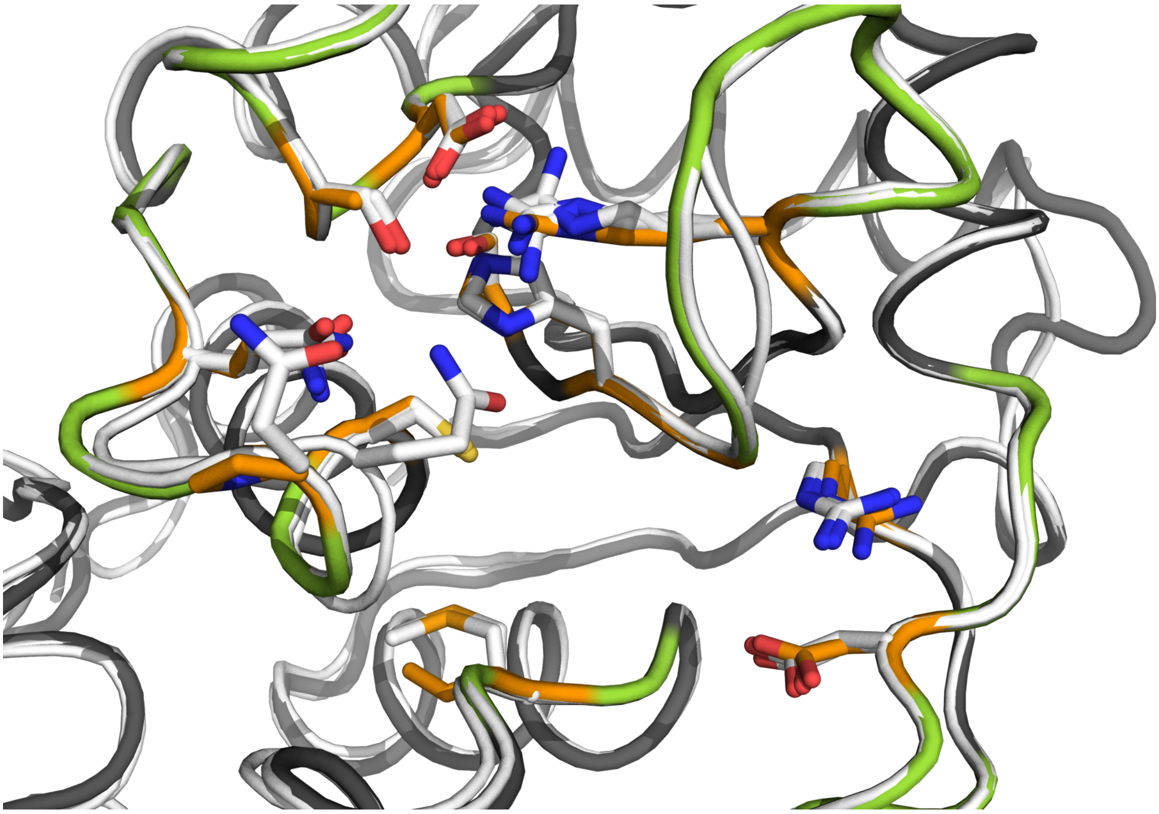


Figure 4.5. Superposition of manual model and templates 2BTWa and 2BU3a. The model is in dark gray, orange (active site residues), and green (B-loops); the templates are in white.

Several conclusions can be drawn from the large number of decoys in the selected cluster. For instance, choosing a small cluster number can potentially lead to a large number of false positives, i.e. structures which are not in the same entropic "valley" as the native structure but in the same cluster. Since about 96 % of all decoys were in the selected cluster, a cluster number of 4 was likely too small: By choosing a larger cluster number or working with an RMSD cutoff instead of an explicit cluster number, this problem could be avoided. However, a smaller amount of decoys in a cluster also results in false negatives, i.e. decoys which *are* in the native structures' entropic valley but are *not* in the same cluster. This would result in a less accurate, less "near-native" choice of the cluster's average and representative structures.

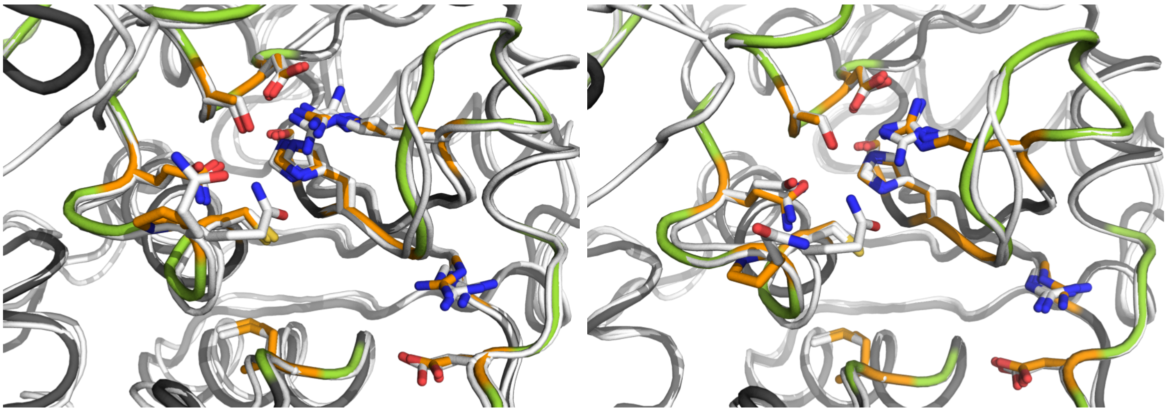
The dimeric models scored worse than the monomeric ones, for a number of reasons. The modeling process for dimers is not as elaborate as the one for monomeric models—dimeric models add another layer of complexity. None of the scores specifically are specifically developed for dimers. For example, the Dope score was calibrated with a spherical model

Table 4.2. Dope, Selectpro, and Qmean6 Z-scores for monomeric and dimeric Modeller models

Model	Cluster ID	Number of decoys	Dope	Selectpro	Qmean6 Z-score
Monomers					
manual-1	0	1931	−1.52	0.658	−1.196
manual-2	1	76	−1.49	0.657	−1.186
manual-3	2	2	−1.41	0.626	−0.397
manual-4	3	9	−1.36	0.655	−1.086
Dimers					
dimer-1	0	1980	−1.33	n/a	−0.976
dimer-2	1	11	−1.15	n/a	−1.186
dimer-3	2	1	−1.29	n/a	−1.741
dimer-4	3	3	−1.23	n/a	−0.684
dimer-5	4	2	−1.29	n/a	−1.320
dimer-6	5	1	−0.84	n/a	−1.328
dimer-7	6	1	−1.18	n/a	−0.972

and scores according to whether supposedly buried residues are actually buried in the model. A dimeric model is not necessarily spherical and the "buriedness" of residues at the dimeric interface is usually not as large as that of actual buried residues. This leads to potential inaccuracies in the scoring of interface residues, which in turn can lead to a lower score. The Qmean6 Z-score basically scores a model on how well it fits into the group of already solved native protein structures in the PDB database. Since a large number of PDB structures are, in fact, small and monomeric, the dimeric CrPCS model does not readily fit into that group. This will be gradually resolved as larger and more difficult protein models are added to the PDB database, improving its diversity.

The active site conformation is well-conserved (Figure 4.5). No residues in either the active site or the B-loops were in unfavored regions of the Ramachandran plot (Figure 4.8a). Two residues are in disallowed regions of the Ramachandran plot, Asp^{30} and Thr^{34}. In general, most of the Ramachandran violations in the manually created models occured around residues 30 to 34 and 80 to 84. The outliers in the 30 to 34-region could be ascribed to a deletion in NsPCS which is not present in eukaryotic PCSs (see Figure 4.1 and Appendix A.2). The outliers in the 80 to 84-region can be ascribed to an error in the manual alignment: Compared to NsPCS, the CrPCS protruding loop is significantly shortened. Therefore, the CrPCS residues should be aligned to the NsPCS residues at the *ends* of the protruding loop in order to be linked correctly and without strain. However, the automatic alignment placed the CrPCS residues in the "middle" of the NsPCS protruding loop, which posed some difficulty for Modeller: The program could not model the residues at the specified positions without introducing strain.

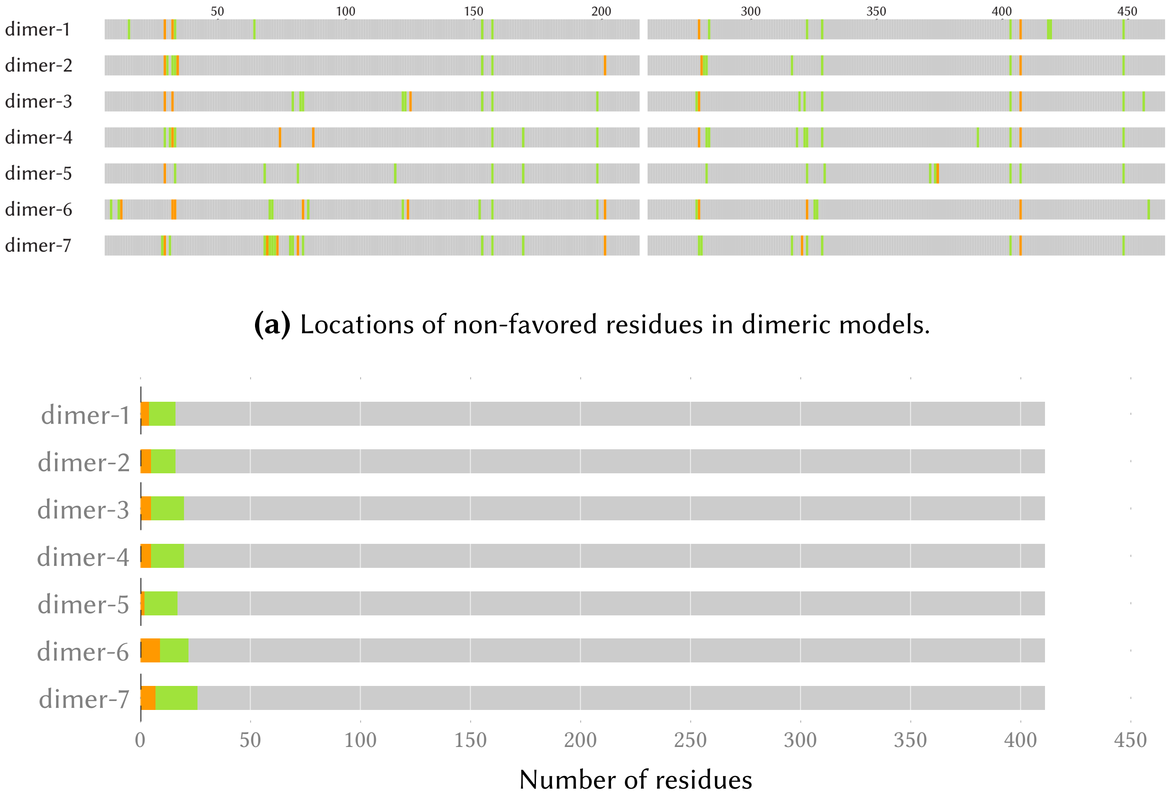


(a) Chain A active site **(b)** Chain B active site

Figure 4.6. Active sites of chains A and B of the manually created dimeric CrPCS model, superposed onto the templates 2BTW and 2BU3. The model is in dark gray, orange (active site residues), and green (B-loops); the templates are in white.

The Ramachandran plots of the manually generated dimers were of a similar quality (Figures 4.6 and 4.7). The favored model, "dimer-1", had four outlier residues and 11 residues in allowed regions. Since about 1 % of outlier residues are common in native proteins, this can be regarded as an indicator for a good model.

There are a number of regions where Ramachandran violations were common. As the model is a homodimer and the conformations of the templates' two chains are largely identical, the locations of Ramachandran violations are mirrored across the two chains of the target models. They can therefore be separated into four distinct groups: Residues 29 to 34/280 to 284, residues 75 to 88/317 to 329, residues 154 and 158/404 and 408, and residues 199 and 202/residue 449. Violations at residues 29 to 34/280 to 284 are probably due to insertions not present in the template, as is the case for the monomeric model. The reason for Ramchandran violations in the 75 to 88-/317 to 329-regions can be attributed to an alignment error in the protruding loops of the two chains. The Leu^{154}/Leu^{404} residues are part of a 3_{10} helix, whose Φ and Ψ angles delimitate the allowed regions in a Ramachandran plot. Thus, small changes in the Φ/Ψ-angles may easily induce Ramachandran violations. The close proximity of the Asp^{158}/Asp^{408} residues to these helices may be the reason for their deviations from ideal values.



(a) Locations of non-favored residues in dimeric models.

(b) Number of residues among favored, allowed, and outlier groups in dimeric models.

Figure 4.7. Ramachandran plot overview for dimeric models. Green residues are in allowed regions, orange ones in outlier regions, light gray ones in favored regions.

Models Created by Modeling Servers

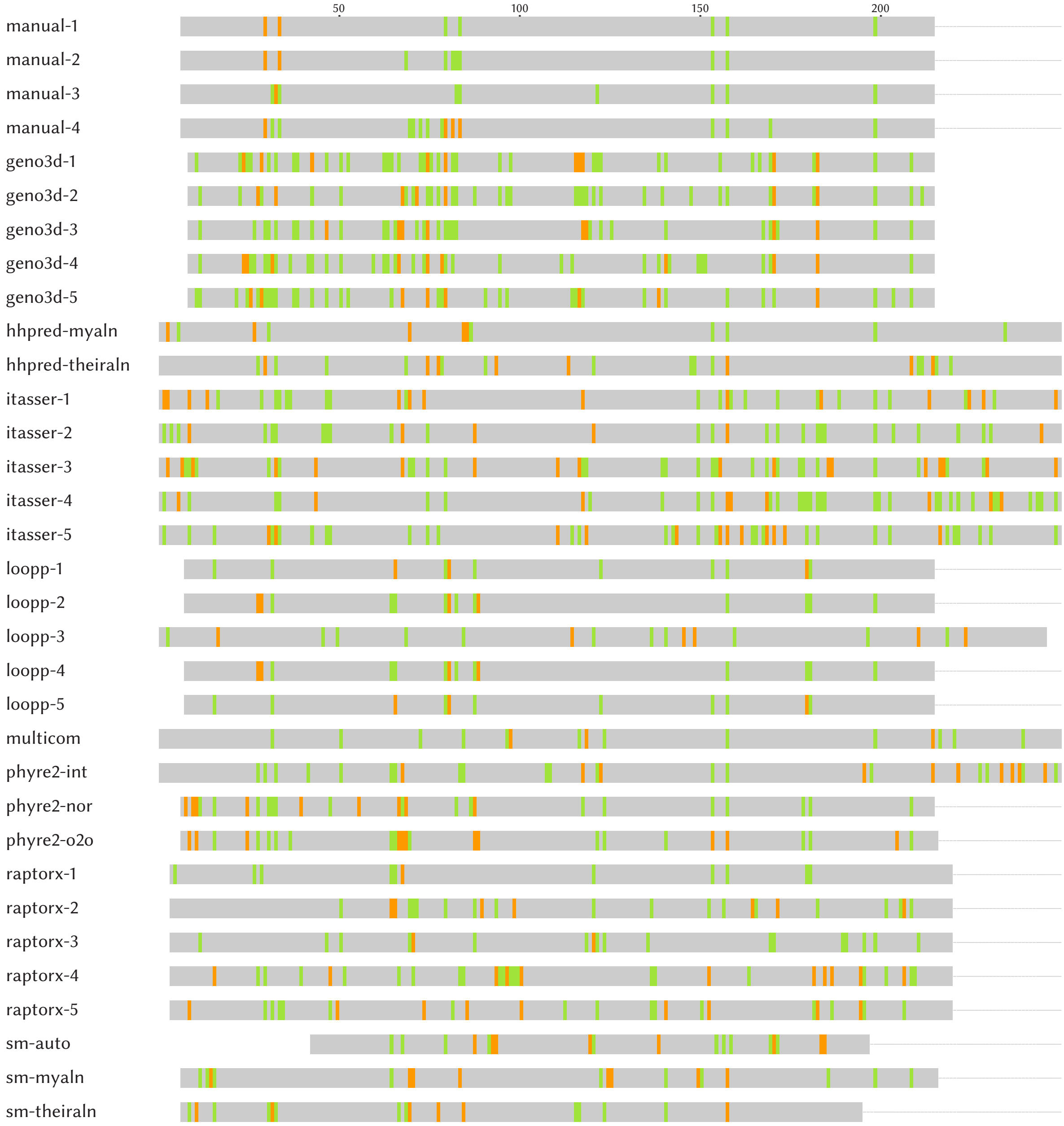
The models with the fewest outlier residues are the manually created models 1 to 3, the RaptorX models 1 and 3, the Loopp model 1, and the single Multicom model (Figure 4.8). The model with the highest relative number of residues in favored regions was manual-1 (96.6 %), closely followed by manual-2 and manual-3 (both 95.7 %), hhpred-myaln (95.2 %), all Loopp models (94.6 %), and RaptorX model 1 (94.9 %). The I-Tasser models, while generally of high quality according to the scores, did not score in the sam manner in the Ramachandran plots. The best I-Tasser models have 87.5 % and 86.3 % residues in favored regions, the rest of the models are in the 82 % to 83 % range.

Where possible, the use of the manually refined alignment (indicated by the "-myaln" suffix in Figure 4.8) generally led to fewer outliers and a higher number of residues in the favored regions, suggesting that the manual refinement of the alignment was successful. The Phyre^2 server with its "one to one" workflow allowed the selection of a single template together with an alignment. Here, the use of the manual alignment together with template 2BU3a did not lead to an improvement with regard to Phyre^2's "normal" mode. As Phyre^2 usually uses multiple templates for a single target model, this could result from the (comparably) low number of templates, which might increase the signal-to-noise ratio.

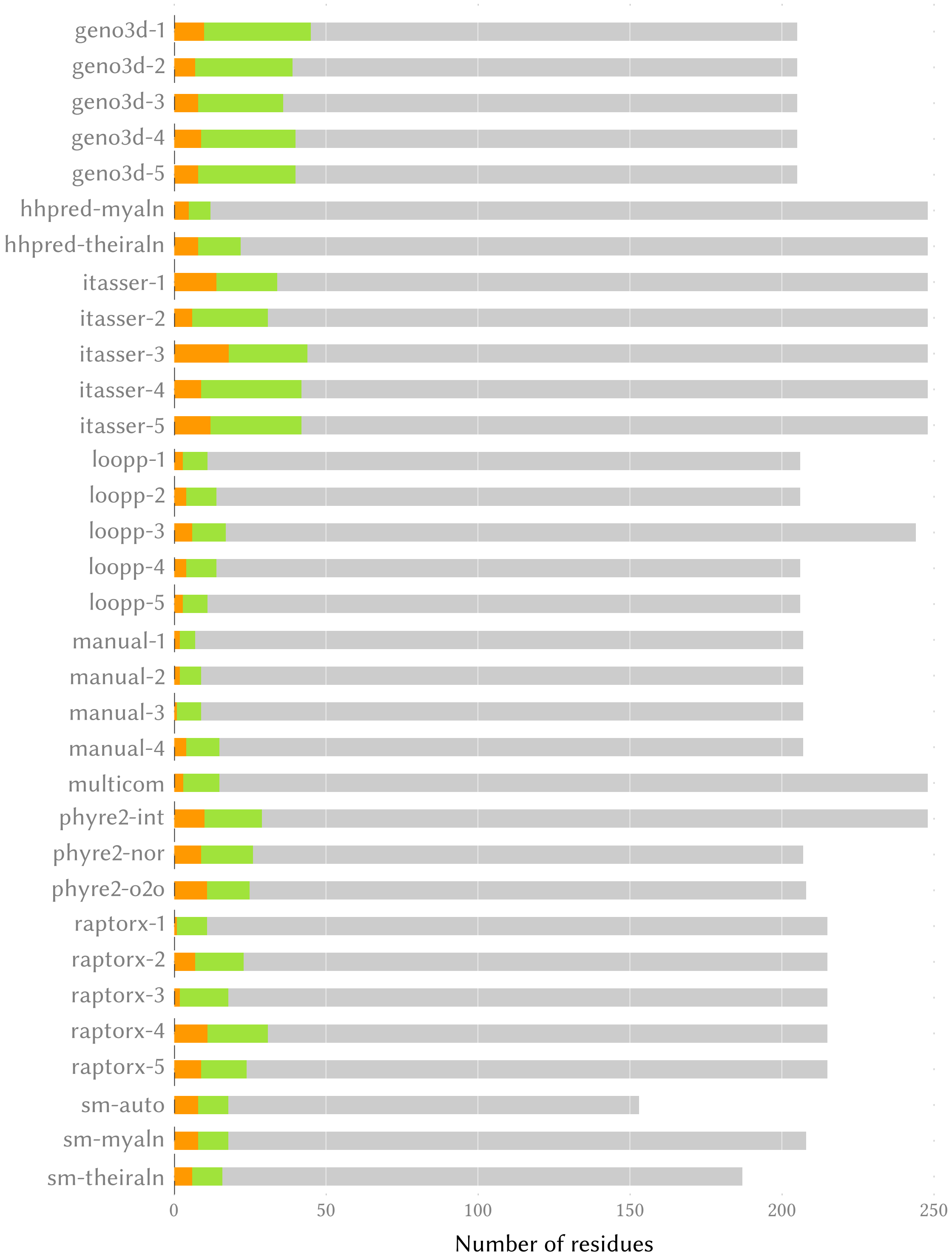
As mentioned in Section 4.1.1, almost all servers used 2BU3 and/or 2BTW as templates, in various combinations and alignments. The top ten models for each score can be found in Table 4.3, the actual scores for all models have been moved to the appendix (Appendix A.4). Additionally, a superposition of the ten best models is shown in Figure 4.9.

Surprisingly, few of the servers which scored high in the Critical Assessment of Methods of Protein Structure Prediction 9 (CASP9) [96] managed to place a model among the top ten. Perhaps more surprisingly, the servers that were only included in order to compare them to the CASP9 servers, Geno3D and Loopp, produced some of the highest-ranked models *in silico*. In almost all cases, the models built with the manual alignment were better, except for the one built with SwissModel server: Here, the alignment generated by the server ("SM-theiraln") scored much better than the manual one ("SM-myaln").

While the scores are all well-suited for "complete" monomeric models—i.e., models without missing residues and which are not part of a multi-protein complex—, they are not as well suited for assessing dimeric models. The scores do not correlate very well with each other (cf., the R^2 values in Table 4.4). Despite these drawbacks, the Dope score tended to agree best with the ranking of the models obtained by the manual modeling procedure (Appendix A.4). Therefore, the Dope score is considered to be a good choice for comparative modeling.



(a) Locations of non-favored residues in monomeric models



(b) Number of residues among favored, allowed, and outlier groups in the monomeric models.

Figure 4.8. Ramachandran plot overview for monomeric models. Green residues are in allowed regions, orange ones in outlier regions, light gray ones in favored regions.

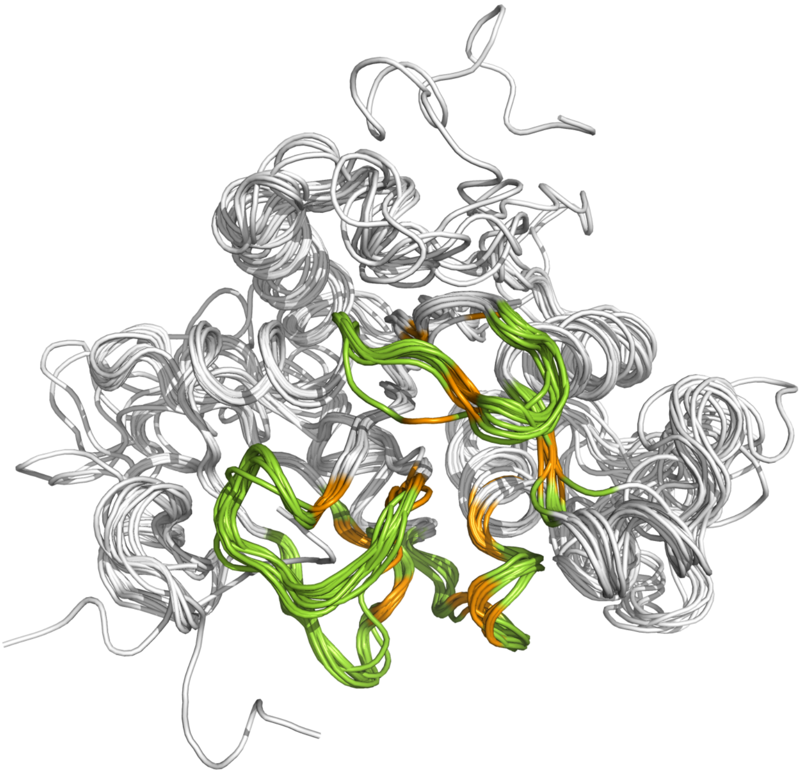


Figure 4.9. Maximum-likelihood superpositioning of the best manual- and server-generated models. Active site residues are orange, B-loops are colored green. The "best" models are those which are mentioned in Table 4.3. The superpositioning was achieved using an alignment between all manually- and server-generated models (see Appendix A.3), downweighting variable regions, and correcting for correlations among atoms. This way, regions of the models which are structurally similar are weighted higher than structurally variable regions. All heavy (i.e. non-hydrogen) atoms were superposed. The alignment was generated using Theseus [156].

Table 4.3. Top ten models for each score

Position	Dope	Selectpro	Qmean6 Z-score
1	Manual-1	Manual-1	Manual-3
2	Manual-2	Manual-2	Loopp-1
3	Manual-3	Manual-4 and Loopp-1	Loopp-2
4	Manual-4	Loopp-2	RaptorX-1
5	Loopp-2	Loopp-1	SM-theiraln
6	Loopp-1	RaptorX-1	Manual-4
7	RaptorX-1	Phyre^2-o2o	Phyre^2-o2o
8	Geno3D-4 and SM-theiraln	Manual-3	Manual-2
9	Geno3D-3	Geno3D-4	Manual-1
10	Geno3D-2 and Geno3D-5	I-Tasser-4	SM-myaln

Table 4.4. R^2 correlation between Dope, Selectpro, and Qmean6

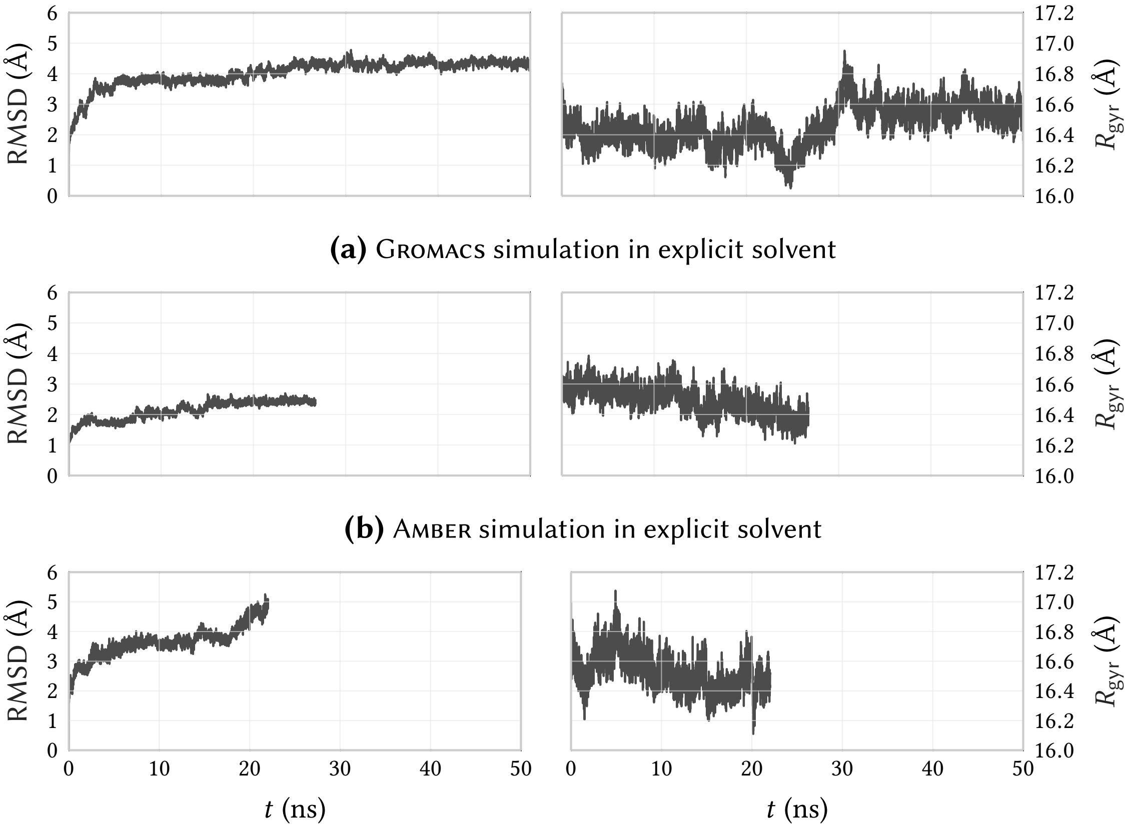
	Dope	Selectpro	Qmean6
Dope		0.693	0.658
Selectpro	0.693		0.411
Qmean6	0.658	0.411	

4.2. Molecular Dynamics

Both manually built models, the monomeric Modeller model "manual-1" and the dimeric Modeller model "dimer-1" in Table 4.2, were submitted to MD simulations. However, a visual inspection of the trajectories revealed a different behavior of the Amber and Gromacs simulations with respect to the elimination of rotational and translational degrees of freedom: Apparently, not all such movements were removed from the Gromacs simulations. Therefore, the radii of gyration (R_{gyr}s) were calculated as well: Both RMSD and R_{gyr} are considered as measures for the conformational stability of the models. The R_{gyr}s of all models are similar, at around 16.6 Å for the monomers and 22 Å for the dimers (Figures 4.10 and 4.14).

Generally, the simulations in explicit solvent seemed to be both more stable and computationally efficient for this system. While using explicit solvent results in more realistic conditions, one could argue that a model in an implicit solvent simulation is able to traverse conformational space faster, leading to a net improvement in computational efficiency. However, in order to see this effect, the simulations would have to be run for a longer time.

Interestingly, the explicit solvent simulations were faster than the implicit solvent ones for the chosen system. This is mainly due to the different scaling behavior of implicit and explicit solvent simulation algorithms: While the calculation per solvent molecule is fairly simple, massively parallelizable, and usually highly optimized in modern MD packages, it scales poorly with system size. For example, if a macromolecule has an R_{gyr} of a, and a new, larger system has an R_{gyr} of $2a$, the volume of our larger system's simulation box has to increase by a factor of $2^3 = 8$ so as to not violate the minimum image convention (see Section 2.2.6). The calculation of implicit solvent is not as straightforward and not as easily parallelizable. However, it doesn't require calculating interactions with countless numbers of solvent molecules and therefore scales much better with system size. If the chosen model is small, it allows for a small box size and a relatively small number of solvent molecules.



(a) Gromacs simulation in explicit solvent

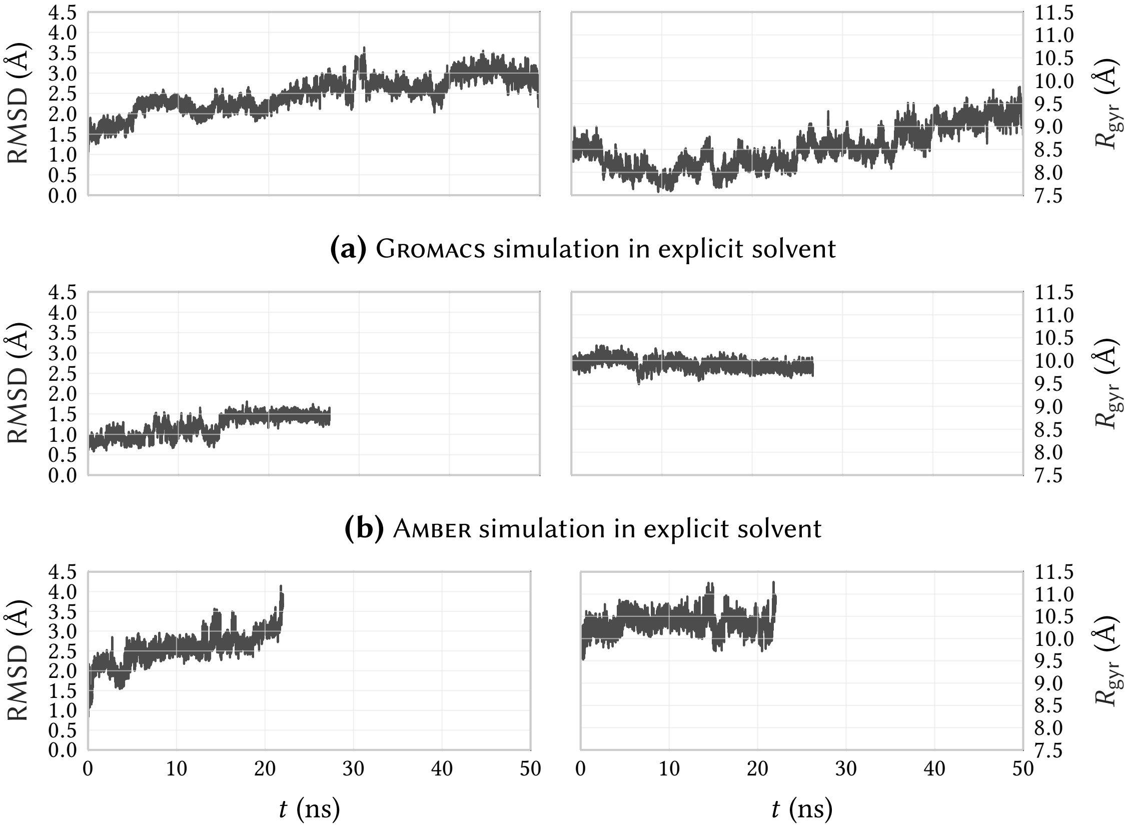
(b) Amber simulation in explicit solvent

(c) Amber simulation in implicit solvent

Figure 4.10. Heavy-atom RMSD and R_{gyr} plots for MD simulations of monomeric models.

4.2.1. Monomeric Model MD Simulations

Due to computational and time constraints, not all simulations could be carried out until stability. From the RMSD plots of the simulations of the monomeric models (Figure 4.10), it is obvious that both the Amber and Gromacs simulations reached a stable state of the protein (on the MD timescale). The Gromacs simulation in explicit solvent is considered to be the second most stable. In contrast, no plateau could be achieved in the Amber simulation in implicit solvent.



(a) Gromacs simulation in explicit solvent

(b) Amber simulation in explicit solvent

(c) Amber simulation in implicit solvent

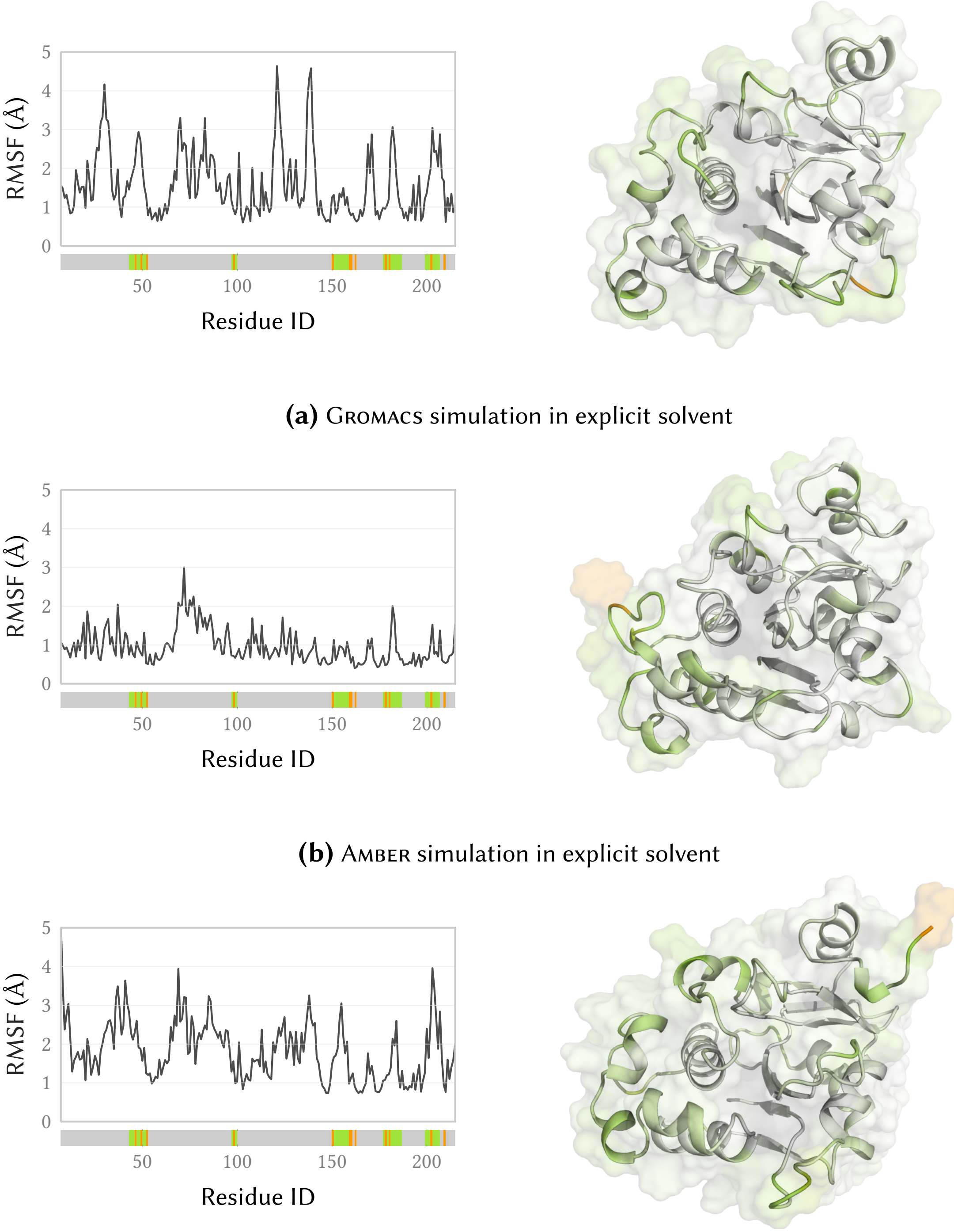
Figure 4.11. Active site-only RMSD and R_{gyr} plots for MD simulations of monomeric models.

The active site residues (Figure 4.11) were stable in the explicit solvent simulations. This is supported by the rather large root mean square fluctuation (RMSF) value of several active site residues in the Gromacs simulation (Figure 4.12a). The RMSD value was large compared to the Amber simulation in explicit solvent. However, as mentioned before, this might be a consequence of the problems encountered when fitting a Gromacs trajectory using the supplied Gromacs tools. In comparison, the Amber simulation in explicit solvent is remarkably stable, with relatively small RMSD, R_{gyr}, and RMSF values. The Amber simulation in implicit solvent has not yet reached a stable plateau at the very end of the simulation. It seems to be in the middle of undergoing a conformational change, as indicated by a shift in both RMSD and R_{gyr}. The catalytically important residue His^{160}, which is part of the catalytic triad, flips its side chain around several times at the start of the simulation. The B-loop 5 is not stable either, detaching and re-attaching liberally throughout the course of the simulation.

The main fluctuation in the Amber simulation in explicit solvent is in the protruding loop; other regions can be considered stable (Figure 4.12b). It is assumed that the relatively high RMSF in the protruding loop region is a consequence of the alignment error mentioned earlier. Additionally, the protruding loop has five residues with long, bulky side chains (Arg^{69}, Arg^{70}, Trp^{72}, Lys^{73}, Trp^{76}, Arg^{77}, Trp^{78}), most of which are solvent-exposed and hence very flexible. In general, however, the Amber simulation in explicit solvent can be considered stable from 24 ns onwards.

Judging from its RMSF, the Gromacs simulation was not as stable (Figure 4.12a). It did, however, result in an RMSD plateau: The slight increase in RMSD at $t \approx 24$ ns was the result of B-loop 5 (residues 199 to 206) “detaching” from the rest of the protein; it stayed in this conformation until the end of the simulation. Several loops contributed to the high RMSF of the simulation, among them the region around residues 30 to 34, where three residues were inserted compared to the template. Other regions with high RMSF values were the B-loop 1, 124 to 126, 137 to 139, and B-loop 4. The first B-loop’s RMSF is relatively large due to the difference in secondary structure between the chosen templates and the target: The template structures have 3_{10}-helices at the start of the loop, CrPCS has a proline residue instead. Since a 3_{10} helix is less stable than an α-helix and proline tends to disrupt helical structures, the modeled structure is probably not very accurate in this region. The region around residues Leu^{124} to Leu^{126} is close to an insert in the original alignment (see Figure 4.1), and the two leucine residues are solvent-exposed. Additionally, the same situation as in B-loop 1 is present: A proline residue in CrPCS is aligned to a region in a 3_{10}-helix. B-loop 4 has several residues with large, solvent-exposed side chains. A similar situation was found in the B-loop 5 region: solvent-exposed arginine and lysine residues contribute to a high RMSF. A visual inspection of the simulation indicated a stable molecule, except for the aforementioned conformational change in B-loop 5.

The Amber simulation in implicit solvent did not achieve a stable conformation within the given timeframe, judging from the inability to form an RMSD plateau. Its RMSF plot also had a high baseline of around 1 Å to 1.5 Å, compared to the 0.5 Å to 1 Å baseline of the simulations in explicit solvent. Several regions changed their conformations over the



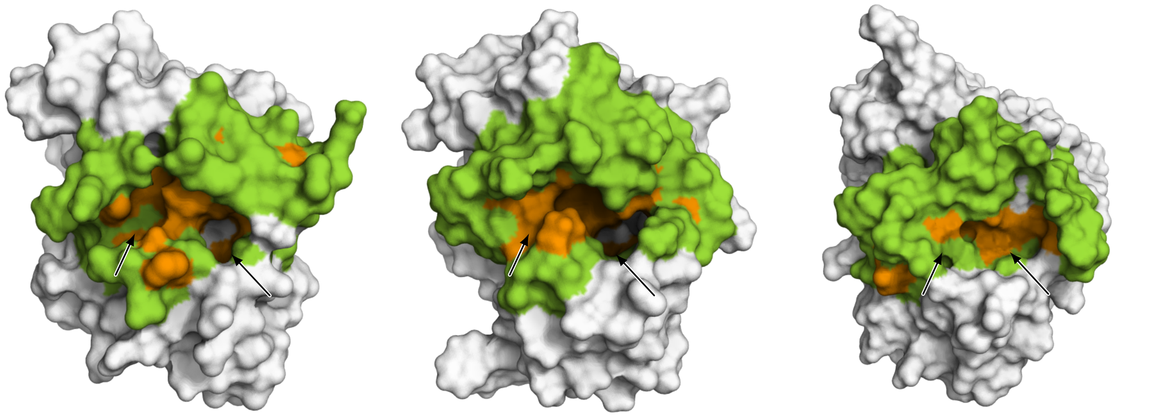
(a) Gromacs simulation in explicit solvent

(b) Amber simulation in explicit solvent

(c) Amber simulation in implicit solvent

Figure 4.12. RMSF plots (left) and pseudo-b-factors (right) of monomeric models. For the sake of orientation, the locations of active site residues (orange) and B-loops (green) are shown below each RMSF plot. The b-factors are indicated in the reference structures used for RMSF calculation. White regions have a small RMSF, green regions have a medium one, and orange regions have the largest values. The positioning is the same as for the previous figures in this chapter, i.e. the active site is in front and center.

course of the trajectory, the most prominent of which are the N-terminus, the 30 to 34 insert region, the B-loop 1, the protruding loop and the helix following it, the 3_{10} helix around residues 116 to 120, B-loop 3, and B-loop 5. In B-loop 1, a flip in the main chain of Phe^{43} results in the entire loop "shifting" at around $t = 8$ ns, and again at $t = 19$ ns. In B-loop 3, the loop-internal hydrogen bond network is switched, resulting in a different configuration of hydrogen bonds and a subsequent change in conformation. B-loop 5 is held in place almost entirely by electrostatic interactions between the side chains of Tyr^{200} (B-loop 5) and Lys^{150} (B-loop 3), and Lys^{203} (B-loop 5) and Asp^{158} (B-loop 3). These interactions cease at around $t = 5$ ns, resulting in the loop "detaching" from the rest of the protein. A visual inspection yielded the same results: The model can not yet be considered stable.



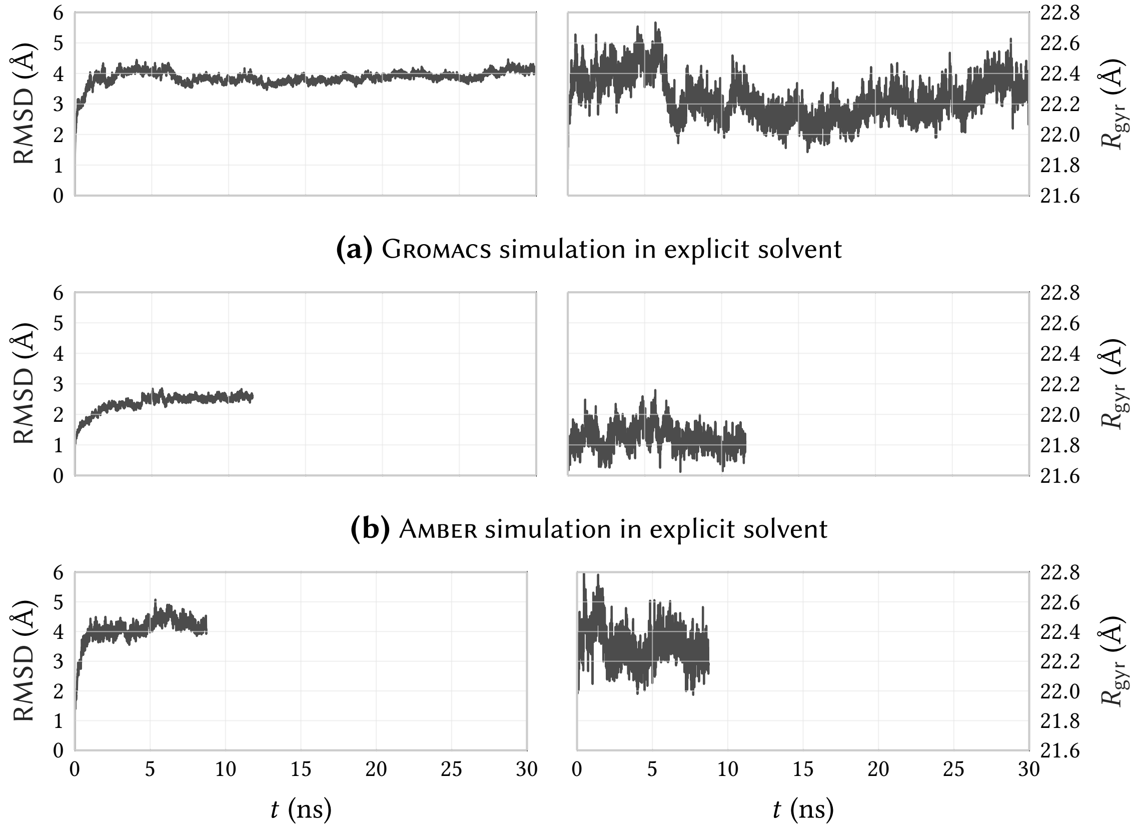
(a) Gromacs simulation in explicit solvent **(b)** Amber simulation in explicit solvent **(c)** Amber simulation in implicit solvent

Figure 4.13. Locations of putative secondary substrate binding sites in monomeric reference structures. Active site residues are orange, B-loop residues are green. For each structure, two cavities can be seen: The right, deeper one is the primary substrate binding site, the left, more shallow one is the putative second substrate binding site.

The cavity of a potential second substrate binding site close to the active site is readily visible in the reference structures of both explicit solvent simulations (Figure 4.13). In the reference structure of the Amber simulation in implicit solvent, the cavity adopts a markedly different structure. This second substrate binding site is much more pronounced than the one in NsPCS [170]: While the bound ligand in NsPCS mostly reacts with water, forming γ-EC or a derivative compound, the cavity in CrPCS is larger and can therefore accomodate larger substrates, leading to the formation of PC_n. The sites were assessed visually and were present over the entire trajectories.

In general, care has been taken to submit the models to the same environmental conditions despite the choice of two different force fields (`ff03` and `GROMOS96 53A6`). However, where suggestions from the literature indicated differing values for the force fields in order to obtain realistic results, these values were used instead. Despite these differences, the models can be considered stable under all explicit solvent conditions. Therefore, the missing C-terminus has only a minor influence on the general protein stability. However, the shape of the substrate binding sites fluctuates over the course of the trajectories, as is indicated by the differently shaped primary and secondary substrate binding sites in Figure 4.13. Because of that, docking attempts with GSH and CysGSH were inconclusive (data not shown).

4.2.2. Dimeric Model MD Simulations



(a) Gromacs simulation in explicit solvent

(b) Amber simulation in explicit solvent

(c) Amber simulation in implicit solvent

Figure 4.14. Heavy-atom RMSD and R_{gyr} plots for MD simulations of dimeric models.

For the dimeric models, the simulations in explicit solvent can be considered stable (Figure 4.14). The RMSD of the Amber simulation in explicit solvent varies around 2.6 Å from 6 ns onwards, the RMSD of the Gromacs simulation in explicit solvent is stable at about 4 Å. The Amber simulation in implicit solvent did not achieve stability, oscillating around an

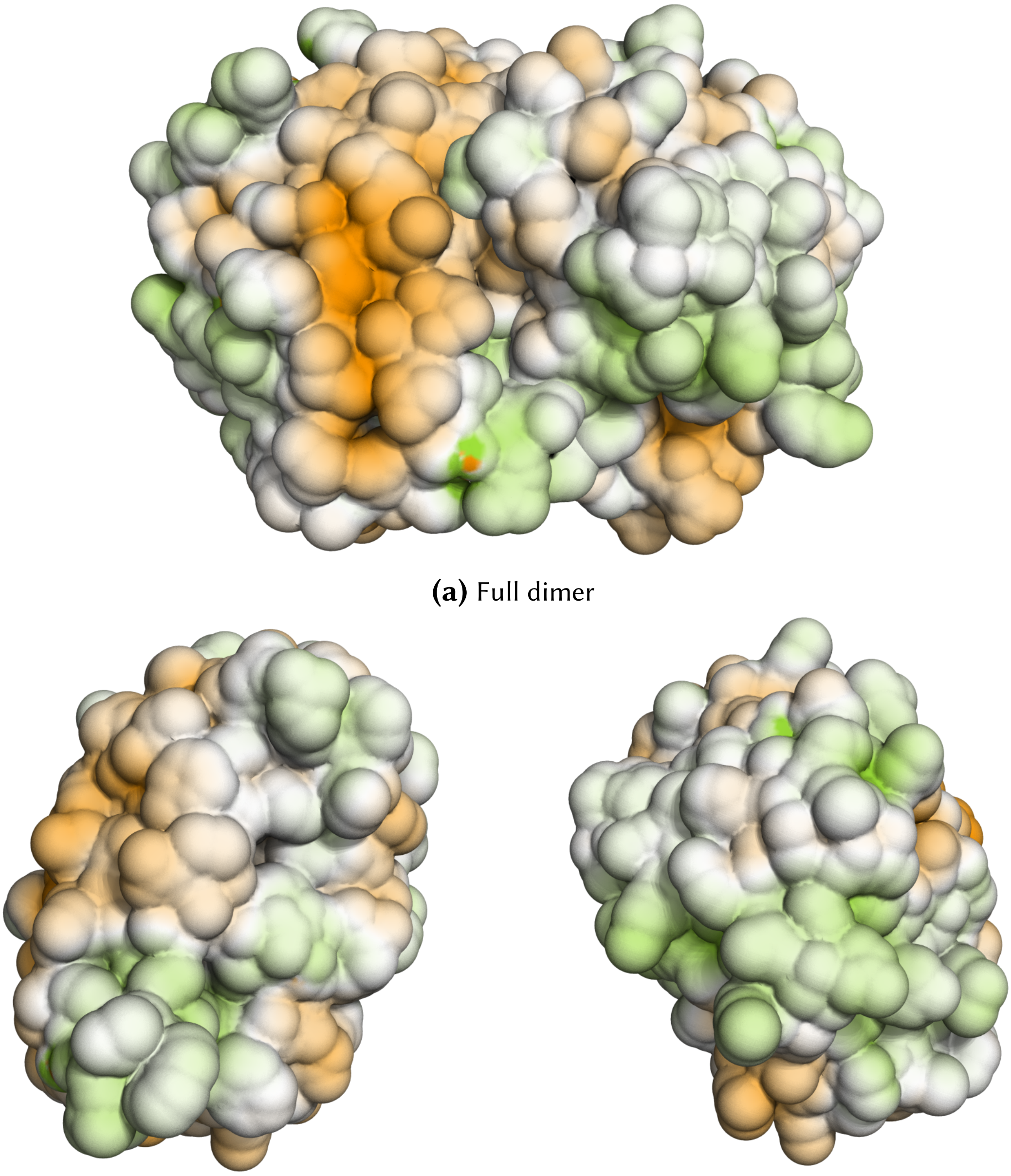
RMSD of about 3.5 Å to 4.8 Å, with relatively large fluctuations. The Gromacs simulation in explicit solvent was stable according to its RMSD. From RMSF calculations, the stable period was estimated to start at 8 ns onwards.

The dimeric interface consists of the first two N-terminal α-helices and their adjacent residues, the B-loop 4, and part of the protruding loop region. Consequently, the B-loop 4 is much more stable in the dimeric models than it is in the monomeric ones. The RMSDs of the protruding loop regions are higher for all monomers than they are for the dimers, indicating a larger amount of fluctuation and a less stable structure in these regions. A visual inspection also suggested that the regions of the inserts (residues 30 to 34 and 280 to 284) were more stable in the dimers than they were in the monomers.

As evident from Figure 4.16, the active site in the Gromacs simulation in explicit solvent "opens up" in both chains.

This is due to two residues which are also part of the B-loop 5, Asp^{202} and Arg^{209} (chain 1)/Asp^{452} and Arg^{459} (chain 2), as this loop loses contact to the rest of the protein during the simulation, resulting in increases in both RMSD and R_{gyr}. The active site of the Amber simulation in explicit solvent is completely stable. The B-loop 5, which loses contact in both chains in the Gromacs simulation, is completely stable here in chain 1. In chain 2, it loses contact, resulting in a noticable RMSD increase, but "reattaches" soon after and keeps its conformation over the course of the entire trajectory. In the Amber simulation in implicit solvent, the most unstable region was the α-helix N-terminally adjacent to the B-loop 2. This instability was visible in both chains.

In general, the dimeric models were at least as stable in their MD simulations as the monomeric ones, if not more so. Therefore, as in the monomeric models, the missing C-terminus does not impact the general protein stability. However, the RMSF plots show that the dimeric chains did not generally behave the similarly to each other in all models. This indicates that, if a dimeric form of the protein exists *in vivo*, its two chains could have different functions as postulated by Vivares, Arnoux, and Pignol for the dimeric NsPCS [170]. Again, the explicit solvent simulations are faster than the implicit ones, even for the increased system size relative to the simulations of the monomers. Doing MD simulations of the dimeric CrPCS system in Gromacs seems to strike a good balance between stability and computational efficiency.

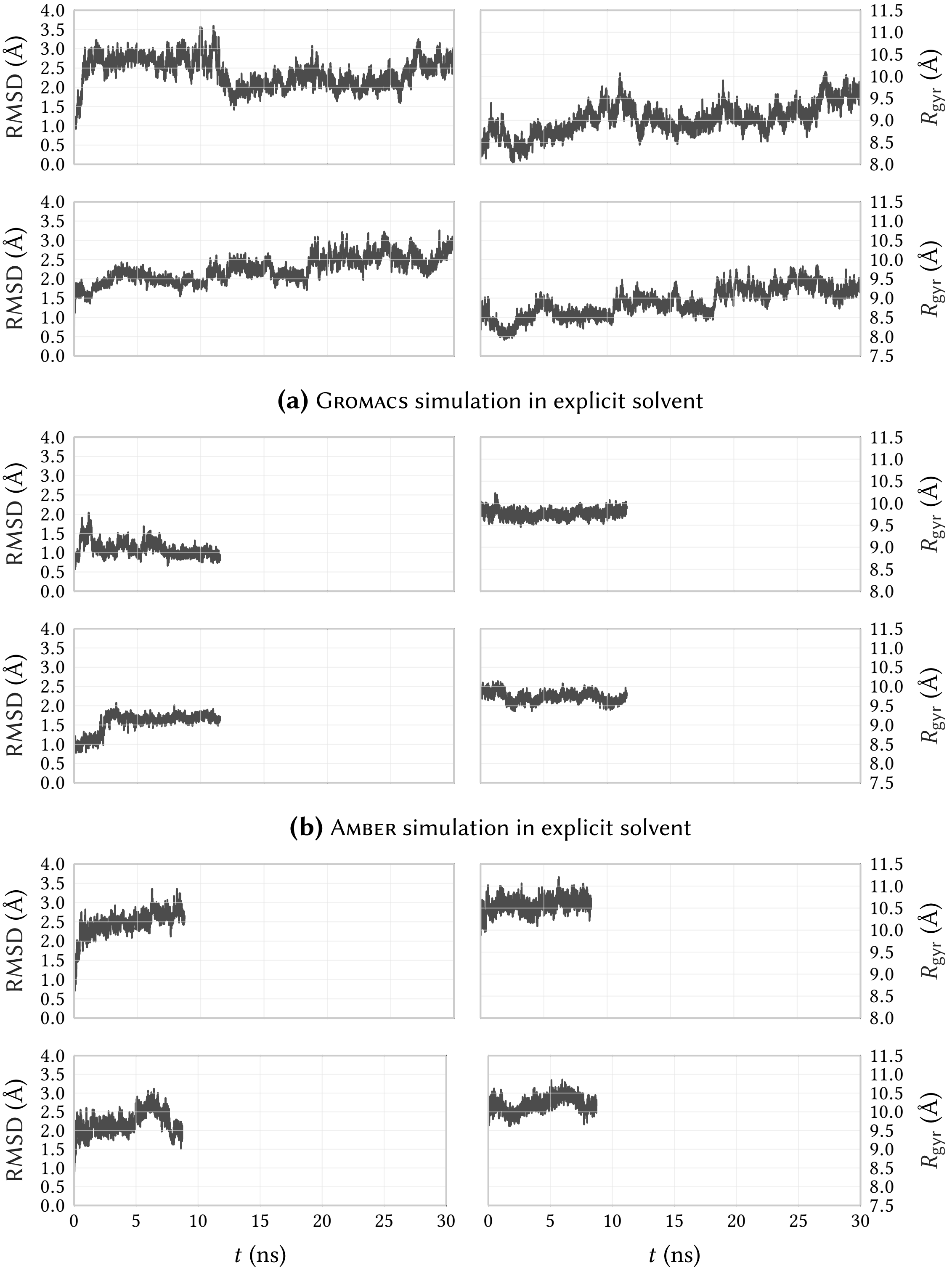


(a) Full dimer

(b) Chain 2

(c) Chain 1

Figure 4.15. Electrostatic surface potential at the dimeric interface, calculated using Apbs. Orange values are negative, green values are positive. In (a), both dimeric subunits are displayed. In (b) and (c), the subunits are separated and rotated with the dimeric interface part exposed. It can be clearly seen that the electrostatic potential is weaker at the dimeric interface than it is on the rest of the protein, which indicates that these residues might in fact be buried.

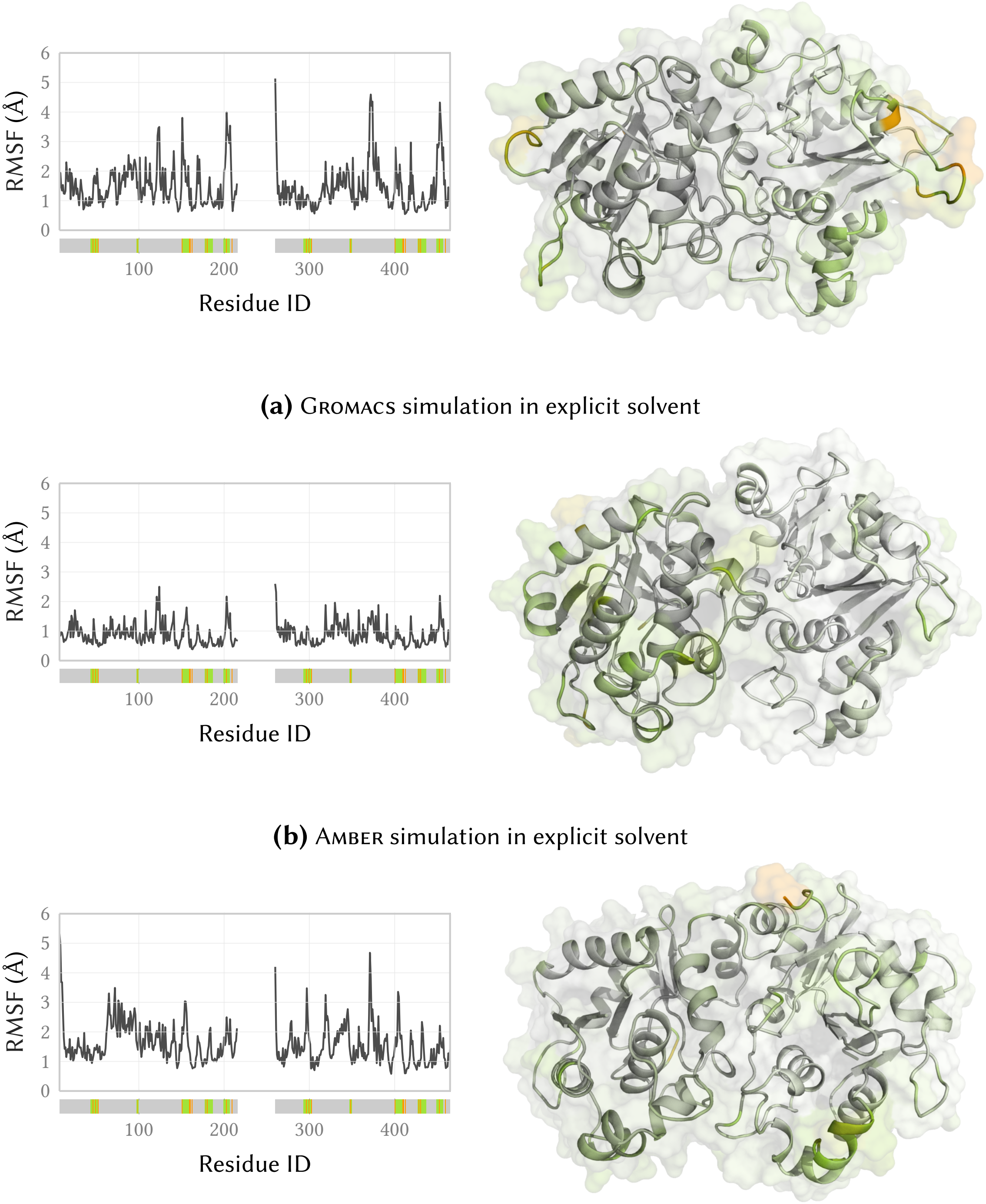


(a) Gromacs simulation in explicit solvent

(b) Amber simulation in explicit solvent

(c) Amber simulation in implicit solvent

Figure 4.16. Active site RMSD and R_{gyr} plots for MD simulations of dimeric models. The first chain is on top, the second chain on the bottom.



(a) Gromacs simulation in explicit solvent

(b) Amber simulation in explicit solvent

(c) Amber simulation in implicit solvent

Figure 4.17. RMSF plots (left) and pseudo-b-factors (right) of dimeric models. For the sake of orientation, the locations of active site residues (orange) and B-loops (green) are pictured below each RMSF plot (left). The b-factors (right) are overlayed over the reference structures used for RMSF calculation. White (chain 1) or gray (chain 2) regions have a small RMSF, green regions have a medium one, and orange regions have the largest values. Chain 1 is always the right subunit, chain 2 the left one. The viewport is again chosen so that the active site of chain 1 is front and center.

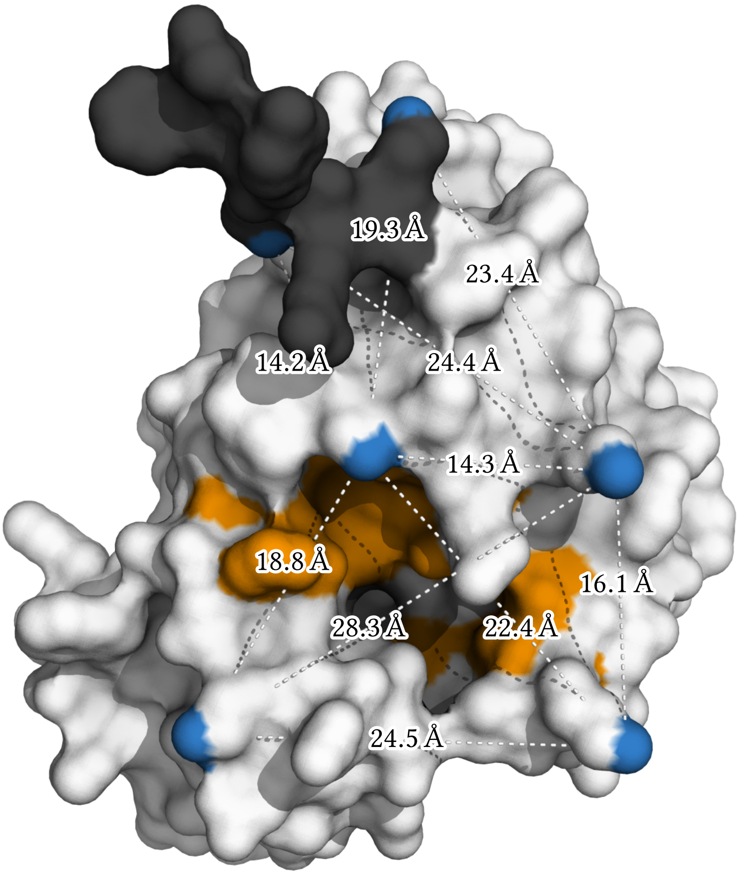
4.3. Suitability for Crosslinking

Crosslinking will be done by linking the ϵ-amino groups of two solvent-exposed lysine residues with a linker of predetermined length. The two N-terminal truncated lysines have been re-added here in order to be able to explore all crosslinking possibilities. No lysines were present in the truncated C-terminal domain, therefore it is not included in the figures of this Section.

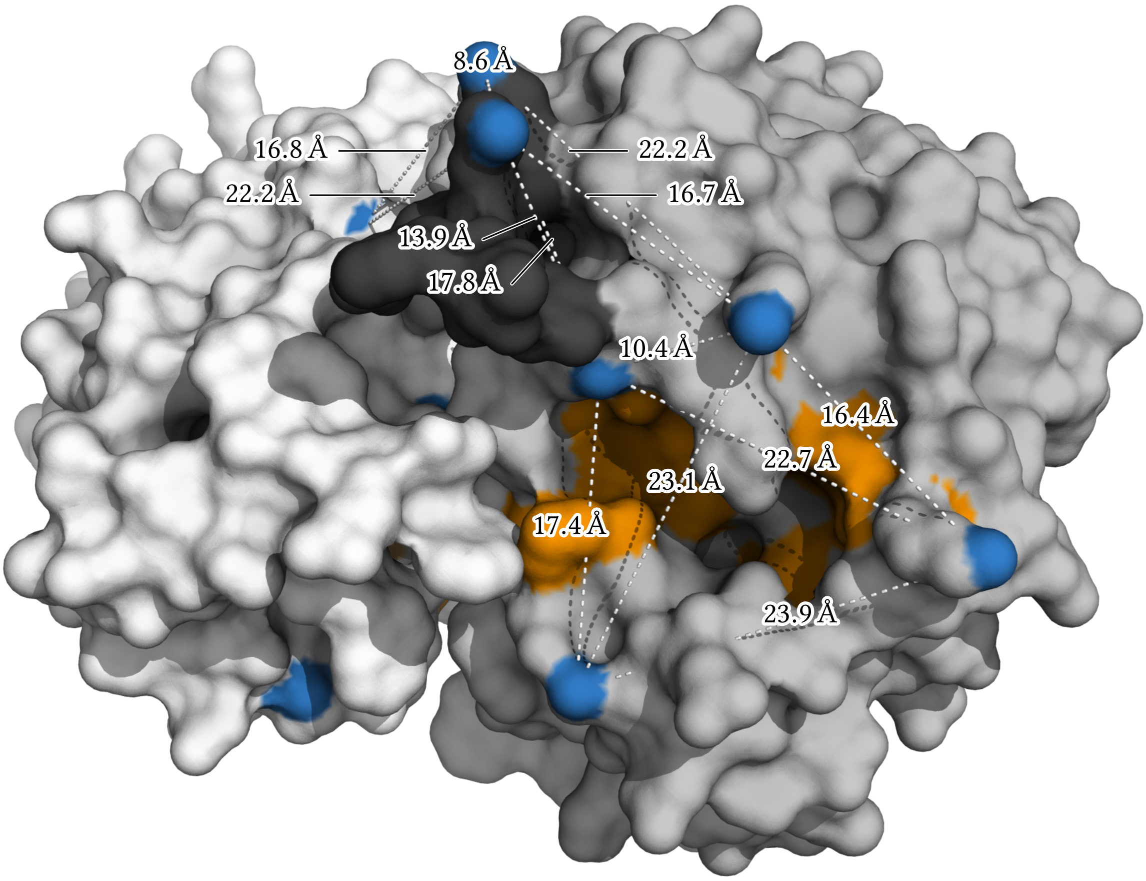
Depending on the length and flexibility of the spacer, there are a number of residues which can be used for this (Figure 4.18). Four of those residues are arranged around the active site in a distinctive pattern; two are on the "back" from the viewport in Figure 4.18a and on the left of chain 1 in Figure 4.18b; and two lysines were truncated from the N-terminus during the modeling process in order to improve modeling quality (see Section 3.3 and Figure 4.1).

Crosslinking can potentially also be used to test whether CrPCS occurs as a dimer: At least one solvent-exposed lysine residue in the monomeric model is buried by the dimeric interface in the dimeric model. If a dimeric CrPCS would be subjected to crosslinking and subsequent mass spectrometry analysis, fragments involving this buried residue would not occur in the analysis because the spacer could not link to this buried lysine residue.

The average values and standard deviations of the lysine distances over all trajectories are shown in Appendix A.5.1.



(a) Monomeric model



(b) Dimeric model

Figure 4.18. ϵ-amino groups of surface-facing lysine residues potentially suitable for crosslinking. The ϵ-amino nitrogens are marked in blue, the active site is marked orange for orientation. Potential crosslinkings between lysines are marked by dotted lines. The crosslinkings only measure direct distances between nitrogens; therefore, some lines pass through the surface of the protein. No direct distances larger than 30 Å have been taken into consideration. As both the lysine sidechains and the linker are usually flexible, this is not necessarily prohibitive as long as the linker is long enough. In figure (b), chain 1 is white while chain 2 is light gray. A similar, mirrored configuration of lysine residues is on the "back" of the dimer.

4.4. Cd^{2+} Binding Sites

Cd^{2+} binding sites were investigated in reference to Maier et al. [88], who tested PCSs from several different organisms for contiguous cysteine-rich metal-binding motifs as these occur frequently within MTs and CPX-type ATPases [33]. Among the investigated PCSs, seven Cd^{2+} binding motifs were identified, four of which were located in the N-terminus. According to Maier et al. and Ha et al., only these N-terminal Cd^{2+} binding sites are necessary for protein function [56, 88]. While the sequences of the binding motifs vary considerably, their locations relative to each other are fairly constant.

In order to find out whether Cd^{2+} binding motifs also exist in CrPCS, the alignment generated for modeling was extended with the sequences for TaPCS and SpPCS, the subjects of the experiments undertaken by Maier et al. The regions aligned with Cd^{2+} binding sites in TaPCS or SpPCS were inspected for residues which conferred Cd^{2+} binding ability in TaPCS and SpPCS: cysteines, aspartates, and glutamates.

The locations of the binding motifs in the PCSs examined by Maier et al. in comparison to CrPCS are shown in Figure 4.19. Most of the residues are well-conserved in the eukaryotic CrPCS but not in the prokaryotic NsPCS. The cysteine residues that are responsible for the binding affinity towards Cd^{2+} are almost completely conserved between the eukaryotic PCSs. NsPCS has hardly any conserved binding motif in its N-terminus. It is hypothesized that PCS used to have additional function beside Cd^{2+} detoxification and/or homeostasis (see Section 1.3.2): Zn^{2+} and Pb^{2+} can activate PCS as well [31], and due to this relatively low specificity, Clemens and Peršoh propose that PCS might be responsible for Zn^{2+} detoxification as well.

Incidentally, the location of the second and third binding site forms a "cysteine patch" in the NsPCS template as well. This patch is also visible in the CrPCS model (Figure 4.20). It is also known that the template 2BTW had a Ca^{2+} atom at a putative Cd^{2+} binding site (colored gray in Figure 4.20). Cd^{2+} can block Ca^{2+} channels [153] and it can displace many metal cofactors as well, including Ca^{2+}, from their binding sites [151]. This indicates that some amount of chemical equivalence exists between Cd^{2+} and Ca^{2+}, further pointing at a nonspecific Cd^{2+} or metal binding site at that location. However, the cysteine at the fourth Cd^{2+} binding site is not visible on the surface (Figure 4.20).

The studies by Maier et al. were performed *in vitro* at a Cd^{2+} concentration of 10 µM which is far from physiological conditions (see Section 1.3.3). Cd^{2+} concentrations this high would not occur in the presence of GSH and/or PCs. Their high Cd^{2+} binding affinities would effectively lower the concentration of active free Cd^{2+}. Additionally, Vatamaniuk et al. and Tsuji et

```
           50   55   60        80   85   90        100  105  110
CrPCS   DEPAFCGLASLAMTLN | WRWFHEAMLDCCRPLDAV | GITLYQASCLARCNGA | ER
NsPCS   VNQAYCGVASIIMVLN | DNFFSNEKTKAVIAPEVV | GMTLDELGRLIASYGV | ED
TaPCS   SEPAFCGLASLSVVLN | WRWFDESMLDCCEPLHKV | GITFGKVVCLAHCAGA | HD
SpPCS   NEPAFCGLGTLCMILN | WRWYDQYMLDCCRSLSDI | GVTLEEFSCLANCNGL | DE

         130  135  140  145
CrPCS   FRREVEAVCGSGEEHIVV
NsPCS   FRKQVAENLKQDGNFVIV
TaPCS   FRAHLTRCASSQDCHLIS
SpPCS   FRKDVISCSTIENKIMAI
```

Figure 4.19. Cd^{2+} binding sites as proposed by Maier et al. [88]. The last three Cd^{2+} binding motifs found in the sequences of SpPCS and TaPCS have been omitted as they were not homologous to the CrPCS sequence. Cysteines are highlighted in yellow. The dark gray shaded regions are the binding motifs as proposed in the paper. The light gray shaded residues in the NsPCS sequence are located within 5 Å of the Ca^{2+} atom in 2BTW.

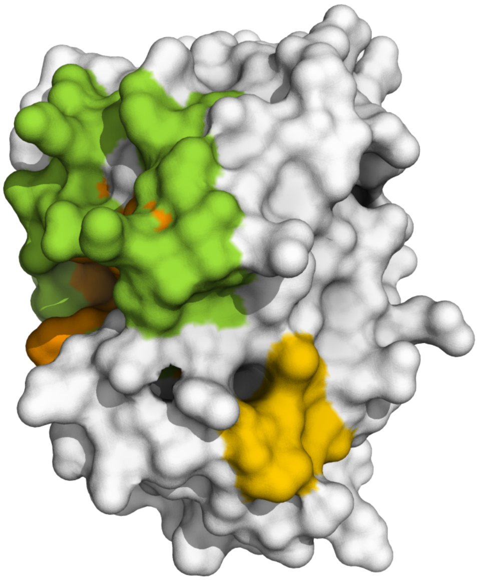


Figure 4.20. Location of the cysteines of the second and third binding sites in Figure 4.19. The primary substrate binding site is visible to the left, active site residues are colored orange, B-loops are green. Cysteines are marked in yellow. The first potential Cd^{2+} binding site shown in Figure 4.19 is the active site. The cysteine residues of the second and third Cd^{2+} binding site form a “patch” of cysteines. The fourth Cd^{2+} binding site is not visible on the surface of the model.

al. showed that activation of AtPCS occurs by means of thiol-blocked GSH/PCS molecules instead of direct binding of Cd^{2+} by the enzyme [160, 165]. In light of these findings, the validity of the Maier et al. approach can be called into question.

5. Conclusion

A number of stable 3D structure models of phytochelatin synthase of *Chlamydomonas reinhardtii* was built, together with protocols to simulate them in recent Amber and Gromacs force fields. While not complete, the models show that the missing C-terminus has only a minor influence on the protein stability in general. The presence of a cysteine-rich Cd^{2+} binding site was shown; however, no accurate prediction can be made as to whether it has a regulatory function. The presence of a putative second acylation site adjacent to the primary active site was shown as well. This second acylation site could hold a secondary substrate for the transpeptidase part of the PCS reaction.

The models have a number of lysine residues suitable for crosslinking; four of them are close to the active site and can be used to verify the validity of the built models by mass spectrometry. Since two lysines exist in the unmodeled N-terminus of the model, crosslinkings between these two N-terminal lysines and those around the active site could allow the addition of distance restraints to the modeling process, thereby allowing to model the N-terminus as well.

While experimental data (D. Dobritzsch, personal communication) indicate that CrPCS is monomeric *in vivo*, the demonstrated stability of the dimeric complex *in silico* suggests that dimeric PCS molecules might in fact exist. Calculation of electrostatic potentials show that the dimeric interface is fairly-defined and consists mainly of noncharged residues. Since these residues would be solvent-exposed in a monomeric model, the dimeric model is more stable due to burying those residues in its core.

6. Outlook

The modeling process can be accelerated by using a cricital distance cutoff instead of clustering metrics. This approach would be both simpler and lead to the creation of clusters with fewer false positives. Depending on the size of the protein to be modeled, a cutoff of 1 Å to 2 Å should be suitable. A further way to potentially improve the monomeric model is to include both chains of both `2BTW` and `2BU3` as template structures. This could potentially relieve the stability problems with the protruding loop. Crosslinking between lysines, in-gel digestion, and subsequent mass spectrometric analysis of the fragments could result in distance restraints between solvent-exposed lysine residues, which could improve the model as well. This could be especially helpful for modeling the six missing N-terminal residues, as they include two lysine residues as well. The most important improvement upon the model, however, would be to correctly model the missing C-terminus. Unfortunately, no immediately apparent templates exist and *ab initio* modeling is futile for twelve residues or more. Including restraints obtained from small-angle X-ray scattering could improve the predictions. While this would not allow discerning the exact atomic positions, it would reveal the general shape of the protein and show whether the C-terminal end is disordered or has a distinct conformation.

All MD simulations could be run for a longer time, which would show whether the created models are truly stable. A longer stable period would allow for a better choice of a reference structure for docking experiments, and also allow for more accurate ligand–protein analyses, such as molecular mechanics/generalized Born surface area (MM/GBSA).

It is assumed that substrate binding at the second site takes place only in combination with either substrate binding at the first site, or creation of the acyl-enzyme intermediate. In order to elucidate the substrate binding at the secondary binding site, additional models of the acyl-enzyme intermediates at the first acylation site would have to be constructed. Further MD simulations of the resulting acyl-enzyme could reveal whether conformational changes are induced due to substrate binding at the first site.

As several potential Cd^{2+} binding sites were found, MD simulations of this model with bound Cd^{2+} could also lead to further insight into activation of this enzyme.

Bibliography

Printed Sources

[1] B. Alder and T. Wainwright. “Phase transition for a hard sphere system.” *J Chem Phys* 27 (1957), pages 1208–1209.

[2] M. P. Allen and D. J. Tildesley. *Computer simulation of liquids.* New York, NY, USA, 1989.

[3] S. F. Altschul et al. “Basic local alignment search tool.” *J Mol Biol* 215 (1990), pages 403–410. DOI: 10.1016/S0022-2836(05)80360-2.

[4] S. F. Altschul et al. “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.” *Nucleic Acids Res* 25 (1997), pages 3389–3402.

[5] R. Anandakrishnan, B. Aguilar, and A. V. Onufriev. “H++ 3.0: automating pK prediction and the preparation of biomolecular structures for atomistic molecular modeling and simulations.” *Nucleic Acids Res* 40 (2012), W537–W541. DOI: 10.1093/nar/gks375.

[6] K.-J. Appenroth. “Definition of ”Heavy Metals” and Their Role in Biological Ssystems.” In: *Soil Heavy Metals.* Volume 19. Soil Biology. 2010, pages 19–29. DOI: 10.1007/978-3-642-02436-8_2.

[7] A. J. M. Baker, R. D. Reeves, and A. S. M. Hajar. “Heavy metal accumulation and tolerance in British populations of the metallophyte Thlaspi caerulescens J. & C. Presl (Brassicaceae).” *New Phytol* 127 (1994), pages 61–68. DOI: 10.1111/j.1469-8137.1994.tb04259.x.

[8] N. A. Baker et al. “Electrostatics of nanosystems: application to microtubules and the ribosome.” *Proc Natl Acad Sci U S A* 98 (2001), pages 10037–10041. DOI: 10.1073/pnas.181342398.

[9] R. E. Bank and M. Holst. “A New Paradigm for Parallel Adaptive Meshing Algorithms.” *SIAM Review* 45 (2003), pages.

[10] P. Benkert, M. Künzli, and T. Schwede. "QMEAN server for protein model quality estimation." *Nucleic Acids Res* 37 (2009), W510–W514. DOI: 10.1093/nar/gkp322.

[11] P. Benkert, T. Schwede, and S. C. Tosatto. "QMEANclust: estimation of protein model quality by combining a composite scoring function with structural density information." *BMC Struct Biol* 9 (2009), page 35. DOI: 10.1186/1472-6807-9-35.

[12] R. M. Bennett-Lovsey et al. "Exploring the extremes of sequence/structure space with ensemble fold recognition in the program Phyre." *Proteins* 70 (2008), pages 611–625. DOI: 10.1002/prot.21688.

[13] H. Berendsen and W. Gunsteren. "Molecular Dynamics Simulations: Techniques and Approaches." In: *Molecular Liquids.* Edited by A. Barnes, W. Orville-Thomas, and J. Yarwood. Volume 135. NATO ASI Series. 1984, pages 475–500. DOI: 10.1007/978-94-009-6463-1_16.

[14] H. Berendsen, D. van der Spoel, and R. van Drunen. "GROMACS: A message-passing parallel molecular dynamics implementation." *Comput Phys Commun* 91 (1995), pages 43–56. DOI: 10.1016/0010-4655(95)00042-E.

[15] F. C. Bernstein et al. "The Protein Data Bank: a computer-based archival file for macromolecular structures." *J Mol Biol* 112 (1977), pages 535–542.

[16] R. Blum et al. "Function of phytochelatin synthase in catabolism of glutathione-conjugates." *Plant J* 49 (2007), pages 740–749. DOI: 10.1111/j.1365-313X.2006.02993.x.

[17] A. Bondi. "van der Waals Volumes and Radii." *J Phys Chem* 68 (1964), pages 441–451. DOI: 10.1021/j100785a001.

[18] A. Bräutigam et al. "Physiological characterization of cadmium-exposed Chlamydomonas reinhardtii." *Plant Cell Environ* 34 (2011), pages 2071–2082. DOI: 10.1111/j.1365-3040.2011.02404.x.

[19] K. Bryson, D. Cozzetto, and D. T. Jones. "Computer-assisted protein domain boundary prediction using the DomPred server." *Curr Protein Pept Sci* 8 (2007), pages 181–188. DOI: 10.2174/138920307780363415.

[20] D. W. A. Buchan et al. "Protein annotation and modelling servers at University College London." *Nucleic Acids Res* 38 (2010), W563–W568. DOI: 10.1093/nar/gkq427.

[21] G. Bussi, D. Donadio, and M. Parrinello. "Canonical sampling through velocity rescaling." *J Chem Phys* 126, 014101 (2007), page 014101. DOI: 10.1063/1.2408420.

[22] I. Cakmak. "Tansley Review No. 111. Possible Roles of Zinc in Protecting Plant Cells from Damage by Reactive Oxygen Species." *New Phytol* 146 (2000), pages 185–205.

[23] T. Caliński and J. Harabasz. "A dendrite method for cluster analysis." *Communications in Statistics* 3 (1974), pages 1–27. DOI: 10.1080/03610927408827101.

[25] R. L. Chaney et al. "Phytoremediation of soil metals." *Curr Opin Biotechnol* 8 (1997), pages 279–284. DOI: 10.1016/S0958-1669(97)80004-3.

[26] J. Cheng. "A multi-template combination algorithm for protein comparative modeling." *BMC Struct Biol* 8 (2008), page 18. DOI: 10.1186/1472-6807-8-18.

[27] J. Cheng and P. Baldi. "A machine learning information retrieval approach to protein fold recognition." *Bioinformatics* 22 (2006), pages 1456–1463. DOI: 10.1093/bioinformatics/btl102.

[28] N. K. Clay et al. "Glucosinolate metabolites required for an Arabidopsis innate immune response." *Science* 323 (2009), pages 95–101. DOI: 10.1126/science.1164627.

[29] W. W. Cleland and M. M. Kreevoy. "Low-barrier hydrogen bonds and enzymic catalysis." *Science* 264 (1994), pages 1887–1890.

[30] S. Clemens. "Evolution and function of phytochelatin synthases." *J Plant Physiol* 163 (2006), pages 319–332. DOI: 10.1016/j.jplph.2005.11.010.

[31] S. Clemens and D. Peršoh. "Multi-tasking phytochelatin synthases." *Plant Sci* 177 (2009), pages 266–271. DOI: 10.1016/j.plantsci.2009.06.008.

[32] S. Clemens et al. "Tolerance to toxic metals by a gene family of phytochelatin synthases from plants and yeast." *EMBO J* 18 (1999), pages 3325–3333. DOI: 10.1093/emboj/18.12.3325.

[33] C. Cobbett and P. Goldsbrough. "Phytochelatins and metallothioneins: roles in heavy metal detoxification and homeostasis." *Annu Rev Plant Biol* 53 (2002), pages 159–182. DOI: 10.1146/annurev.arplant.53.100301.135154.

[35] J. W. Cooley and J. W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series." *Math Comput* 19 (1965), pages 297–301.

[36] T. Darden, D. York, and L. Pedersen. "Particle mesh Ewald: An $N \log N$ method for Ewald sums in large systems." *J Chem Phys* 98 (1993), pages 10089–10092. DOI: 10.1063/1.464397.

[37] D. L. Davies and D. W. Bouldin. "A Cluster Separation Measure." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-1 (1979), pages 224–227. DOI: 10.1109/TPAMI.1979.4766909.

[38] P. Debye. "Näherungsformeln für die Zylinderfunktionen für große Werte des Arguments und unbeschränkt veränderliche Werte des Index." German. *Math Ann* 67 (1909), pages 535–558. DOI: 10.1007/BF01450097.

[39] P. Di Tommaso et al. "T-Coffee: a web server for the multiple sequence alignment of protein and RNA sequences using structural information and homology extension." *Nucleic Acids Res* 39 (2011), W13–W17. DOI: 10.1093/nar/gkr245.

[40] D. P. Dixon, A. Lapthorn, and R. Edwards. "Plant glutathione transferases." *Genome Biol* 3 (2002), reviews3004.1–reviews3004.10. DOI: 10.1186/gb-2002-3-3-reviews3004.

[41] T. J. Dolinsky et al. "PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations." *Nucleic Acids Res* 32 (2004), W665–W667. DOI: 10.1093/nar/gkh381.

[42] T. J. Dolinsky et al. "PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations." *Nucleic Acids Res* 35 (2007), W522–W525. DOI: 10.1093/nar/gkm276.

[43] Y. Duan et al. "A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations." *J Comput Chem* 24 (2003), pages 1999–2012. DOI: 10.1002/jcc.10349.

[44] S. R. Eddy. "Hidden Markov models." *Curr Opin Struct Biol* 6 (1996), pages 361–365.

[45] P. P. Ewald. "Die Berechnung optischer und elektrostatischer Gitterpotentiale." German. *Ann Phys* 369 (1921), pages 253–287. DOI: 10.1002/andp.19213690304.

[46] G. J. Fosmire. "Zinc toxicity." *Am J Clin Nutr* 51 (1990), pages 225–227.

[47] G. M. Gadd. "Metals, minerals and microbes: geomicrobiology and bioremediation." *Microbiology* 156 (2010), pages 609–643. DOI: 10.1099/mic.0.037143-0.

[48] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Upper Saddle River, NJ, USA, 1971.

[49] W. Gekeler et al. "Algae sequester heavy metals via synthesis of phytochelatin complexes." *Arch Microbiol* 150 (1988), pages 197–202. DOI: 10.1007/BF00425162.

[50] J. C. Gordon et al. “H++: a server for estimating pKas and adding missing hydrogens to macromolecules.” *Nucleic Acids Res* 33 (2005), W368–W371. DOI: 10.1093/nar/gki464.

[51] J. Gough et al. “Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure.” *J Mol Biol* 313 (2001), pages 903–919. DOI: 10.1006/jmbi.2001.5080.

[52] M. Greger and T. Landberg. “Use of Willow in Phytoextraction.” *Int J Phytoremed* 1 (1999), pages 115–123. DOI: 10.1080/15226519908500010.

[53] E. Grill, E. L. Winnacker, and M. H. Zenk. “Phytochelatins, a class of heavy-metal-binding peptides from plants, are functionally analogous to metallothioneins.” *Proc Natl Acad Sci U S A* 84 (1987), pages 439–443.

[54] E. Grill et al. “Phytochelatins, the heavy-metal-binding peptides of plants, are synthesized from glutathione by a specific gamma-glutamylcysteine dipeptidyl transpeptidase (phytochelatin synthase).” *Proc Natl Acad Sci U S A* 86 (1989), pages 6838–6842.

[55] W. F. van Gunsteren et al. *Biomolecular Simulation: The GROMOS96 Manual and User Guide.* Vdf Hochschulverlag AG an der ETH Zürich. Zürich, Switzerland, 1996, pages 1–1042.

[56] S. B. Ha et al. “Phytochelatin synthase genes from Arabidopsis and the yeast Schizosaccharomyces pombe.” *Plant Cell* 11 (1999), pages 1153–1164. DOI: 10.1105/tpc.11.6.1153.

[57] V. H. Hassinen et al. “Plant metallothioneins—metal chelators with ROS scavenging activity?” *Plant Biol (Stuttg)* 13 (2011), pages 225–232. DOI: 10.1111/j.1438-8677.2010.00398.x.

[58] B. Hess et al. “GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation.” *J Chem Theory Comput* 4 (2008), pages 435–447. DOI: 10.1021/ct700301q.

[59] B. Hess et al. “LINCS: A Linear Constraint Solver for Molecular Simulations.” *J Comput Chem* (1997). DOI: 10.1002/(SICI)1096-987X(199709)18:12<1463::AID-JCC4>3.0.CO;2-H.

[60] M. R. Hestenes and E. Stiefel. “Methods of Conjugate Gradients for Solving Linear Systems.” *J Res Natl Bureau Stand* 49 (1952), pages 409–436.

[61] M. Holst. “Adaptive Numerical Treatment of Elliptic Systems on Manifolds.” *Advances in Computational Mathematics* 15 (2001), pages 139–191. DOI: 10.1023/A:1014246117321.

[62] W. G. Hoover. “Canonical dynamics: Equilibrium phase-space distributions.” *Phys Rev A* 31 (1985), pages 1695–1697. DOI: 10.1103/PhysRevA.31.1695.

[63] R. Howden et al. “Cadmium-sensitive, cad1 mutants of Arabidopsis thaliana are phytochelatin deficient.” *Plant Physiol* 107 (1995), pages 1059–1066.

[64] J. D. Hunter. “Matplotlib: A 2D Graphics Environment.” *Comput Sci Eng* 9 (2007), pages 90–95. DOI: 10.1109/MCSE.2007.55.

[65] D. T. Jones. “GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences.” *J Mol Biol* 287 (1999), pages 797–815. DOI: 10.1006/jmbi.1999.2583.

[66] D. T. Jones. “Protein secondary structure prediction based on position-specific scoring matrices.” *J Mol Biol* 292 (1999), pages 195–202. DOI: 10.1006/jmbi.1999.3091.

[68] W. L. Jorgensen et al. “Comparison of simple potential functions for simulating liquid water.” *J Chem Phys* 79 (1983), pages 926–935. DOI: 10.1063/1.445869.

[69] R. H. Juang, K. F. McCue, and D. W. Ow. “Two purine biosynthetic enzymes that are required for cadmium tolerance in Schizosaccharomyces pombe utilize cysteine sulfinate in vitro.” *Arch Biochem Biophys* 304 (1993), pages 392–401. DOI: 10.1006/abbi.1993.1367.

[70] W. Kabsch and C. Sander. “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.” *Biopolymers* 22 (1983), pages 2577–2637. DOI: 10.1002/bip.360221211.

[71] J. H. Kägi. “Overview of metallothionein.” *Methods Enzymol* 205 (1991), pages 613–626. DOI: 10.1016/0076-6879(91)05145-L.

[72] J. A. de Knecht et al. “Evidence against a role for phytochelatins in naturally selected increased cadmium tolerance in Silene vulgaris (Moench) Garcke.” *New Phytol* 122 (1992), pages 681–688. DOI: 10.1111/j.1469-8137.1992.tb00097.x.

[73] Y. O. Korshunova et al. “The IRT1 protein from Arabidopsis thaliana is a metal transporter with a broad substrate range.” *Plant Mol Biol* 40 (1999), pages 37–44. DOI: 10.1023/A:1026438615520.

[74] U. Krämer. "Metal hyperaccumulation in plants." *Annu Rev Plant Biol* 61 (2010), pages 517–534. DOI: 10.1146/annurev-arplant-042809-112156.

[75] L. R. Lado, T. Hengl, and H. I. Reuter. "Heavy metals in European soils: A geostatistical analysis of the FOREGS Geochemical database." *Geoderma* 148 (2008), pages 189–199. DOI: 10.1016/j.geoderma.2008.09.020.

[76] B. Lane, R. Kajioka, and T. Kennedy. "The wheat-germ Ec protein is a zinc-containing metallothionein." *Biochem Cell Biol* 65 (1987), pages 1001–1005. DOI: 10.1139/o87-131.

[77] M. M. Lasat. "Phytoextraction of metals from contaminated soil: A review of plant/soil/metal interaction and assessment of pertinent agronomic issues." *Journal of Hazardous Substance Research* 2 (2000), pages 1–25.

[78] A. Leach. *Molecular Modelling: Principles and Applications*. 2nd edition. 9, 2001.

[79] Y.-F. Lin and M. G. M. Aarts. "The molecular mechanism of zinc and cadmium stress response in plants." *Cell Mol Life Sci* 69 (2012), pages 3187–3206. DOI: 10.1007/s00018-012-1089-z.

[80] E. Lindahl, B. Hess, and D. van der Spoel. "GROMACS 3.0: a package for molecular simulation and trajectory analysis." *J Mol Model* 7 (2001), pages 306–317. DOI: 10.1007/s008940100045.

[81] M. Lingenheil et al. "The "Hot-Solvent/Cold-Solute" Problem Revisited." *J Chem Theory Comput* 4 (2008), pages 1293–1306. DOI: 10.1021/ct8000365.

[82] X. Liu, K. Fan, and W. Wang. "The number of protein folds and their distribution over families in nature." *Proteins* 54 (2004), pages 491–499. DOI: 10.1002/prot.10514.

[83] E. Lombi et al. "Physiological evidence for a high-affinity cadmium transporter highly expressed in a Thlaspi caerulescens ecotype." *New Phytol* 149 (2001), pages 53–60. DOI: 10.1046/j.1469-8137.2001.00003.x.

[84] F. London. "Zur Theorie und Systematik der Molekularkräfte." German. *Z Phys A* 63 (1, 1930), pages 245–279. DOI: 10.1007/bf01421741.

[85] S. C. Lovell et al. "Structure validation by C_α geometry: ϕ, ψ and C_β deviation." *Proteins* 50 (2003), pages 437–450. DOI: 10.1002/prot.10286.

[86] P. Madejón et al. "Bioaccumulation of As, Cd, Cu, Fe and Pb in wild grasses affected by the Aznalcóllar mine spill (SW Spain)." *Sci Total Environ* 290 (2002), pages 105–120. DOI: 10.1016/S0048-9697(01)01070-1.

[87] M. S. Madhusudhan et al. “Alignment of multiple protein structures based on sequence and structure features.” *Protein Eng Des Sel* 22 (2009), pages 569–574. DOI: 10.1093/protein/gzp040.

[88] T. Maier et al. “Localization and functional characterization of metal-binding sites in phytochelatin synthases.” *Planta* 218 (2003), pages 300–308. DOI: 10.1007/s00425-003-1091-7.

[89] E. Marentes and W. E. Rauser. “Different proportions of cadmium occur as Cd-binding phytochelatin complexes in plants.” *Physiol Plant* 131 (2007), pages 291–301. DOI: 10.1111/j.1399-3054.2007.00960.x.

[90] S. Matsumoto et al. “Functional analysis of phytochelatin synthase from Arabidopsis thaliana and its expression in Escherichia coli and Saccharomyces cerevisiae.” *Sci Technol Adv Mater* 5 (2004), pages 377–381. DOI: 10.1016/j.stam.2004.01.005.

[92] F. Melo and E. Feytmans. “Assessing protein structures with a non-local atomic interaction energy.” *J Mol Biol* 277 (1998), pages 1141–1152. DOI: 10.1006/jmbi.1998.1665.

[93] D. G. Mendoza-Cózatl et al. “Phytochelatin-cadmium-sulfide high-molecular-mass complexes of Euglena gracilis.” *FEBS J* 273 (2006), pages 5703–5713. DOI: 10.1111/j.1742-4658.2006.05558.x.

[94] D. Mendoza-Cózatl et al. “Sulfur assimilation and glutathione metabolism under cadmium stress in yeast, protists and plants.” *FEMS Microbiol Rev* 29 (2005), pages 653–671. DOI: 10.1016/j.femsre.2004.09.004.

[95] T. Mi et al. “Minimotif Miner 3.0: database expansion and significantly improved reduction of false-positive predictions from consensus sequences.” *Nucleic Acids Res* 40 (2012), pages D252–D260. DOI: 10.1093/nar/gkr1189.

[96] J. Moult et al. “Critical assessment of methods of protein structure prediction (CASP)–round IX.” *Proteins* 79 Suppl 10 (2011), pages 1–5. DOI: 10.1002/prot.23200.

[97] A. G. Murzin et al. “SCOP: a structural classification of proteins database for the investigation of sequences and structures.” *J Mol Biol* 247 (1995), pages 536–540. DOI: 10.1006/jmbi.1995.0159.

[98] N. Mutoh and Y. Hayashi. “Isolation of mutants of Schizosaccharomyces pombe unable to synthesize cadystin, small cadmium-binding peptides.” *Biochem Biophys Res Commun* 151 (1988), pages 32–39.

[99] J. Myers et al. "A simple clustering algorithm can be accurate enough for use in calculations of pKs in macromolecules." *Proteins* 63 (2006), pages 928–938. DOI: 10.1002/prot.20922.

[100] K. Nagel, U. Adelmeier, and J. Voigt. "Subcellular distribution of cadmium in the unicellular green alga Chlamydomonas reinhardtii." *J Plant Physiol* 149 (1996), pages 86–90. DOI: 10.1016/S0176-1617(96)80178-7.

[101] S. B. Needleman and C. D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *J Mol Biol* 48 (1970), pages 443–453. DOI: 10.1016/0022-2836(70)90057-4.

[102] E. Nieboer and D. H. Richardson. "The replacement of the nondescript term 'heavy metals' by a biologically and chemically significant classification of metal ions." *Environ Poll B* 1 (1980), pages 3–26. DOI: 10.1016/0143-148X(80)90017-8.

[103] S. Nosé. "A molecular dynamics method for simulations in the canonical ensemble." *Mol Phys* 52 (1984), pages 255–268. DOI: 10.1080/00268978400101201.

[104] S. Nosé. "A unified formulation of the constant temperature molecular dynamics methods." *J Chem Phys* 81 (1984), pages 511–519. DOI: 10.1063/1.447334.

[105] S. Nosé and M. Klein. "Constant pressure molecular dynamics for molecular systems." *Mol Phys* 50 (1983), pages 1055–1076. DOI: 10.1080/00268978300102851.

[106] C. Notredame, D. G. Higgins, and J. Heringa. "T-Coffee: A novel method for fast and accurate multiple sequence alignment." *J Mol Biol* 302 (2000), pages 205–217. DOI: 10.1006/jmbi.2000.4042.

[107] A. Onufriev, D. Bashford, and D. A. Case. "Exploring protein native states and large-scale conformational changes with a modified generalized born model." *Proteins* 55 (2004), pages 383–394. DOI: 10.1002/prot.20033.

[108] C. Oostenbrink et al. "A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6." *J Comput Chem* 25 (2004), pages 1656–1676. DOI: 10.1002/jcc.20090.

[109] C. A. Orengo et al. "CATH—a hierarchic classification of protein domain structures." *Structure* 5 (1997), pages 1093–1108.

[110] D. F. Ortiz et al. "Heavy metal tolerance in the fission yeast requires an ATP-binding cassette-type vacuolar membrane transporter." *EMBO J* 11 (1992), pages 3491–3499.

[111] D. F. Ortiz et al. “Transport of metal-binding peptides by HMT1, a fission yeast ABC-type vacuolar membrane protein.” *J Biol Chem* 270 (1995), pages 4721–4728. DOI: 10.1074/jbc.270.9.4721.

[112] Y. Osaki et al. “Characterization of phytochelatin synthase produced by the primitive red alga Cyanidioschyzon merolae.” *Metallomics* 1 (2009), pages 353–358. DOI: 10.1039/b823013g.

[113] M. Parrinello and A. Rahman. “Polymorphic transitions in single crystals: A new molecular dynamics method.” *J Appl Phys* 52 (1981), pages 7182–7190. DOI: 10.1063/1.328693.

[114] W. R. Pearson and D. J. Lipman. “Improved tools for biological sequence comparison.” *Proc Natl Acad Sci U S A* 85 (1988), pages 2444–2448.

[115] N. S. Pence et al. “The molecular physiology of heavy metal transport in the Zn/Cd hyperaccumulator Thlaspi caerulescens.” *Proc Natl Acad Sci U S A* 97 (2000), pages 4956–4960. DOI: 10.1073/pnas.97.9.4956.

[116] J. Peng and J. Xu. “RaptorX: exploiting structure information for protein alignment by statistical inference.” *Proteins* 79 Suppl 10 (2011), pages 161–171. DOI: 10.1002/prot.23175.

[117] F. Pérez and B. E. Granger. “IPython: a System for Interactive Scientific Computing.” *Comput Sci Eng* 9 (2007), pages 21–29.

[118] M. Punta et al. “The Pfam protein families database.” *Nucleic Acids Res* 40 (2012), pages D290–D301. DOI: 10.1093/nar/gkr1065.

[119] A. Rahman. “Correlations in the Motion of Atoms in Liquid Argon.” *Phys Rev* 136 (1964), A405–A411. DOI: 10.1103/PhysRev.136.A405.

[120] A. Randall and P. Baldi. “SELECTpro: effective protein model selection using a structure-based energy function resistant to BLUNDERs.” *BMC Struct Biol* 8 (2008), page 52. DOI: 10.1186/1472-6807-8-52.

[121] W. E. Rauser. “Roots of Maize Seedlings Retain Most of their Cadmium Through Two Complexes.” *Journal of Plant Physiology* 156 (2000), pages 545–551. DOI: 10.1016/S0176-1617(00)80171-6.

[122] P. A. Rea. “Phytochelatin synthase: of a protease a peptide polymerase made.” *Physiol Plant* 145 (2012), pages 154–164. DOI: 10.1111/j.1399-3054.2012.01571.x.

[123] P. A. Rea. "Plant ATP-binding cassette transporters." *Annu Rev Plant Biol* 58 (2007), pages 347–375. DOI: 10.1146/annurev.arplant.57.032905.105406.

[124] P. A. Rea, O. K. Vatamaniuk, and D. J. Rigden. "Weeds, worms, and more. Papain's long-lost cousin, phytochelatin synthase." *Plant Physiol* 136 (2004), pages 2463–2474. DOI: 10.1104/pp.104.048579.

[125] M. Remmert et al. "HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment." *Nat Methods* 9 (2012), pages 173–175. DOI: 10.1038/nmeth.1818.

[126] N. D. Romanyuk et al. "Mutagenic definition of a papain-like catalytic triad, sufficiency of the N-terminal domain for single-site core catalytic enzyme acylation, and C-terminal domain for augmentative metal activation of a eukaryotic phytochelatin synthase." *Plant Physiol* 141 (2006), pages 858–869. DOI: 10.1104/pp.106.082131.

[127] A. Rosakis and W. Köster. "Divalent metal transport in the green microalga Chlamydomonas reinhardtii is mediated by a protein similar to prokaryotic Nramp homologues." *BioMetals* 18 (2005), pages 107–120. DOI: 10.1007/s10534-004-2481-4.

[128] S. Rosales-Mendoza, L. M. T. Paz-Maldonado, and R. E. Soria-Guerra. "Chlamydomonas reinhardtii as a viable platform for the production of recombinant proteins: current status and perspectives." *Plant Cell Rep* 31 (2012), pages 479–494. DOI: 10.1007/s00299-011-1186-8.

[130] B. Rost. "Twilight zone of protein sequence alignments." *Protein Eng* 12 (1999), pages 85–94.

[131] A. Roy, A. Kucukural, and Y. Zhang. "I-TASSER: a unified platform for automated protein structure and function prediction." *Nat Protoc* 5 (2010), pages 725–738. DOI: 10.1038/nprot.2010.5.

[132] A. Roy et al. "A protocol for computer-based protein structure and function prediction." *J Vis Exp* (2011), e3259. DOI: 10.3791/3259.

[133] R. Ruotolo et al. "Domain organization of phytochelatin synthase: functional properties of truncated enzyme species identified by limited proteolysis." *J Biol Chem* 279 (2004), pages 14686–14693. DOI: 10.1074/jbc.M314325200.

[134] J. Sachs. *Handbuch der Experimental-Physiologie der Pflanzen: Untersuchungen über die allgemeinen Lebensbedingungen der Pflanzen und die Functionen ihrer Organe.* German. Handbuch der physiologischen Botanik. 1865.

[135] A. Sali and T. L. Blundell. “Comparative protein modelling by satisfaction of spatial restraints.” *J Mol Biol* 234 (1993), pages 779–815. DOI: 10.1006/jmbi.1993.1626.

[136] D. E. Salt et al. “Phytoremediation: a novel strategy for the removal of toxic metals from the environment using plants.” *Biotechnology (N Y)* 13 (1995), pages 468–474. DOI: 10.1038/nbt0595-468.

[137] H. Schat et al. “The role of phytochelatins in constitutive and adaptive heavy metal tolerances in hyperaccumulator and non-hyperaccumulator metallophytes.” *J Exp Bot* 53 (2002), pages 2381–2392. DOI: 10.1093/jxb/erf107.

[138] H. A. Scheraga, M. Khalili, and A. Liwo. “Protein-folding dynamics: overview of molecular simulation techniques.” *Annu Rev Phys Chem* 58 (2007), pages 57–83. DOI: 10.1146/annurev.physchem.58.032806.104614.

[140] B. Schuster-Böckler, J. Schultz, and S. Rahmann. “HMM Logos for visualization of protein families.” *BMC Bioinform* 5 (2004), page 7. DOI: 10.1186/1471-2105-5-7.

[141] T. Schwede et al. “SWISS-MODEL: An automated protein homology-modeling server.” *Nucleic Acids Res* 31 (2003), pages 3381–3385.

[142] J. Shao et al. “Clustering Molecular Dynamics Trajectories: 1. Characterizing the Performance of Different Clustering Algorithms.” *J Chem Theory Comput* 3 (2007), pages 2312–2334. DOI: 10.1021/ct700119m.

[143] M.-Y. Shen and A. Sali. “Statistical potential for assessment and prediction of protein structures.” *Protein Sci* 15 (2006), pages 2507–2524. DOI: 10.1110/ps.062416606.

[144] M. R. Shirts et al. “Accurate and efficient corrections for missing dispersion interactions in molecular simulations.” *J Phys Chem B* 111 (2007), pages 13052–13063. DOI: 10.1021/jp0735987.

[145] D. Shortle, K. T. Simons, and D. Baker. “Clustering of low-energy conformations near the native structures of small proteins.” *Proc Natl Acad Sci U S A* 95 (1998), pages 11158–11162. DOI: 10.1073/pnas.95.19.11158.

[146] C. J. A. Sigrist et al. “PROSITE: a documented database using patterns and profiles as motif descriptors.” *Brief Bioinform* 3 (2002), pages 265–274.

[147] T. F. Smith and M. S. Waterman. “Identification of common molecular subsequences.” *J Mol Biol* 147 (1981), pages 195–197. DOI: 10.1016/0022-2836(81)90087-5.

[148] J. Söding. “Protein homology detection by HMM-HMM comparison.” *Bioinformatics* 21 (2005), pages 951–960. DOI: 10.1093/bioinformatics/bti125.

[149] J. Söding, A. Biegert, and A. N. Lupas. “The HHpred interactive server for protein homology detection and structure prediction.” *Nucleic Acids Res* 33 (2005), W244–W248. DOI: 10.1093/nar/gki408.

[150] D. M. Speiser et al. “Purine biosynthetic genes are required for cadmium tolerance in Schizosaccharomyces pombe.” *Mol Cell Biol* 12 (1992), pages 5301–5310.

[151] S. J. Stohs and D. Bagchi. “Oxidative mechanisms in the toxicity of metal ions.” *Free Radic Biol Med* 18 (1995), pages 321–336.

[152] K. Straif et al. “A review of human carcinogens—part C: metals, arsenic, dusts, and fibres.” *Lancet Oncol* 10 (2009), pages 453–454. DOI: 10.1016/S1470-2045(09)70134-2.

[153] D. Swandulla and C. M. Armstrong. “Calcium channel block by cadmium in chicken sensory neurons.” *Proc Natl Acad Sci U S A* 86 (1989), pages 1736–1740. DOI: 10.1073/pnas.86.5.1736.

[154] W. C. Swope et al. “A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters.” *J Chem Phys* 76 (1982), pages 637–649.

[155] M. Takahashi et al. “Role of nicotianamine in the intracellular delivery of metals and plant reproductive development.” *Plant Cell* 15 (2003), pages 1263–1280. DOI: 10.1105/tpc.010256.

[156] D. L. Theobald and P. A. Steindel. “Optimal simultaneous superpositioning of multiple structures with missing data.” *Bioinformatics* 28 (2012), pages 1972–1979. DOI: 10.1093/bioinformatics/bts243.

[157] D. L. Theobald and D. S. Wuttke. “Accurate structural correlations from maximum likelihood superpositions.” *PLoS Comput Biol* 4 (2008), e43. DOI: 10.1371/journal.pcbi.0040043.

[158] D. L. Theobald and D. S. Wuttke. “THESEUS: maximum likelihood superpositioning and analysis of macromolecular structures.” *Bioinformatics* 22 (2006), pages 2171–2172. DOI: 10.1093/bioinformatics/btl332.

[159] N. Tsuji et al. “Characterization of phytochelatin synthase-like protein encoded by alr0975 from a prokaryote, Nostoc sp. PCC 7120.” *Biochem Biophys Res Commun* 315 (2004), pages 751–755. DOI: 10.1016/j.bbrc.2004.01.122.

[160] N. Tsuji et al. “Comparative analysis of the two-step reaction catalyzed by prokaryotic and eukaryotic phytochelatin synthase by an ion-pair liquid chromatography assay.” *Planta* 222 (2005), pages 181–191. DOI: 10.1007/s00425-005-1513-9.

[161] B. K. Vallat, J. Pillardy, and R. Elber. “A template-finding algorithm and a comprehensive benchmark for homology modeling of proteins.” *Proteins* 72 (2008), pages 910–928. DOI: 10.1002/prot.21976.

[162] B. K. Vallat et al. “Building and assessing atomic models of proteins from structural templates: learning and benchmarks.” *Proteins* 76 (2009), pages 930–945. DOI: 10.1002/prot.22401.

[163] D. Van Der Spoel et al. “GROMACS: Fast, flexible, and free.” *J Comput Chem* 26 (2005), pages 1701–1718. DOI: 10.1002/jcc.20291.

[164] O. K. Vatamaniuk et al. “Mechanism of heavy metal ion activation of phytochelatin (PC) synthase: blocked thiols are sufficient for PC synthase-catalyzed transpeptidation of glutathione and related thiol peptides.” *J Biol Chem* 275 (2000), pages 31451–31459. DOI: 10.1074/jbc.M002997200.

[165] O. K. Vatamaniuk et al. “Phytochelatin synthase, a dipeptidyltransferase that undergoes multisite acylation with γ-glutamylcysteine during catalysis: stoichiometric and site-directed mutagenic analysis of arabidopsis thaliana PCS1-catalyzed phytochelatin synthesis.” *J Biol Chem* 279 (2004), pages 22449–22460. DOI: 10.1074/jbc.M313142200.

[166] O. K. Vatamaniuk et al. “Worms take the ’phyto’ out of ’phytochelatins’.” *Trends Biotechnol* 20 (2002), pages 61–64.

[167] L. Verlet. “Computer ”Experiments” on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules.” *Phys Rev* 159 (1967), pages 98–103. DOI: 10.1103/PhysRev.159.98.

[168] M. Vestergaard et al. “Chelation of cadmium ions by phytochelatin synthase: role of the cysteine-rich C-terminal.” *Anal Sci* 24 (2008), pages 277–281. DOI: 10.2116/analsci.24.277.

[169] A. Viterbi. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm.” *IEEE T Inform Theory* 13 (1967), pages 260–269. DOI: 10.1109/TIT.1967.1054010.

[170] D. Vivares, P. Arnoux, and D. Pignol. “A papain-like enzyme at work: native and acyl-enzyme intermediate structures in phytochelatin synthesis.” *Proc Natl Acad Sci U S A* 102 (2005), pages 18848–18853. DOI: 10.1073/pnas.0505833102.

[171] B. Volesky and Z. R. Holan. “Biosorption of heavy metals.” *Biotechnol Prog* 11 (1995), pages 235–250. DOI: 10.1021/bp00033a001.

[172] H.-C. Wang et al. “Phytochelatin synthase is regulated by protein phosphorylation at a threonine residue near its catalytic site.” *J Agric Food Chem* 57 (2009), pages 7348–7355. DOI: 10.1021/jf9020152.

[173] J. Wang, P. Cieplak, and P. A. Kollman. “How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules?” *J Comput Chem* 21 (2000), pages 1049–1074. DOI: 10.1002/1096-987X(200009)21:12<1049::AID-JCC3>3.0.CO;2-F.

[174] Z. Wang, J. Eickholt, and J. Cheng. “MULTICOM: a multi-level combination approach to protein structure prediction and its assessments in CASP8.” *Bioinformatics* 26 (2010), pages 882–888. DOI: 10.1093/bioinformatics/btq058.

[175] J. J. Ward et al. “Prediction and functional analysis of native disorder in proteins from the three kingdoms of life.” *J Mol Biol* 337 (2004), pages 635–645. DOI: 10.1016/j.jmb.2004.02.002.

[176] J. J. Ward et al. “The DISOPRED server for the prediction of protein disorder.” *Bioinformatics* 20 (2004), pages 2138–2139. DOI: 10.1093/bioinformatics/bth195.

[177] A. M. Waterhouse et al. “Jalview Version 2—a multiple sequence alignment editor and analysis workbench.” *Bioinformatics* 25 (2009), pages 1189–1191. DOI: 10.1093/bioinformatics/btp033.

[178] L. Woodcock. “Isothermal molecular dynamics calculations for liquid salts.” *Chem Phys Lett* 10 (1971), pages 257–261. DOI: 10.1016/0009-2614(71)80281-6.

[179] S. Wu and Y. Zhang. “LOMETS: a local meta-threading-server for protein structure prediction.” *Nucleic Acids Res* 35 (2007), pages 3375–3382. DOI: 10.1093/nar/gkm251.

[180] Y. Zhang and J. Skolnick. “The protein structure prediction problem could be solved using the current PDB library.” *Proc Natl Acad Sci U S A* 102 (2005), pages 1029–1034. DOI: 10.1073/pnas.0407152101.

[181] Y. Zhang and J. Skolnick. “TM-align: a protein structure alignment algorithm based on the TM-score.” *Nucleic Acids Res* 33 (2005), pages 2302–2309. DOI: 10.1093/nar/gki524.

[182] R. Zhou. “Free energy landscape of protein folding in water: explicit vs. implicit solvent.” *Proteins* 53 (2003), pages 148–161. DOI: 10.1002/prot.10483.

Online Sources

[24] D. Case et al. *AMBER 12*. University of California, San Francisco. 2012. URL: http://www.ambermd.org.

[34] C. Combet et al. *Geno3D: Automatic Modeling of Proteins' Three-Dimensional Structure*. 2010. URL: http://geno3d-pbil.ibcp.fr/cgi-bin/geno3d_automat.pl?page=/GENOHLP/genohlp_help2.html.

[67] E. Jones, T. Oliphant, and P. Peterson. *SciPy: Open source scientific tools for Python*. 2001. URL: http://www.scipy.org/.

[91] W. McKinney. *Pandas: A Python Data Analysis Library*. 2008. URL: http://pandas.pydata.org.

[129] G. van Rossum. *The Python Programming Language*. URL: http://www.python.org.

[139] L. Schrödinger. *The PyMOL Molecular Graphics System, Version 1.3r1*. 2010. URL: http://www.pymol.org.

A. Appendix

A.1. *Modi Operandi* of Modeling Servers

In this section, the workings of all used modeling servers are described in some detail.

A.1.1. HHpred

Taken from [149].

1. HHsearch to find suitable templates:
   a) Creation of HMMs of the query sequence by finding similar sequences using HHblits or Psi-Blast
   b) Comparing query HMM to HMMs generated for multiple structural databases
2. HHsearch also finds remotely homologous templates (since structural similarity is much more conserved than sequence similarity, these might still be good as templates)
3. The multiple alignment created by HHsearch is then fed into Modeller

A.1.2. I-Tasser

Taken from [131, 132].

1. Lomets, a threading meta-server, finds suitable templates and threads query sequence onto target structure
2. Lomets generates spacial restraints for further use by I-Tasser by structural alignment of conserved sequences
3. Building a structural model:
   - Aligned fragments are assembled directly

   - Non-aligned fragments are built using *ab initio* modeling

4. Minimization using the I-Tasser force field, which incorporates:
   - Hydrogen bonds
   - Knowledge-based energy terms
   - Sequence-based contact predictions (using Svmseq)
   - Spatial restraints collected in step 2

5. Creation of decoys

6. Clustering of decoys to identify possible native folds (using Spicker)

7. Creation of representative cluster centroids (only Cα and side-chain center of mass)

8. Repeat steps 3 to 8 twice

9. Creation of full atomic model from cluster centroids

10. Refinement of full atomic model by fragment-guided MD
    a) H-bond networks
    b) Backbone torsion angles
    c) Bond lengths
    d) Side-chain rotamer orientations

A.1.3. Raptor-X

Taken from [116].

1. Psi-Blast and HHpred to identify homologs, then regression tree-based scoring function to create sequence profile

2. Neff to evaluate sparsity of alignment: The sparser it is, the more valued is structural information over sequence alignment

3. Neff is also used to decide whether to use position-specific of context-specific gap penalty (context-specific depends on:
   - Predicted secodary structure type
   - Predicted solvent accessibility

   - Amino acid identity
   - Hydropathy count
   - Buriedness)

4. Ranking of all templates by predicted query-template alignment
   - Neural network-based approach
   - Quality of an alignment is defined as the TM-score of the 3D model built from this alignment using Modeller

5. Selection of mutually similar templates:
   a) Exclusion of all but top 20 templates
   b) Exclusion of templates with quality ≥ 10 % less than highest rated template
   c) Exclusion of templates whose TM-score with first-ranked template < 0.65

6. If there are no templates left, go to last step

7. If there are ≥ 2 template left: Creation of initial probabilistic alignment matrices (PAMs) for target-template pairs (a PAM encodes all possible alignments between two sequences)

8. Creation of structural alignment PAMs using TMalign and/or Matt

9. Iterative adjustment of to maximize consistency among all PAMs (i.e. to improve alignment quality)

10. Create 3D models from alignment(s) using Modeller

A.1.4. Phyre^2

Taken from the supplementary material to [12].

1. HHsearch to find close and remote sequence homologs

2. Prediction of query secondary structure by taking the average confidence score of PsiPred, SSPro and JNet prediction methods

3. Disorder prediction using Disopred

4. Profile creation by combining secondary structure prediction and sequence alignment

5. Profile is scanned against fold library, which is constructed in the following way:
    a) Sequences of structures deposited in PDB and SCOP…
    b) …are scanned against nonredundant sequence database
    c) Profile is constructed and deposited in “fold library”
    d) Known and predicted secondary structures are also incorporated in those profiles
6. Alignment process returns a score on which which alignments are ranked
7. Computation of E-values of ranked alignments
8. Top ten highest scoring alignments are used to create full 3D models
9. Missing/inserted regions are repaired using a loop library and reconstruction procedure
10. Side chains are placed on the model using a fast graph-based algorithm and side chain rotamer library

A.1.5. Multicom

Taken from [174].

1. Template identification by three methods and search against in-house databases:
    - Psi-Blast (PSSM)
    - HHsearch (HMMs)
    - Compass (profile)
    - Sensitive machine learning fold recognition method [27]
2. Ranking of templates according to E-value
3. Alignment of query to top 10 templates using Spem
4. Template combination in order to improve performance [26]:
    a) Query aligned to top template and other templates with E-value ≤ 10 to 20 and ≥ 75 % coverage)
    b) Fragments from alignments below E-value/coverage threshold
    c) Removal of top template, rinse and repeat

→ Generation of up to 10 multiple alignments

5. Model generation
    - Models with query–template alignment (QTA):
        a) Modeller 7v7 to create 10 models
        b) Model with minimum Modeller energy is returned as predicted model
    - Models without QTA:
        a) Rosetta to create 200 models
        b) Clustering by Rosetta
        c) Centroids of several large clusters are returned as predicted models
6. Model evaluation using ModelEvaluator; comparison of several model parameters with values predicted from sequence by means of the Scratch suite
    - Secondary structure
    - Solvent accessibility
    - Contact map
    - β-sheet topology
7. Calculation of Gdt_ts using a support vector machine approach
8. Ranking of predicted models using Gdt_ts

A.1.6. Swiss-Model

Taken from [141].

1. Creation of ExPDB database from PDB
    a) Removal of theoretical models
    b) Removal of low quality and Cα trace structures
    c) Addition of information for template selection to PDB header (e.g. probably quaternary structure, quality indicators such as empirical force field energy, Anolea mean force potential scores)
2. Template selection by Psi-BLASTing query sequence against sequences in ExPDB database

  - If templates cover distinct regions of target sequence (i.e. query has multiple domains), modeling process will be split into separate independent batches

3. Superposition of up to 5 template structures using an iterative least squares algorithm

4. Creation of structural alignment after removing incompatible templates (i.e. omitting structures with high Cα-RMSD to first template)

5. Local pairwise alignment of target sequence to main template structures

   - Placement of insertions and deletions optimized considering template structure context

6. Backbone atom positions of template structure are averaged

   → Templates weighted by sequence similarity to target sequence

   - Exclusion of significantly deviating atom positions

7. Construction of non-aligned regions using constraint space programming

   a) Best loop selected using a scoring scheme, which accounts for:

      - Force field energy
      - Steric hindrance
      - Favorable interactions (i.e. hydrogen bond formation)

   b) If no suitable loop can be identified, flanking residues are included to allow for more flexibility

   c) For loops > 10 residues: Selection of loop conformation from loop library derived from experimental structures

8. Side chain modeling based on weighted positions of corresponding residues in template structures

   a) Starting with conserved residues

   b) Iso-sterical replacement with template structure side chains

   c) Possible side chain confirmations are selected from backbone dependent rotamer library

   d) Scoring function (accounts for hydrogen bonds, disulfide bridges) applied to select most likely conformation

9. Energy minimization using `GROMOS96` force field

A.1.7. Geno3D

Taken from [34].

1. Psi-Blast against PDB entries to find suitable template and generate a QTA
2. User selects templates from list
3. Secondary structure is predicted for query and templates
4. Secondary structure information is used to recompute QTA
5. Calculation of distance and dihedral angle restraints from QTA
6. Statistical restraints are used for gaps
7. Restraints are used as input for Cns software
8. Cns produces models that fit these restraints as well as possible

A.1.8. Loopp

Taken from [161, 162].

1. Phase I—template identification, using:
    - Site-specific residue frequency profile (Psi-Blast)
    - Raw sequence and profile of the template generated from target profile database
    - Template secondary structure (Sable [query]/Dssp [target])
    - Exposed surface area of template (Sable [query]/Dssp [target])
    - Thom2 contacts between structural sites

    → Combination of those factors to obtain 20 "features"
2. Evaluation of all 20 features with all known protein structures in the database
3. Tree-based approach that returns database hits from one search branch and performs remaining search branches on the remaining entries template database:
    a) Psi-Blast with 3 iterations:
        i. Rank hits according to SC and return top scoring hits
        ii. Remove hits from search database

    b) Comparison of all features
        i. Rank hits according to SC and return top scoring hits
        ii. Remove hits from search database
    c) Exactly the same as previous branch
        i. Rank hits according to SC and return top scoring hits
        ii. Remove hits from search database
    d) Comparison of same 20 features, transformed to a uniform distribution
        i. Rank hits according to SC and return top scoring hits
        ii. Remove hits from search database
    e) Comparison of a quadratic expansion of 12 uniformly distributed features
        i. Rank hits according to SC and return top scoring hits

4. Create QTA using Ssaln and build models using Modeller

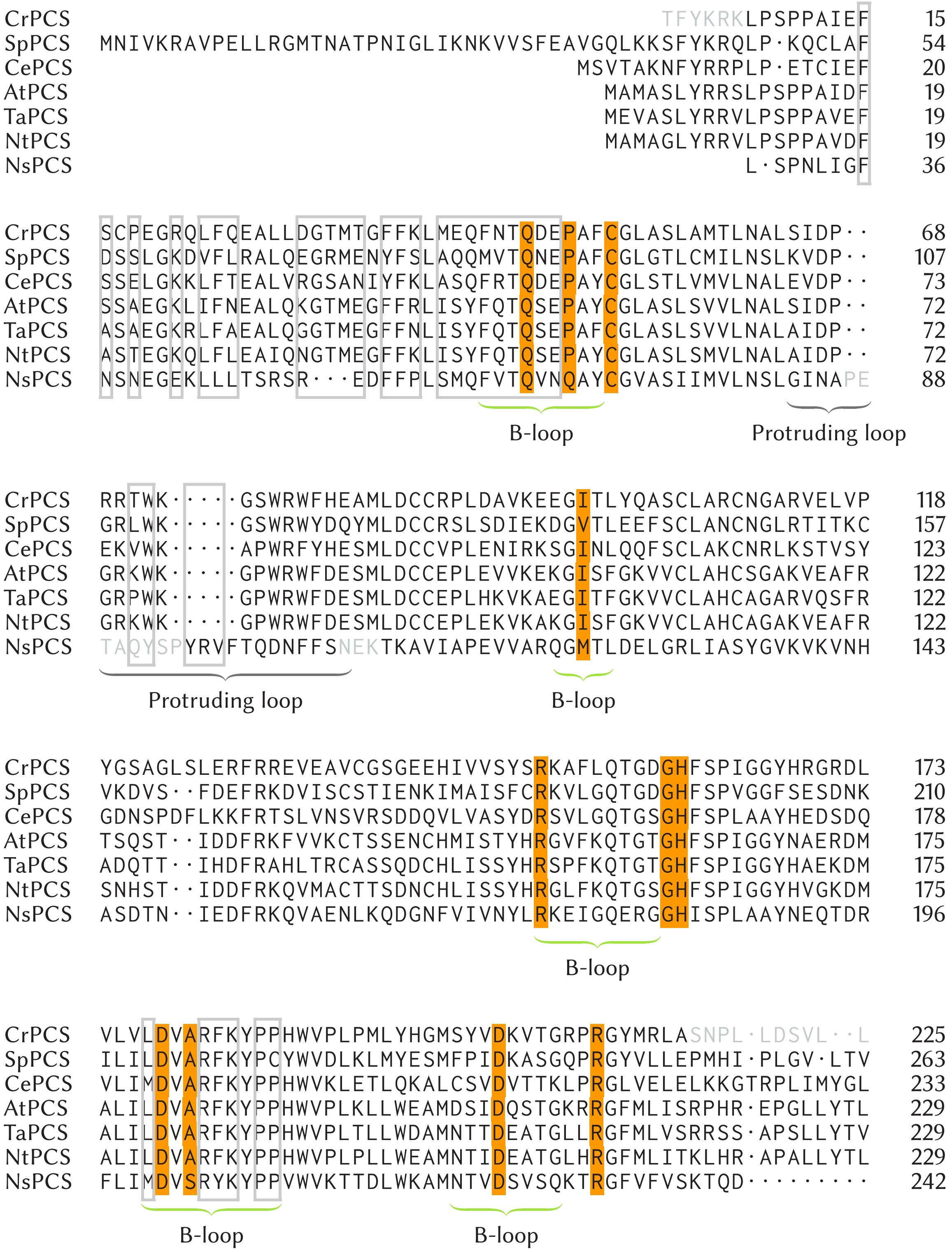
5. Clustering and ranking of returned hits according to SC and TM-score

6. Phase II—template modeling. Repeat previous branches 0, 1, and 2 (the last one five times), but also use the following measures for creating scores:
    - Z-score (more sensitive, but 100 to 1000 times more computationally intensive)
    - Eneall (all-atom potential)
    - Te13 (residue-based contact potential)
    - Fready (coarse-grained potential for pair interactions)
    - Sift (assessment score; includes multiple measures; see [162, page 5])

A.2. CrPCS Alignment from Vivares, Arnoux, and Pignol [170]

This is the initial alignment which was used to as a starting point for the manual modeling process and subsequently refined. As described in Figure 4.1, an orange background indicates residues in the active site, and gray frames mark residues at the putative dimeric interface. Light gray regions in the CrPCS sequence were cut out from the resulting model as they had to be modeled *ab initio*, light gray regions in the NsPCS sequence were not present in the

2BU3 template. The locations of these residues are taken from Vivares, Arnoux, and Pignol [170].



```
CrPCS                                           TFYKRKLPSPPAIEF    15
SpPCS   MNIVKRAVPELLRGMTNATPNIGLIKNKVVSFEAVGQLKKSFYKRQLP·KQCLAF    54
CePCS                                  MSVTAKNFYRRPLP·ETCIEF    20
AtPCS                                    MAMASLYRRSLPSPPAIDF    19
TaPCS                                    MEVASLYRRVLPSPPAVEF    19
NtPCS                                    MAMAGLYRRVLPSPPAVDF    19
NsPCS                                              L·SPNLIGF    36

CrPCS   SCPEGRQLFQEALLDGTMTGFFKLMEQFNTQDEPAFCGLASLAMTLNALSIDP··    68
SpPCS   DSSLGKDVFLRALQEGRMENYFSLAQQMVTQNEPAFCGLGTLCMILNSLKVDP··   107
CePCS   SSELGKKLFTEALVRGSANIYFKLASQFRTQDEPAYCGLSTLVMVLNALEVDP··    73
AtPCS   SSAEGKLIFNEALQKGTMEGFFRLISYFQTQSEPAYCGLASLSVVLNALSIDP··    72
TaPCS   ASAEGKRLFAEALQGGTMEGFFNLISYFQTQSEPAFCGLASLSVVLNALAIDP··    72
NtPCS   ASTEGKQLFLEAIQNGTMEGFFKLISYFQTQSEPAYCGLASLSMVLNALAIDP··    72
NsPCS   NSNEGEKLLLTSRSR···EDFFPLSMQFVTQVNQAYCGVASIIMVLNSLGINAPE    88
                                      B-loop            Protruding loop

CrPCS   RRTWK·····GSWRWFHEAMLDCCRPLDAVKEEGITLYQASCLARCNGARVELVP   118
SpPCS   GRLWK·····GSWRWYDQYMLDCCRSLSDIEKDGVTLEEFSCLANCNGLRTITKC   157
CePCS   EKVWK·····APWRFYHESMLDCCVPLENIRKSGINLQQFSCLAKCNRLKSTVSY   123
AtPCS   GRKWK·····GPWRWFDESMLDCCEPLEVVKEKGISFGKVVCLAHCSGAKVEAFR   122
TaPCS   GRPWK·····GPWRWFDESMLDCCEPLHKVKAEGITFGKVVCLAHCAGARVQSFR   122
NtPCS   GRKWK·····GPWRWFDESMLDCCEPLEKVKAKGISFGKVVCLAHCAGAKVEAFR   122
NsPCS   TAQYSPYRVFTQDNFFSNEKTKAVIAPEVVARQGMTLDELGRLIASYGVKVKVNH   143
          Protruding loop                 B-loop

CrPCS   YGSAGLSLERFRREVEAVCGSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDL   173
SpPCS   VKDVS··FDEFRKDVISCSTIENKIMAISFCRKVLGQTGDGHFSPVGGFSESDNK   210
CePCS   GDNSPDFLKKFRTSLVNSVRSDDQVLVASYDRSVLGQTGSGHFSPLAAYHEDSDQ   178
AtPCS   TSQST··IDDFRKFVVKCTSSENCHMISTYHRGVFKQTGTGHFSPIGGYNAERDM   175
TaPCS   ADQTT··IHDFRAHLTRCASSQDCHLISSYHRSPFKQTGTGHFSPIGGYHAEKDM   175
NtPCS   SNHST··IDDFRKQVMACTTSDNCHLISSYHRGLFKQTGSGHFSPIGGYHVGKDM   175
NsPCS   ASDTN··IEDFRKQVAENLKQDGNFVIVNYLRKEIGQERGGHISPLAAYNEQTDR   196
                                          B-loop

CrPCS   VLVLDVARFKYPPHWVPLPMLYHGMSYVDKVTGRPRGYMRLASNPL·LDSVL··L   225
SpPCS   ILILDVARFKYPCYWVDLKLMYESMFPIDKASGQPRGYVLLEPMHI·PLGV·LTV   263
CePCS   VLIMDVARFKYPPHWVKLETLQKALCSVDVTTKLPRGLVELELKKGTRPLIMYGL   233
AtPCS   ALILDVARFKYPPHWVPLKLLWEAMDSIDQSTGKRRGFMLISRPHR·EPGLLYTL   229
TaPCS   ALILDVARFKYPPHWVPLTLLWDAMNTTDEATGLLRGFMLVSRRSS·APSLLYTV   229
NtPCS   ALILDVARFKYPPHWVPLPLLWEAMNTIDEATGLHRGFMLITKLHR·APALLYTL   229
NsPCS   FLIMDVSRYKYPPVWVKTTDLWKAMNTVDSVSQKTRGFVFVSKTQD·········   242
            B-loop               B-loop
```

```
CrPCS   TCDVR··SAPEDWRPAE·AFVRSGAAAL                                250
SpPCS   GLNKY··SWRNVSKHILQQAATVKNADNLAE··ILLS·········INQSSIPLI     305
CePCS   ··KAYVNINDSDFATSVIS···························WNQFLLCD·     258
AtPCS   SCKDE··SWIEIAKYLKEDVPRLVSSQHVDSVEKIISVVFKSLPSNFNQFIRWVA     282
TaPCS   SCGHG··SWKSMAKYCVEDVPNLLKDESLDNVTTLLSRLVESLPANAGDLIKCVI     282
NtPCS   SCKHE··SWVTISKHLMDDLPVLLSSENVKGIKDVLSTVLSNLPSNFVEFIKWIA     282
NsPCS   ·······················································     242

CrPCS                                                               250
SpPCS   QERSNSSK····SG·········································     315
CePCS   ······PLED···························DEEEFQLCCRKFGQC···     277
AtPCS   EIRITEDSNQNLSAEEKSRLKLKQLVLKEVHETELFKHINKFL·········STV     328
TaPCS   EVRRKEEGESSLSKEEKERLFLKEKVLQQIRDTDLFRVVHELQYPKGLCGSCSSS     337
NtPCS   EVRRQEENGQNLSDEEKGRLAIKEEVLKQVQDTPLYKHVTSILFSKNSICQ SKA     336
NsPCS   ·······················································     242

CrPCS                                                               250
SpPCS   ·····························DFEHFKECIRST··············     327
CePCS   ·······FAPHAMCCTQKTFDADQ·········KNSCTECSTDQNEACKMICSEI     316
AtPCS   GYEDSLTYAAAKACCQGAEILSGSP·S·KEFCCRETCVKCIKGPDDSEGTVVTGV     381
TaPCS   SDEDSLAEIAATVCCQGAAFLSGNLVSRDGFCCRETCIKCIEANGDGLKTVISGT     392
NtPCS   ASDSSLANVAANICCQGAGLFAGRSGSSDRFCCLQTCVRCYRATGGNSATVVSGT     391
NsPCS   ·······················································     242

CrPCS                                                               250
SpPCS   ·······KTYHLFLKHTN·················TNVEYITMAFWAIFSLPMIQ     358
CePCS   ···R·RTRFAEVFSSSAVAALLIAWPFEKGYSERSDRIGN···············     352
AtPCS   VVRDGNEQKVDLLVPSTQTECECGP······EATYPAGNDVFTALLLALPPQTWS     430
TaPCS   VVSKGNEQAVDLLLPTSSSKTSLCNSNLKSKIVKYPSSTDVLTVLLLVLQPNTWL     447
NtPCS   VVNGNGEQGVDVLVPTSLAKTSCCPSGQAGCSPMHPASNDVLTALLLALPPHTWS     446
NsPCS   ·······················································     242

CrPCS                                                               250
SpPCS   KALPKGVLEEIQSLLKEVEISEI·NTQLTALKKQLDSLTHCCKT·······DTGC     405
CePCS   ······LAEKYKNEFSAETM·····NEMNQLTTQIRTLISCSKPPVVININKPDA     396
AtPCS   GIKDQALMHEMKQLISMASLPTLLQEEVLHLRRQLQLLKRCQEN·······K·EE     477
TaPCS   GIKDENVKAEFQSLVSTDNLPDLLKQEILHLRRQLHYLAGCKGQ·······E·AC     494
NtPCS   RIKDTKVLQEIENLVSAENLPPLLQEEILHLRGQFLLLKKCKDN       K VE     493
NsPCS   ·······················································     242

CrPCS                              250
SpPCS   CSSSCCKN·············T     414
CePCS   TSNKCCKNKIGQSCACANDVNL     418
AtPCS   DDLAAPA··············Y     485
TaPCS   QEPPS················P     500
NtPCS   EDLAAPP              F     501
NsPCS   ······················     242
```

A.3. Merged Manual and Automatically Created Alignments

This is the full alignment whose fingerprint was presented in Figure 4.2. This alignment is used to compare the accuracy of the automated alignment algorithms used by the different modeling servers. Some of the servers use several alignments and weigh them according to predicted accuracy; in those cases, all alignments are reproduced here in order of precedence.As in the previous section, an orange background in the “manual” sequence marks active site residues, and a green background marks the B-loops.

```
2bu3a                  L··S·P·NLIGFNSNEGEKLL···LTS···R···SR··     51
2btwb                   L·S·P·NLIGFNSNEGEKLL···LTS···R···SR··     51
3k8ua                                                              0
manual          TFYKRK··LPS·P·PAIEFSCPEGRQLF···QEA···L···LDGT     32
geno3d                    S·P·PAIEFSCPEGRQLFQEALLD···G···TM··     25
hhpred                 L··PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     27
itasser1        TFYKRK··L·PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     33
itasser2        TFYKRK·L··PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     33
itasser3        TFYKR··KL·PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     33
itasser4        TFYKRK·LP·S·P·PAIEFSCPEGRQLF···QEALLDG···TM··     33
itasser5        TFYKR··KL·PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     33
itasser6        TFYKR··KL·PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     33
itasser7        TFYKRK·L··PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     33
itasser8        TFYKR··KL·PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     33
itasser9        TFYKRK·LP·S·P·PAIEFSCPEGRQLF···QEA···L···LDGT     32
itasser10       TFYKR··KL·PSP·PAIEFSCPEGRQLF···QEALLDG···TM··     33
loopp                  P··S·P·PAIEFSCPEGRQLF···QEALLDG···TM··     26
loopp2                 P··S·P·PAIEFSCPEGRQLF···QEALLDG···TM··     26
loopp5                  P·S·P·PAIEFSCPEGRQLF···QEALLDG···TM··     26
multicom        TFYKRKLP··S·P·PAIEFSCPEGRQLF···QEALLDG···TM··     33
phyre2int       TFYKRK·L··P·SPPAIEFSCPEGRQLF···QEALLDG···TM··     33
phyre2nrm       TFYKRK·L··P·SPPAIEFSCPEGRQLF···QEALLDG···TM··     33
phyre2o2o       TFYKRK·L··P·SPPAIEFSCPEGRQLF···QEALLDG···TM··     33
raptorx            KRKLP··S·P·PAIEFSCPEGRQLF···QEA···LLDGTM··     30
swissmodel             L··P·SPPAIEFSCPEGRQLF···QEALLDG···TM··     27
```

```
2bu3a        ·EDFFPL·SMQ·FVTQVNQAYC·GVASI·IMVLNSLGINAPETAQ    91
2btwb        ·EDFFPL·SMQ·FVTQVNQAYC·GVASI·IMVLNSLGINAPETAQ    91
3k8ua           HYK······LVPQIDTRDCGPAVLA·S·VAKHYGSN·YS···    30
manual       MTGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDP··RRT    71
geno3d       ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSID·PRRTW    64
hhpred       ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSID·PRRTW    66
itasser1     ·TGFFKL·M·EQFNTQDEPAFCGLASLA·M·TLNALSID·PRRTW    72
itasser2     ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDPRRTWK    73
itasser3     ·TGFFKLME·Q·FNTQDEPAFC·GLASLAM·TLNALSID·PRRTW    72
itasser4     ·TGFFKLME·Q·FNTQDEPAFC·GLASLAM·TLNALSID·PRRTW    72
itasser5     ·TGFFKL·M·EQFNTQDEPAFC·GLASL·A·MTLNALSIDPRRTW    72
itasser6     ·TGFFKL·MEQ·FNTQDEPAFC·GLASLAM·TLNALSID·PRRTW    72
itasser7     ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDPRRTWK    73
itasser8     ·TGFFKL·MEQ·FNTQDEPAFC·GLASLAM·TLNALSID·PRRTW    72
itasser9     MTGFFKLME·Q·FNTQDEPAFC·GLASLAM·TLNALSIDPRRTWK    73
itasser10    ·TGFFKL·M·EQFNTQDEPAFCGLASLA·M·TLNALSID·PRRTW    72
loopp        ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDPRRTWK    66
loopp2       ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDPRRTWK    66
loopp5       ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDPRRTWK    66
multicom     ·TGFFKLME·Q·FNTQDEPAFC·GLASLAM·TLNALSIDPRRTWK    73
phyre2int    ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDP·····    68
phyre2nrm    ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDP·····    68
phyre2o2o    ·TGFFKL·MEQ·FNTQDEPAFC·GLASLAM·TLNALSIDP·····    68
raptorx      ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSID·PRRTW    69
swissmodel   ·TGFFKL·MEQ·FNTQDEPAFC·GLASL·AMTLNALSIDP·····    62

2bu3a        YSPYRVFTQDNFFSN·EKTK····AVIA·PEVVAR·QGMTLDELG   129
2btwb        YSPYRVFTQDNFFSN·EKTK····AVIA·PEVVAR·QGMTLDELG   129
3k8ua        ···········IAYL···········RE·LSKTNKQGT·TALGIV    51
manual       WK·····GSWRWFHE·AMLD····CCRP·LDAVKE·EGITLYQAS   104
geno3d       KGSWRWFHE·······AMLD····CCRP·LDAVKE·EGITLYQAS    96
hhpred       KGSWRWFH·EAMLD··········CCRP·LDAVKE·EGITLYQAS    98
itasser1     KGSWRWFHEAMLDCC···········RP·LDAVKEEGI·TLYQAS   104
itasser2     GS·WRWFH·EAMLD··········CCRP·LDAVKE·EGITLYQAS   104
itasser3     KGSWRWFHEAMLDCC·RPLD····AVKE·E······GI·TLYQAS   104
itasser4     KGSWRWFHEAMLDCC·RPLD··········AVKEE·GI·TLYQAS   104
itasser5     KGSWRWFHEAMLDCC·R·········PL·DAVKEE·GI·TLYQAS   104
itasser6     KGSWRWFHEAMLDCC·RPLD·······A···VKEE·GI·TLYQAS   104
itasser7     GS·WRWFH·EAMLD··········CCRP·LDAVKE·EGITLYQAS   104
itasser8     KGSWRWFHEAMLDCC·RPLD····AVKE·E······GI·TLYQAS   104
itasser9     G·SWRWFHEAMLDCC·RPLD····AVK·······E·EGITLYQAS   104
itasser10    KGSWRWFHEAMLDCC···········RP·LDAVKEEGI·TLYQAS   104
loopp        GS·WRWF·······H·EAML····DCCRPLDAVKE·EGITLYQAS    97
loopp2       GS·WRWFH········EAML····DCCRPLDAVKE·EGITLYQAS    97
loopp5       GS·WRWF·······H·EAML····DCCRPLDAVKE·EGITLYQAS    97
multicom     GS·WRWFHEAMLDCC····R····PL···DAVKEE·GI·TLYQAS   104
phyre2int    ···RRTWKGSWRWF····HEAMLDCCRP·LDAVKE·EGITLYQAS   104
phyre2nrm    ···RRTWKGSWRWF····HEAMLDCCRP·LDAVKE·EGITLYQAS   104
phyre2o2o    ···RRTWKGSWRWF····HEAMLDCCRP·LDAVKE·EGITLYQAS   104
raptorx      KGSWRWFHEAMLDCCR···········P·LDAVKE·EGITLYQAS   101
swissmodel   ···RRTWKGSWRWF····HEAMLDCCRP·LDAVKE·EGITLYQAS    98
```

```
2bu3a        RLIASYGVKVK··V··N··H··A··SDT··N··IEDFRKQVAENL   160
2btwb        RLIASYGVKVK··V··N··H··AS··DT··N··IEDFRKQVAENL   160
3k8ua        EAAKKLGFETR··S··I··KADMT··LF··D··YNDL········    76
manual       CLARCNGARVE··L··V··P··YG··SA··GLSLERFRREVEAVC   137
geno3d       CLARCNGARVE··LVPY··G··SA··GL··S··LERFRREVEAVC   129
hhpred       CLARCNGARVE··L··V··P··Y··GSAGLS··LERFRREVEAVC   131
itasser1     CLARCNGARVE··L··V··PYGSA··GL··S··LERFRREVEAVC   137
itasser2     CLARCNGARVE··L··VPYG··S··AGL··S··LERFRREVEAVC   137
itasser3     CLARCNGARVE··L··V··PYGSA··GL··S··LERFRREVEAVC   137
itasser4     CLARCNGARVE··L··V··PYGSA··GL··S··LERFRREVEAVC   137
itasser5     CLARCNGARVE··L··V··PYGSA··GL··S··LERFRREVEAVC   137
itasser6     CLARCNGARVE··L··V··PYGSA··GL··S··LERFRREVEAVC   137
itasser7     CLARCNGARVE··L··V··PYGS··AGL··S··LERFRREVEAVC   137
itasser8     CLARCNGARVE··L··V··PYGSA··GL··S··LERFRREVEAVC   137
itasser9     CLARCNGARVE··L··V··PYGSA··GL··S··LERFRREVEAVC   137
itasser10    CLARCNGARVE··L··V··PYGSA··GL··S··LERFRREVEAVC   137
loopp        CLARCNGARVE··L··V··P··Y··GSAGLS··LERFRREVEAVC   130
loopp2       CLARCNGARVE··L··V··P··Y··GSAGLS··LERFRREVEAVC   130
loopp5       CLARCNGARVE··L··V··P··YG··SAGLS··LERFRREVEAVC   130
multicom     CLARCNGARVELVP··Y··G··S··AGL··S··LERFRREVEAVC   137
phyre2int    CLARCNGARVE··L··V··P··YGSAGL··S··LERFRREVEAVC   137
phyre2nrm    CLARCNGARVE··L··V··P··YGSAGL··S··LERFRREVEAVC   137
phyre2o2o    CLARCNGARVE··L··V··P··YGSAGL··S··LERFRREVEAVC   137
raptorx      CLARCNGARVE··L··V··P··YGSAGL··S··LERFRREVEAVC   134
swissmodel   CLARCNGARVE··LVPY··G··S··AGL··S··LERFRREVEAVC   131

2bu3a        KQDGNFVIVNYLRKEIGQERGGHISPLAAYNEQTDRFLIMD·V·S   203
2btwb        KQDGNFVIVNYLRKEIGQERGGHISPLAAYNEQTDRFLIMD·V·S   203
3k8ua        ···TYPFIVHVIK····GKRLQHYYVVYGSQ··NNQLII·G·DPD   110
manual       GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
geno3d       GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   172
hhpred       GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD V A   174
itasser1     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLV·L·DVA   180
itasser2     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
itasser3     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
itasser4     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
itasser5     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLV·LDV·A   180
itasser6     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
itasser7     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
itasser8     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
itasser9     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
itasser10    GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLV·L·DVA   180
loopp        GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   173
loopp2       GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   173
loopp5       GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   173
multicom     GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
phyre2int    GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
phyre2nrm    GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
phyre2o2o    GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   180
raptorx      GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   177
swissmodel   GSGEEHIVVSYSRKAFLQTGDGHFSPIGGYHRGRDLVLVLD·V·A   174
```

```
2bu3a        RYKYPPVWVKTTDLWKAMN·TVDSVSQKTRGFVF·VS········    238
2btwb        RYKYPPVWVKTTDLWKAMN·TVDSVSQKTRGFVF·VS            238
3k8ua        PSVK·VTRMSKERFQSE·WT··········GLAI·FLAPQ·····    137
manual       RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR·LASNPLLDSV    223
geno3d       RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR·LA········    207
hhpred       RFKYPPHWVPLPMLYHGMS YVDKVTGRPRGYMRLAS            210
itasser1     RFKYPPHWVPLPMLYHG·MSYVDKVTGRPRGYMR·LASNPLLDSV    223
itasser2     RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR LASNPLLDSV    223
itasser3     RFKYPPHWVPLPMLYHG·MSYVDKVTGRPRGYMR LASNPLLDSV    223
itasser4     RFKYPPHWVPLPMLYHG·MSYVDKVTGRPRGYMR LASNPLLDSV    223
itasser5     RFKYPPHWVPLPMLYHG·MSYVDKVTGRPRGYMR·LASNPLLDSV    223
itasser6     RFKYPPHWVPLPMLYHG·MSYVDKVTGRPRGYMR·LASNPLLDSV    223
itasser7     RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR·LASNPLLDSV    223
itasser8     RFKYPPHWVPLPMLYHG·MSYVDKVTGRPRGYMR·LASNPLLDSV    223
itasser9     RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR·LASNPLLDSV    223
itasser10    RFKYPPHWVPLPMLYHG·MSYVDKVTGRPRGYMR·LASNPLLDSV    223
loopp        RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR·LA········    208
loopp2       RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR·LA········    208
loopp5       RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR·LA········    208
multicom     RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMRLASNPLLDSVL    224
phyre2int    RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR LASNPLLDSV    223
phyre2nrm    RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR LASNPLLDSV    223
phyre2o2o    RFKYPPHWVPLPMLYHG·MSYVDKVTGRPRGYMR·LASNPLLDSV    223
raptorx      RFKYPPHWVPLPMLYHGMS·YVDKVTGRPRGYMR·LASNPLL···    217
swissmodel   RFKYPPHWVPLPMLY······························    189

2bu3a        ···························    238
2btwb                                       238
3k8ua        ···························    137
manual       LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
geno3d       ···························    207
hhpred                                      210
itasser1     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser2     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser3     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser4     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser5     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser6     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser7     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser8     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser9     LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
itasser10    LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
loopp        ···························    208
loopp2       ···························    208
loopp5       ···························    208
multicom     LTCDVRSAPEDWRPAEAFVRSGAAAL     250
phyre2int    LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
phyre2nrm    LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
phyre2o2o    LLTCDVRSAPEDWRPAEAFVRSGAAAL    250
raptorx      ···························    217
swissmodel   ···························    189
```

A.4. Dope, Selectpro, and Qmean6 Z scores of All Models

Model	Dope	Selectpro	Qmean6 Z-score
Manual-1	−1.52	0.658	−1.196
Manual-2	−1.49	0.657	−1.186
Manual-3	−1.41	0.626	−0.397
Manual-4	−1.36	0.655	−1.086
Geno3D-1	−0.95	0.587	−2.347
Geno3D-2	−0.96	0.574	−2.116
Geno3D-3	−0.98	0.577	−1.413
Geno3D-4	−0.99	0.618	−1.792
Geno3D-5	−0.96	0.576	−2.348
HHpred-myaln	−0.20	0.282	−1.772
HHpred-theiraln	0.67	0.291	−2.105
I-Tasser-1	−0.34	0.595	−2.742
I-Tasser-2	−0.43	0.595	−2.501
I-Tasser-3	−0.54	0.570	−2.540
I-Tasser-4	−0.42	0.602	−2.263
I-Tasser-5	−0.36	0.576	−2.523
Loopp-1	−1.20	0.644	−0.654
Loopp-2	−1.25	0.645	−0.797
Loopp-3	2.20	0.208	−5.045
Multicom	0.02	0.483	−2.032
Phyre^2-int	−0.42	0.542	−1.988
Phyre^2-nor	0.28	0.589	−1.361
Phyre^2-o2o	−0.51	0.631	−1.102
RaptorX-1	−1.17	0.641	−1.012
RaptorX-2	1.32	0.044	−2.766
RaptorX-3	1.15	0.004	−3.608
RaptorX-4	1.14	0.273	−3.660
RaptorX-5	0.81	0.368	−2.942
SM-auto	0.37	0.489	−4.007
SM-myaln	−0.02	0.548	−1.275

SM-theiraln	−0.99	0.591	−1.015

A.5. Distances Between ϵ-Nitrogens of Solvent-Exposed Lysine Residues

This section lists the distances of the ϵ-nitrogens of solvent-exposed lysine residues for all simulations as mean ± standard deviation. All values are in Å.

A.5.1. Monomers

Table A.2. Lysine distances in monomer Gromacs explicit solvent MD

Residue	38	73	94	151	183	203
38		17.5 ± 3.1	27.8 ± 1.6	36.1 ± 1.3	29.6 ± 1.5	41.1 ± 2.6
73	17.5 ± 3.1		17.0 ± 2.6	36.0 ± 2.0	27.4 ± 3.3	41.6 ± 4.2
94	27.8 ± 1.6	17.0 ± 2.6		24.1 ± 2.2	19.0 ± 2.8	29.8 ± 4.5
151	36.1 ± 1.3	36.0 ± 2.0	24.1 ± 2.2		19.6 ± 3.4	10.4 ± 2.6
183	29.6 ± 1.5	27.4 ± 3.3	19.0 ± 2.8	19.6 ± 3.4		28.3 ± 5.2
203	41.1 ± 2.6	41.6 ± 4.2	29.8 ± 4.5	10.4 ± 2.6	28.3 ± 5.2	

Table A.3. Lysine distances in monomer Amber explicit solvent MD

Residue	38	73	94	151	183	203
38		15.5 ± 2.9	31.0 ± 1.8	37.6 ± 1.2	31.7 ± 2.1	36.9 ± 3.1
73	15.5 ± 2.9		19.0 ± 1.5	32.9 ± 1.7	27.1 ± 3.1	28.7 ± 3.4
94	31.0 ± 1.8	19.0 ± 1.5		24.1 ± 2.1	17.8 ± 2.8	17.6 ± 4.5
151	37.6 ± 1.2	32.9 ± 1.7	24.1 ± 2.1		16.1 ± 3.1	10.5 ± 3.0
183	31.7 ± 2.1	27.1 ± 3.1	17.8 ± 2.8	16.1 ± 3.1		17.9 ± 3.0
203	36.9 ± 3.1	28.7 ± 3.4	17.6 ± 4.5	10.5 ± 3.0	17.9 ± 3.0	

Table A.4. Lysine distances in monomer Amber implicit solvent MD

Residue	38	73	94	151	183	203
38		21.8 ± 3.3	32.8 ± 2.1	38.7 ± 1.4	31.1 ± 1.6	42.2 ± 3.9
73	21.8 ± 3.3		23.3 ± 2.2	39.6 ± 2.4	29.7 ± 2.1	39.4 ± 4.5
94	32.8 ± 2.1	23.3 ± 2.2		22.8 ± 3.2	13.9 ± 3.9	22.2 ± 4.8
151	38.7 ± 1.4	39.6 ± 2.4	22.8 ± 3.2		18.9 ± 2.5	10.2 ± 2.8
183	31.1 ± 1.6	29.7 ± 2.1	13.9 ± 3.9	18.9 ± 2.5		23.2 ± 5.5
203	42.2 ± 3.9	39.4 ± 4.5	22.2 ± 4.8	10.2 ± 2.8	23.2 ± 5.5	

A.5.2. Dimers

Table A.5. Lysine distances in dimer Gromacs explicit solvent MD

Residue	38	73	94	151	183	203	288	323	344	401	433	453
38		8.6 ± 1.4	26.1 ± 1.8	36.2 ± 1.5	30.6 ± 1.1	43.7 ± 2.3	27.3 ± 1.1	25.5 ± 1.6	29.7 ± 4.3	26.1 ± 2.3	14.8 ± 1.6	36.0 ± 4.5
73	8.6 ± 1.4		20.8 ± 1.6	36.7 ± 1.7	31.0 ± 1.8	43.5 ± 2.5	29.9 ± 1.4	24.0 ± 1.4	24.3 ± 3.9	23.7 ± 2.8	8.9 ± 1.6	33.4 ± 5.7
94	26.1 ± 1.8	20.8 ± 1.6		27.1 ± 4.0	21.2 ± 3.3	32.7 ± 3.1	28.9 ± 2.9	15.7 ± 2.7	21.8 ± 2.6	35.9 ± 2.8	22.6 ± 2.0	42.7 ± 6.7
151	36.2 ± 1.5	36.7 ± 1.7	27.1 ± 4.0		19.6 ± 3.9	12.2 ± 3.0	33.0 ± 2.4	28.7 ± 2.9	45.9 ± 2.9	56.2 ± 2.3	42.8 ± 1.8	64.0 ± 6.5
183	30.6 ± 1.1	31.0 ± 1.8	21.2 ± 3.3	19.6 ± 3.9		29.0 ± 4.8	15.9 ± 2.8	12.7 ± 2.0	32.6 ± 3.6	43.9 ± 2.0	34.0 ± 1.6	50.1 ± 4.6
203	43.7 ± 2.3	43.5 ± 2.5	32.7 ± 3.1	12.2 ± 3.0	29.0 ± 4.8		43.1 ± 3.8	37.5 ± 3.3	53.1 ± 3.5	64.3 ± 3.2	50.0 ± 2.6	72.4 ± 7.0
288	27.3 ± 1.1	29.9 ± 1.4	28.9 ± 2.9	33.0 ± 2.4	15.9 ± 2.8	43.1 ± 3.8		14.5 ± 1.9	30.3 ± 2.2	35.7 ± 1.5	30.8 ± 0.9	41.2 ± 3.2
323	25.5 ± 1.6	24.0 ± 1.4	15.7 ± 2.7	28.7 ± 2.9	12.7 ± 2.0	37.5 ± 3.3	14.5 ± 1.9		20.7 ± 2.6	33.0 ± 1.8	24.4 ± 1.2	39.0 ± 5.4
344	29.7 ± 4.3	24.3 ± 3.9	21.8 ± 2.6	45.9 ± 2.9	32.6 ± 3.6	53.1 ± 3.5	30.3 ± 2.2	20.7 ± 2.6		21.0 ± 4.6	18.0 ± 4.4	24.7 ± 6.7
401	26.1 ± 2.3	23.7 ± 2.8	35.9 ± 2.8	56.2 ± 2.3	43.9 ± 2.0	64.3 ± 3.2	35.7 ± 1.5	33.0 ± 1.8	21.0 ± 4.6		16.1 ± 2.9	12.2 ± 3.7
433	14.8 ± 1.6	8.9 ± 1.6	22.6 ± 2.0	42.8 ± 1.8	34.0 ± 1.6	50.0 ± 2.6	30.8 ± 0.9	24.4 ± 1.2	18.0 ± 4.4	16.1 ± 2.9		25.4 ± 6.0
453	36.0 ± 4.5	33.4 ± 5.7	42.7 ± 6.7	64.0 ± 6.5	50.1 ± 4.6	72.4 ± 7.0	41.2 ± 3.2	39.0 ± 5.4	24.7 ± 6.7	12.2 ± 3.7	25.4 ± 6.0	

Table A.6. Lysine distances in dimer AMBER explicit solvent MD

Residue	38	73	94	151	183	203	288	323	344	401	433	453
38		13.4 ± 2.1	32.8 ± 2.0	36.2 ± 1.5	32.3 ± 1.2	37.7 ± 2.8	28.4 ± 1.4	26.6 ± 1.2	35.2 ± 3.6	30.1 ± 3.1	19.5 ± 2.8	33.2 ± 4.2
73	13.4 ± 2.1		23.0 ± 1.4	34.8 ± 1.6	28.7 ± 1.7	33.5 ± 3.3	29.0 ± 1.2	22.3 ± 1.5	23.2 ± 2.1	25.2 ± 1.7	11.3 ± 1.4	26.1 ± 3.2
94	32.8 ± 2.0	23.0 ± 1.4		23.9 ± 1.8	18.5 ± 2.2	18.4 ± 2.9	34.3 ± 2.5	21.5 ± 2.7	23.1 ± 2.1	42.9 ± 2.3	30.7 ± 1.7	41.0 ± 3.0
151	36.2 ± 1.5	34.8 ± 1.6	23.9 ± 1.8		12.7 ± 1.9	10.8 ± 3.1	33.1 ± 1.5	25.1 ± 2.0	43.1 ± 2.0	57.3 ± 1.7	45.1 ± 1.3	57.1 ± 2.8
183	32.3 ± 1.2	28.7 ± 1.7	18.5 ± 2.2	12.7 ± 1.9		17.1 ± 3.0	23.3 ± 1.3	13.5 ± 1.7	32.9 ± 2.0	48.1 ± 1.6	37.4 ± 1.1	47.3 ± 2.8
203	37.7 ± 2.8	33.5 ± 3.3	18.4 ± 2.9	10.8 ± 3.1	17.1 ± 3.0		38.9 ± 2.9	28.3 ± 2.7	40.4 ± 3.2	56.9 ± 3.3	43.8 ± 3.3	56.2 ± 3.8
288	28.4 ± 1.4	29.0 ± 1.2	34.3 ± 2.5	33.1 ± 1.5	23.3 ± 1.3	38.9 ± 2.9		13.3 ± 1.4	33.6 ± 1.1	37.4 ± 1.0	32.6 ± 0.8	37.8 ± 1.7
323	26.6 ± 1.2	22.3 ± 1.5	21.5 ± 2.7	25.1 ± 2.0	13.5 ± 1.7	28.3 ± 2.7	13.3 ± 1.4		24.8 ± 1.6	36.5 ± 1.5	28.2 ± 1.5	35.7 ± 2.4
344	35.2 ± 3.6	23.2 ± 2.1	23.1 ± 2.1	43.1 ± 2.0	32.9 ± 2.0	40.4 ± 3.2	33.6 ± 1.1	24.8 ± 1.6		26.7 ± 2.2	21.3 ± 2.2	22.6 ± 3.0
401	30.1 ± 3.1	25.2 ± 1.7	42.9 ± 2.3	57.3 ± 1.7	48.1 ± 1.6	56.9 ± 3.3	37.4 ± 1.0	36.5 ± 1.5	26.7 ± 2.2		14.6 ± 1.7	7.1 ± 1.7
433	19.5 ± 2.8	11.3 ± 1.4	30.7 ± 1.7	45.1 ± 1.3	37.4 ± 1.1	43.8 ± 3.3	32.6 ± 0.8	28.2 ± 1.5	21.3 ± 2.2	14.6 ± 1.7		15.9 ± 3.2
453	33.2 ± 4.2	26.1 ± 3.2	41.0 ± 3.0	57.1 ± 2.8	47.3 ± 2.8	56.2 ± 3.8	37.8 ± 1.7	35.7 ± 2.4	22.6 ± 3.0	7.1 ± 1.7	15.9 ± 3.2	

Table A.7. Lysine distances in dimer Amber implicit solvent MD

Residue	38	73	94	151	183	203	288	323	344	401	433	453
38		13.0 ± 1.8	32.1 ± 2.1	37.6 ± 1.4	31.1 ± 1.5	34.6 ± 3.0	27.3 ± 1.6	33.2 ± 1.3	27.9 ± 2.7	26.3 ± 2.4	15.5 ± 2.0	32.6 ± 2.9
73	13.0 ± 1.8		24.8 ± 2.1	38.6 ± 1.5	30.2 ± 2.1	32.4 ± 3.8	29.2 ± 1.5	29.1 ± 1.8	17.4 ± 2.9	24.8 ± 1.8	8.0 ± 1.7	27.7 ± 3.1
94	32.1 ± 2.1	24.8 ± 2.1		24.9 ± 2.8	16.4 ± 3.0	17.2 ± 3.5	28.7 ± 3.1	17.8 ± 2.7	28.2 ± 3.1	45.3 ± 2.7	29.0 ± 3.4	43.8 ± 3.6
151	37.6 ± 1.4	38.6 ± 1.5	24.9 ± 2.8		14.2 ± 1.9	12.2 ± 2.8	32.0 ± 2.3	29.5 ± 2.6	48.6 ± 2.8	59.2 ± 1.7	43.9 ± 2.5	60.8 ± 3.3
183	31.1 ± 1.5	30.2 ± 2.1	16.4 ± 3.0	14.2 ± 1.9		15.1 ± 3.4	20.6 ± 2.3	16.2 ± 2.4	37.3 ± 2.4	48.6 ± 1.6	34.3 ± 3.3	49.0 ± 2.7
203	34.6 ± 3.0	32.4 ± 3.8	17.2 ± 3.5	12.2 ± 2.8	15.1 ± 3.4		34.0 ± 3.2	28.0 ± 2.6	42.0 ± 3.5	55.2 ± 3.7	38.3 ± 4.7	56.2 ± 4.0
288	27.3 ± 1.6	29.2 ± 1.5	28.7 ± 3.1	32.0 ± 2.3	20.6 ± 2.3	34.0 ± 3.2		15.7 ± 2.2	32.7 ± 1.9	36.6 ± 1.3	29.4 ± 2.1	37.0 ± 2.5
323	33.2 ± 1.3	29.1 ± 1.8	17.8 ± 2.7	29.5 ± 2.6	16.2 ± 2.4	28.0 ± 2.6	15.7 ± 2.2		28.6 ± 1.8	41.5 ± 1.4	30.3 ± 2.8	38.9 ± 2.3
344	27.9 ± 2.7	17.4 ± 2.9	28.2 ± 3.1	48.6 ± 2.8	37.3 ± 2.4	42.0 ± 3.5	32.7 ± 1.9	28.6 ± 1.8		22.3 ± 2.8	13.7 ± 3.0	18.1 ± 3.0
401	26.3 ± 2.4	24.8 ± 1.8	45.3 ± 2.7	59.2 ± 1.7	48.6 ± 1.6	55.2 ± 3.7	36.6 ± 1.3	41.5 ± 1.4	22.3 ± 2.8		18.4 ± 2.3	11.1 ± 2.6
433	15.5 ± 2.0	8.0 ± 1.7	29.0 ± 3.4	43.9 ± 2.5	34.3 ± 3.3	38.3 ± 4.7	29.4 ± 2.1	30.3 ± 2.8	13.7 ± 3.0	18.4 ± 2.3		21.1 ± 3.5
453	32.6 ± 2.9	27.7 ± 3.1	43.8 ± 3.6	60.8 ± 3.3	49.0 ± 2.7	56.2 ± 4.0	37.0 ± 2.5	38.9 ± 2.3	18.1 ± 3.0	11.1 ± 2.6	21.1 ± 3.5	

Statement of Authorship

I declare on oath that I completed this work on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this, nor a similar work, has been published or presented to an examination committee.

Halle, October 27, 2013

Acknowledgements

> “This thought has been buried in my head for the past seven years, and finally refused to be silenced about a year ago. Since then, the entire foundation I had up to that point built my life around has been crumbling. It’s been a struggle to push through and try and come to terms with these thoughts, but one thing … has been sustaining me: People.”
>
> —*CualEsMiNombre*

First and foremost, I wouldn’t have managed this feat without my family, who carried me through all this despite the fact that I was not always (or, in fact, at all) easy on them.

I would also like to thank PD Dr. Thondorf and Dr. Dobritzsch for allowing me to work on a topic that was—at the beginning—a bit mystifying and—at the end—still fascinating. Their laid-back mentorship allowed me to pursue my own interests within the limits of the topic, an opportunity which I exploited shamelessly.

The girls in the Molecular Modeling group always provided snacks, great advice, help with the sometimes overwhelmingly large suites of programs and (last but not least) coffee. I probably am eternally indebted to them, for I can naught function at all without coffee.

I’d like to thank my editors, Sarah and Katharina as well as Lucas. The way I picture it, they could probably only shake their heads while pouring over the mess that I produced here. Thanks, guys, much appreciated—if I can return the favor, just say so.

I’d also like to thank Thomas Zimmermann, who, when computing resources were on the verge of growing too small, provided our group with additional servers and allowed me to administrate them. This was a valuable experience, and I cannot thank him enough for that.

Last but not least, I’d like to thank the people that got my back whenever I fell down and urged me to get back up again. I won’t name you, because you know who you are. Thanks a lot, guys, I’m looking forward to DnD, a beer at the pub, and what have you.