# An Implicit Solvent Model for Biomolecular Monte Carlo Simulations

Zur Erlangung des akademisches Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

von der Fakultät für Physik des

Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Dipl. Phys. Martin Brieg

aus Weimar

Tag der mündlichen Prüfung: 30. Mai 2014

Referent: Prof. Dr. Wolfgang Wenzel

Korreferent: Prof. Dr. Gerd Schön

# *Contents*

# 1 Introduction and Overview

Evolution has come a long way since the development of the first life forms on earth. It has created a vast diversity of living beings, ranging from simple bacteria to complex multicellular organisms such as plants, insects, animals, or even intelligent humans. Despite the huge variety of life forms, they are developed with and do function based on the same toolkit. This toolkit is the genetic code stored in the DNA. The encoded genes are blueprints for proteins, which are assembled inside living cells according to them.

Proteins are a crucial class of biomolecules. They participate in fulfilling or regulating nearly all tasks that are necessary for a cell to function and survive. These tasks include the structural stability of the cell, regulation of cell fusion or division, cargo transport within the cell and to other cells, catalysis of chemical reactions, energy conversion, metabolism, signal transmission, or the expression of genes to build proteins.[1–3] Over the millions of years, evolution has produced ever-new genes and, therefore, proteins that can provide increased chances of survival and reproduction. In combination with natural selection,[4] evolution yielded the diversity of life we observe today.

Intrigued by the possibilities that this toolkit provides, scientists have tried to understand in detail how it works. One of the first achievements was the proposal of the DNA double helical structure by Watson and Crick in 1953[5] along with first experimental evidence from X-ray crystallography.[6,7] Together with the proposal of Gamow that three base pairs of the DNA encode one amino acid,[8] the mechanism how to translate the sequence of base pairs in the DNA into a sequence of amino acids was unraveled based on experiments of Nirenberg and Matthaei.[9,10] Subsequent assembly of the amino acids of a given sequence into a chain forms the encoded protein. However, unraveling the genetic code yielded more and more questions, as it became obvious that there was no simple relation between a protein's amino acid sequence and its function.[3] Even today, the prediction of an unknown protein's function from its amino acid sequence remains a major challenge.[11]

After solving the first three-dimensional structures of proteins, it became apparent that these structures were the missing links between functions and amino acid sequences. The three-dimensional structure, referred to as the protein fold or conformation, arranges particular atoms in just the right way to form the functional groups that allow proteins to fulfill their various functions.[12] Since then, the determination of protein structures has been one of the cornerstones of structural biology and biomolecular research. Crystallizing proteins and solving their structure by

X-ray diffraction has thus become an essential tool for molecular biologists, as is evidenced by the 85,000 protein structures deposited in the Protein Data Bank (PDB) that have been solved using this method. These are 88.4% of all protein structures currently available in the PDB.[13,14] Another technique that allows insights into the structure of proteins is nuclear magnetic resonance (NMR) spectroscopy, which is the main contributor of the remaining 12.6% of protein structures in the PDB.

Although X-ray diffraction provides only a single snapshot of a crystallized protein's structure, they are by no means rigid. Especially NMR spectroscopy provides insights into the dynamical aspects of protein structures.[1,15–17] Therefore, the fold of a protein refers to an ensemble of structures that share common topological features, but may differ, for example, in the packing of certain amino acids or the arrangement of other structural elements. Driven by the development of new experimental techniques and improved computer simulations, more and more data has been gathered in the last decades that stress the essential role of these dynamical to the function of proteins.[15–18]

Due to persisting limitations of experimental techniques, computational methods are used widely by many biomolecular scientists nowadays. Using molecular forcefield models, molecular dynamics simulations[19,20] can in principle generate time-resolved trajectories of the structural ensembles and atomistic mechanisms that underlie a protein's function.[21] This has promoted computer simulations to one of the standard tools for a molecular biologist. This fact was recently recognized by awarding the 2013 Nobel Prize in chemistry to Martin Karplus, Michael Levitt and Arieh Warshel for their groundbreaking work in combining macroscopic, classical and quantum mechanical methods in the 1970s.[22] Warshel and Levitt used these methods to study the behavior of a catalytic site in a protein.[23]

However, this award also marks the greatest problem of computational biomolecular research: it is imperative for biomolecular simulations to use or combine methods that are as accurate as needed, but at the same time as fast as possible. Treating biomolecules and their environment by quantum mechanics on timescales relevant for biological processes is excessively demanding on the computational side for current state-of-the-art supercomputers. Even with classical molecular models, simulations typically reach only the low microsecond timescale with a reasonable amount of invested computation time, wherefore they are unable to elucidate many biologically relevant processes.[21,24–26]

One reason for the large computational cost is the incorporation of the physiological environment into the simulation. Usually this environment is an aqueous solution in which the biomolecule is embedded. The most straightforward method for the inclusion of this environment is to represent every solvent atom explicitly. For a typical biomolecular simulation using an explicit solvent representation, the number of solvent atoms may be much larger than the number of atoms in the biomolecule. Since every solvent atom interacts with every other atom, the number of interactions that must be computed increases quadratically with the number of atoms in the system. Due to this fact, representing the solvent explicitly will become computationally extremely expensive when increasing the size of the investigated system.[27,28]

A vast number of algorithms have been developed to reduce the computational cost for the computation of the interactions in such a system.[29,30] Moreover, computer scientists have designed and built customized hardware for these simulations. Specialized supercomputers based on this hardware could shift the timescale limit of biomolecular molecular dynamics simulations to the low millisecond range.[31–34] However, only one such machine is publicly available, and only for U.S. scientists, which is insufficient to fulfill the high demand of the scientific community.

Enhanced simulation techniques, such as adaptive biasing potentials, can also alleviate the accessible timescale problem to some extent. However, these methods require well-defined reaction coordinates or paths.[35] These may not be available for the process to be studied, or may be challenging to derive beforehand. In conclusion, there is still demand for computational methods that allow the investigation of the structural ensembles and atomistic processes relevant to the function of proteins despite decades of development.

Since the long timescales on which biologically relevant processes take place are the main issue of molecular dynamics, dropping the requirement of having time resolved trajectories of the processes will immediately remove the main issue. Instead, it is sufficient for many studies to have a thermodynamically representative ensemble of structures for a given process. This representative ensemble can be generated using Monte Carlo algorithms. However, this strategy is used only infrequently in computational biomolecular research. One of the reasons is the lack of an adequate simulation package that can use common molecular forcefields.[36]

This is the point where my work sets in. The development and implementation of computational methods for the simulation of biomolecular systems is one cornerstone of computational biophysics that I address in this thesis. I will explain several methods that I have developed and implemented into the SIMONA[37] Monte Carlo simulation framework. These methods enable Monte Carlo simulations of biomolecular systems with common molecular forcefields.

A large challenge for Monte Carlo simulations of biomolecular systems is the inclusion of the solvent as the physiological environment. In these simulations, the conformation of the biomolecule is subject to random perturbations. These perturbations will ultimately lead to overlaps between atoms of the biomolecule and explicit solvent atoms. Such configurations of the system are highly unfavorable and not representative. The proposal of too many non-representative configurations decreases the efficiency and thus the success of Monte Carlo simulations significantly.[36] In this thesis, I describe an implicit solvent model that I have developed and implemented to overcome this challenge. In general, implicit solvent models account for the averaged effects of the solvent onto the biomolecules without requiring an explicit representation of the solvent atoms. Thus, these models are well suited for Monte Carlo simulations.

Due to the low popularity of Monte Carlo simulations, previous implicit solvent models focused on the requirements of molecular dynamics simulations, which can differ substantially from those of Monte Carlo simulations. Consequently, I have designed a new implicit solvent model to fulfill the requirements of Monte Carlo simulations instead. In addition, I investigated how to improve the approximate description of solvent effects by implicit solvent models further and started to extend my implicit solvent model to account for the presence of biological membranes. They represent another important physiological environment for proteins.

To demonstrate the validity and success of the Monte Carlo methods, I will examine the folding of a small protein FSD-EY. A comparison of the protein's folded state in the simulation with that determined by NMR spectroscopy will grant insights into the accuracy of my simulation method and implicit solvent model. In addition, molecular dynamics data from a specialized supercomputer serves as a second reference for validating my methods. Finally, I will try to deduce the folding mechanism of this small protein from my simulation data.

My thesis is structured as follows. In the second chapter, I will provide an introduction into several topics necessary to understand the work I present in this thesis. These topics include the composition of proteins and biological membranes, their general structural features and properties. The chapter also explains how classical molecular forcefields model the interactions within biomolecules such as proteins. It also describes how implicit solvent models include the interactions between biomolecules and their environment, and introduces commonly used molecular surface definitions that are used in these models. Furthermore, it outlines the basics of Monte Carlo simulations.

The third chapter focuses on the methods developed and implemented by me to enable simulations of proteins with common biomolecular forcefields with the SIMONA Monte Carlo simulation framework. At first, I describe the details of the implementation of the AMBER99SB*-ILDN[38–41] biomolecular forcefield terms. I have paid special attention to the different requirements of Monte Carlo simulations in comparison to molecular dynamics for this implementation. The next two sections focus on the implicit solvent model. I explain efficient methods to compute solvent accessible surface area and the Born radii of the generalized Born implicit solvent model. These two methods form the basis of my implicit solvent model. In the last section of this chapter, I present an overview of the achievable simulation performance of SIMONA with the methods implemented by me.

Since implicit solvent models provide only an approximate description of the average solvent effects, the assessment of their accuracy is important for judging the errors that result from their application, as well as determining possible simulation artifacts due to deficiencies of the implicit solvent model. In chapter four, I will describe my contributions to such an assessment that I have performed in cooperation with others. Furthermore, I will present our main conclusions that resulted from this assessment.

Biological membranes are another important physiological environment for proteins, wherefore I have extended my implicit solvent model to account for some basic properties of them. In chapter five, I will first introduce the basic idea of this extension called SLIM and then give details on its implementation. Subsequently, I will review the achievements of the SLIM model, which demonstrate its improved accuracy over prior implicit membrane models and its ability to reproduce established properties of small membrane proteins. Finally, I will present a parallelization strategy for the SLIM model to increase its computational efficiency and provide an overview the resulting Monte Carlo simulation performance.

In chapter six, I will present results on the investigation of the folding of the small protein FSD-EY.[42] As an introduction, I will shortly review the problems of investigating protein folding via computer simulations. Afterwards, I will outline my Monte Carlo simulation setup. Subsequently, I will provide some performance characteristics of my employed Monte Carlo algorithm. Next, I will identify the folded state of FSD-EY in the simulation and compare it to the experimentally determined folded state. In the next step, I will determine FSD-EY's critical folding temperature and try to deduce its folding mechanism.

My thesis closes with a summary of the main results described in this work and a discussion of their implications, as well as with ideas how to continue this work in the future.

# 2 Basic Concepts and Theory

In this chapter, I will introduce several topics that are essential to understanding the systems I investigate or the methods I employ. The first section focuses on the composition, structure, and properties of proteins and biological membranes. The second section outlines the methods with which biomolecular forcefields model intra- and intermolecular interactions. The third section introduces three definitions of molecular surfaces that are used in implicit solvent models. The fourth section reviews the basic theory of implicit solvent models, the physical properties of water, and a common approximate approach. The last section outlines the goal of Monte Carlo simulations, how to carry out such simulations and an extension to increase the efficiency of such simulations for complex systems.

## 2.1 Proteins and Biological Membranes

### Amino Acids

The basic constituents of proteins are amino acids. There are 20 proteinogenic amino acids that can be encoded by the in genes. An amino acid consists of a backbone and a side chain. The backbone is common to all amino acids. It consists of an amine group, an alkyl group, and a carboxyl group (Figure 2.1). The carbon atom of the alkyl group is referred to as the C-alpha atom commonly.

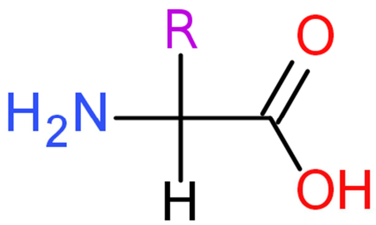


***Figure 2.1. The chemical structure of an amino acid. The backbone consists of the amine group (blue), the alkyl group (black), and the carboxylic acid (red). The side chain R (purple) is bound to the carbon atom of the alkyl group. This atom is called the C-alpha atom. The carbon atoms of the alkyl group and the carboxylic acid are not shown explicitly.***

The amino acids differ in their side chain, which is bound to the C-alpha atom (Figure 2.1). The side chains can contain different chemical groups. As a result, the amino acids have different physical and chemical properties. Key properties for the categorization of the different amino acids are the charge state in solution, the polarity, or the hydrophobicity. The latter is a measure of the solubility of an amino acid in water. Based on these three properties, different categories of amino acids exist. According to Branden and Tooze, these are apolar, polar, positively or negatively charged, and special cases.[43]

Amino acids can react with each other to form peptide bonds. The carboxyl group of one amino acid reacts with the amine group of another amino acid to form a peptide bond under the separation of a water molecule, as illustrated in Figure 2.2. The peptide bond has a partial double bond character. Therefore, rotations around this bond are energetically disfavored, resulting in a planar bond geometry. Since each amino acid contains an amine group and a carboxyl group, it can form up to two peptide bonds. The residual parts of the amino acids after the peptide bond formation are the amino acid residues. Throughout this thesis, I will use the shorthand term residue to refer to amino acid residues. Figure 2.3 shows a ball-and-stick representation of all 20 proteinogenic amino acid residues together with their categorization, one-letter, and three-letter abbreviations.

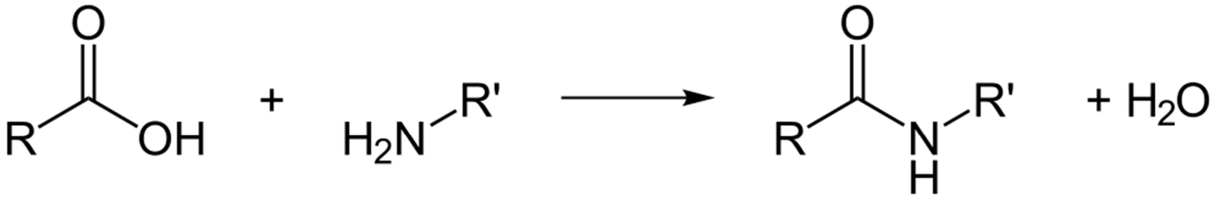


***Figure 2.2. Sketch of the chemical reaction to form a peptide bond between a carboxyl group and an amine group. R and R' label the residual parts of the corresponding amino acids. These parts are referred to as residues commonly.[44]***

### *Protein Primary Structure*

Proteins consist of chains of residues linked by peptide bonds, wherefore they are also referred to as polypeptides. The primary structure of a protein is the sequence in which the different residues are linked into the chain. More commonly, the primary structure is just called the sequence of the protein. By convention, this sequence starts at the residue whose amine group has no peptide bond, which is the N-terminus. The last residue in a chain has no peptide bond at its carboxyl group, which is called the C-terminus.

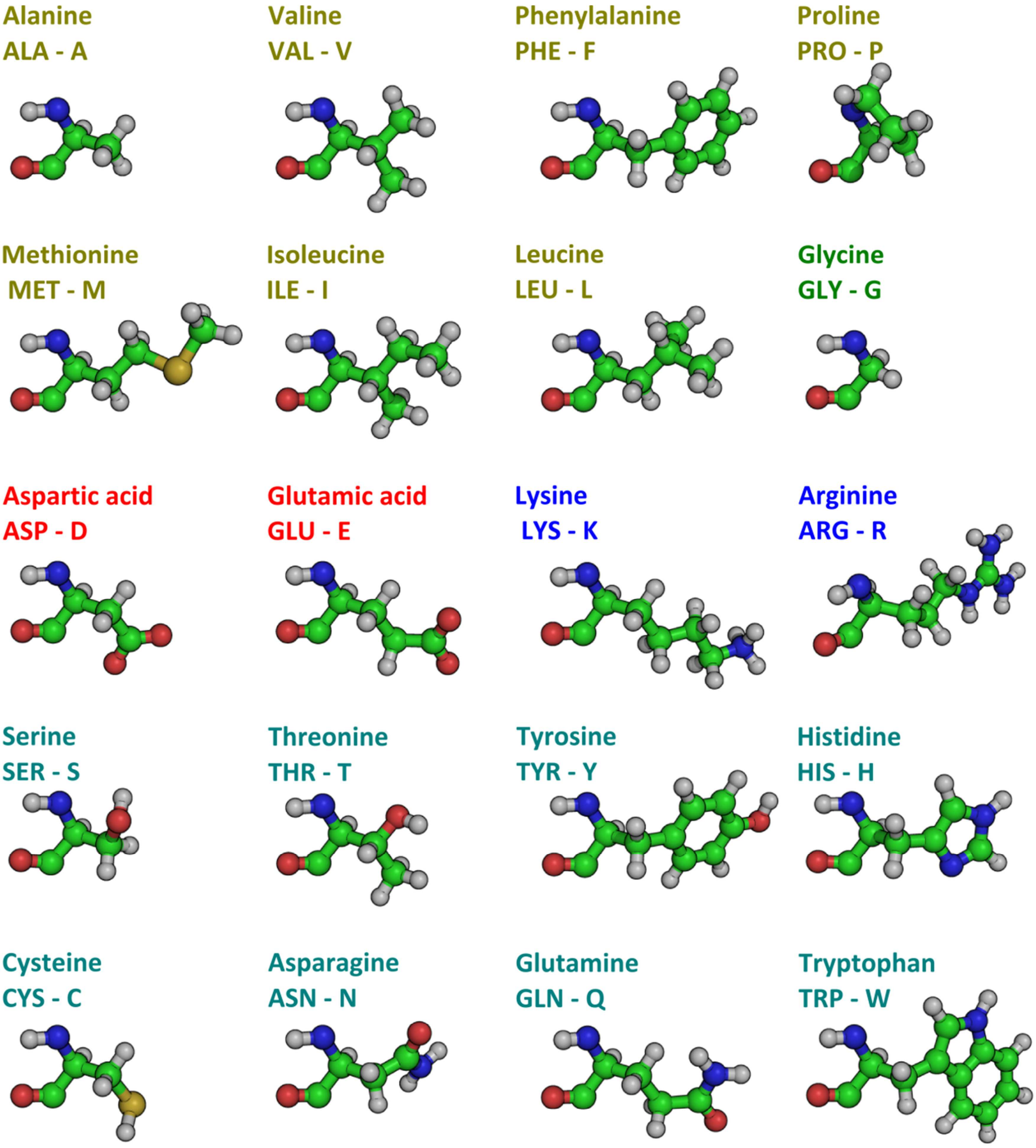


*Figure 2.3. Ball-and-stick representations of all 20 amino acid residues encoded in the genome. Their names, one letter, and three letter abbreviations are also given. The ball color represents the following elements: carbon (green), hydrogen (white), nitrogen (blue), oxygen (red), sulfur (yellow). All amino acid residues are oriented so that their backbone atoms are on the left with the nitrogen atom of the residual amine group at the top left and the oxygen atom of the residual carboxylic acid at the bottom left. The side chains are directed to the right. The amino acids are grouped according to Brandon and Tooze[43] into apolar (yellow label), polar (cyan label), positively charged (blue label), negatively charged (red label) and special cases (green label). Glycine is considered a special case because its side chain consists of only one hydrogen atom. The side chain of Proline is also bound to its residual amine group.*

### *Protein Secondary Structure*

In contrast to the primary structure, the secondary structure describes regular spatial patterns of the atomic positions in a protein. These patterns are the secondary structure elements. Each residue can be part of one such element. Usually one discriminates alpha helices, beta bridges, beta sheets, 3-10 helices, $\pi$-helices, turns, bends, and coil. A common property to discriminate the secondary structure elements is the presence or absence of backbone hydrogen bonding patterns. They form between the residual part of the backbone carboxyl group of one residue and the residual part of the backbone amine group of another residue. According to IUPAC technical report by Arunan et al., hydrogen bonds are an attractive interaction not to be confused with covalent bonds. One hallmark of such a hydrogen bond is the strong directionality of the interaction due to the significant role of electrostatic forces.[45]

Table 2.1 provides a list of the secondary structure elements, together with a short description and their one-letter abbreviations. Figure 2.4 presents visualizations of examples of the secondary structure elements. According to Kabsch and Sander, the two most common secondary structure elements, alpha helices and beta sheets, are cooperative elements. This means that helices are consecutive turns and beta sheets are consecutive beta bridges.[46]

In crystallographic structures, the crystallographers have assigned these elements to the solved X-ray structures based on visual inspection. These data are then available via the Protein Data Bank.[14] However, these assignments are not objective. Proposed pattern recognition algorithms enable an assignment of the secondary structure elements based on objective criteria. These are also suitable for implementations on computers. One of the most common algorithms is the Dictionary of Secondary Structure (DSSP) proposed by Kabsch and Sander.[46] It uses an empirical energy function plus a cutoff criterion to determine the existence of backbone hydrogen bonds. Based on the established hydrogen bonds, the algorithm assigns the secondary structure elements to the residues. In addition, the curvature between the positions of five consecutive C-alpha atoms defines bends. Frishman and Argos proposed a more recent algorithm named STRIDE. This algorithm results in better agreement with the assignments of crystallographers.[47] It also uses backbone dihedral angle information to assign the secondary structure elements to residues.

***Table 2.1. List of commonly used secondary structure elements, their one-letter abbreviation, and the typical characteristic trait of each element. Their definitions are available for example by Kabsch and Sander[46] or Frishman and Argos.[47]***

| Name | Abbr. | Characteristic trait |
|---|---|---|
| **Alpha helix** | H | Consecutive backbone hydrogen bond between residues $i$ and $i + 4$ |
| **Beta bridge** | B | Two backbone hydrogen bonds between two non-overlapping consecutive residue triplets |
| **Beta sheet** | E | A set of consecutive beta bridges |
| **3-10 helix** | G | Consecutive backbone hydrogen bond between residues $i$ and $i + 3$ |
| **$\pi$-Helix** | I | Consecutive backbone hydrogen bond between residues $i$ and $i + 5$ |
| **Turn** | T | Backbone hydrogen bond between two arbitrary residues |
| **Bend** | S | Strong curvature in the backbone chain across five residues |
| **Coil** | C | None of the above |

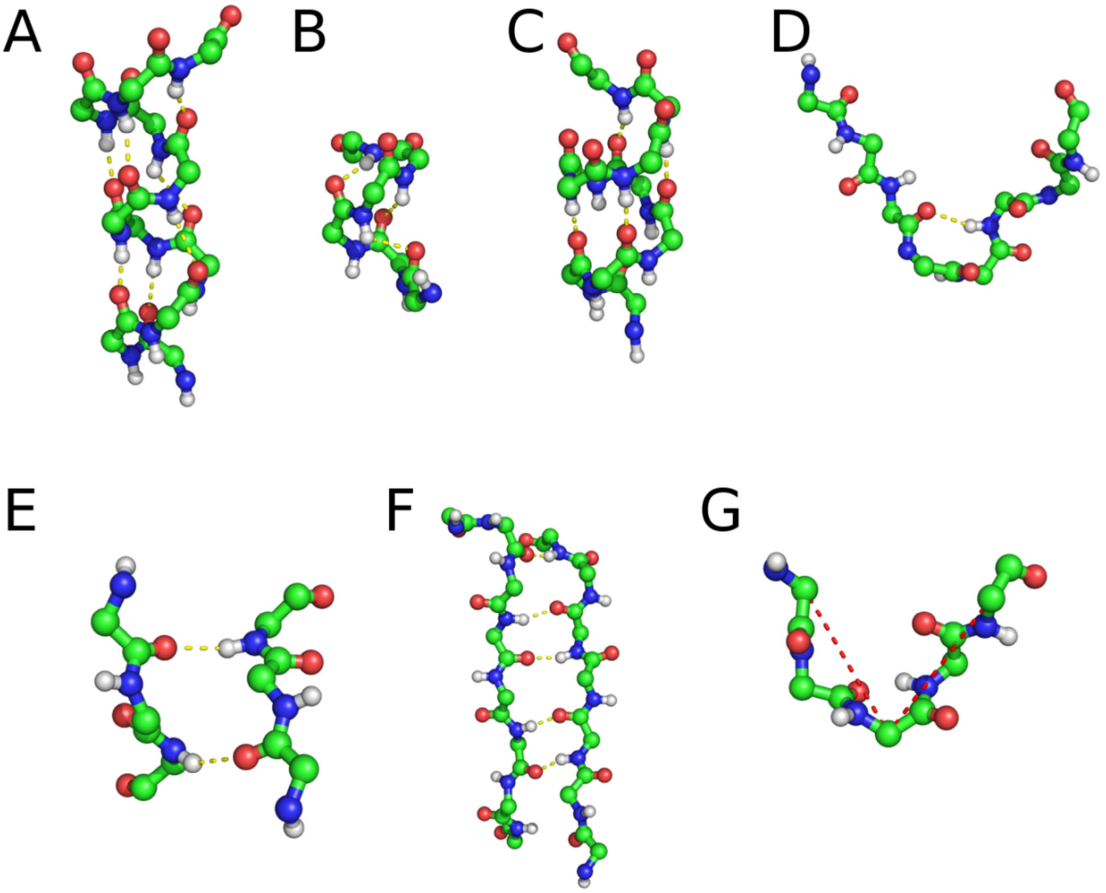


***Figure 2.4. Examples of common secondary structure elements: alpha helix (panel A), 3-10 helix (panel B), π-helix (panel C), turn (panel D), beta bridge (panel E), beta sheet (panel F), bend (panel G). Only backbone atoms of the protein parts are shown. Dashed yellow lines mark hydrogen bonds. The dashed red line in panel G marks the strong bend of the protein backbone.***

Dihedral angles are torsion angles around a given axis. The positions of four atoms define these angles. The first three atoms and last three atoms define a plane. The dihedral angle is the angle between these two planes. The torsion axis is the bond between the second and third atom. For proteins, the two most important dihedral angles are the so-called Φ and Ψ backbone dihedral angles. The former is defined by the backbone atoms $\mathrm{C}^{n-1} - \mathrm{N}^{n} - \mathrm{C}_{\alpha}^{n} - \mathrm{C}^{n}$ and the latter by $\mathrm{N}^{n} - \mathrm{C}_{\alpha}^{n} - \mathrm{C}^{n} - \mathrm{N}^{n+1}$. Here, C is the carbon atom of the residual carboxyl group, N the nitrogen of the residual amine group and $\mathrm{C}_{\alpha}$ the carbon of the backbone alkyl group. The superscript denotes the residue number. Figure 2.5A illustrates both definitions.

The bonds corresponding to the Φ and Ψ dihedral angles have no double bond character. Therefore, they are the main degrees of freedom that determine the three-dimensional structure of a given protein. However, there are some restrictions to these angles, because specific value pairs will lead to atomic overlaps in the polypeptide chain. The Pauli Exclusion Principle energetically disfavors such overlaps. There are also energetically preferred combinations of the Φ and Ψ angles. Those correspond to the most common secondary structure elements, such as alpha helices or beta sheets. For the analysis of a given protein structure, it is common to plot each residue's pair of Φ and Ψ dihedral angles as a scatter plot. This plot is called Ramachandran plot.[48] It gives an overview of the secondary structure content of a protein structure. Figure 2.5B shows an exemplary Ramachandran plot for the protein ubiquitin (PDB code 1UBQ[49]), which contains a mixture of secondary structure elements.

### *Tertiary Structure*

Tertiary protein structure describes the spatial arrangement and packing of the secondary structure elements. To highlight this packing of the secondary structure elements, proteins are visualized in a distinct representation called the cartoon representation. It only provides a trace of the protein backbone, but highlights helices and beta sheets as depicted in Figure 2.6. This representation allows a much easier identification of the most common secondary structure elements and their packing.

A general feature of the tertiary structure in globular proteins not embedded in biological membranes is the burial of hydrophobic residues inside the protein. Thus, these proteins have a hydrophobic core. Polar or charged residues remain at the surface of the protein. These features generate a contribution to the stability of globular protein conformations that alone may already explain their stability.[50] The reason for the hydrophobic core is the so-called hydrophobic effect. It is discussed further in Section 2.4. In contrast, large hydrophobic regions on the outside of proteins allow them to be embedded in biological membranes.

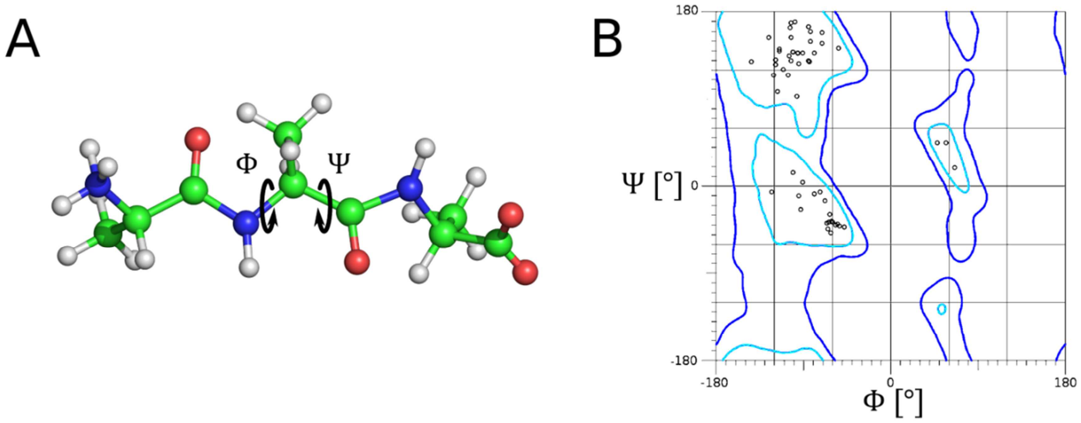


*Figure 2.5. Panel A shows the Φ and Ψ dihedral angles in a small peptide consisting of three alanine residues. Arrows mark the rotation axes of the dihedrals. Panel B shows a Ramachandran plot for the protein Ubiquitin (PDB code 1UBQ[49]). Each black circle marks the dihedral angles of one residue in Ubiquitin. The cyan lines mark preferred regions of the dihedral angles. The dark blue lines mark the excluded regions due to atomic overlaps. Beta sheets correspond to the preferred region in the upper left, right-handed helices to the preferred central left region, and left-handed helices to the preferred central right region. The Ramachandran plot was generated by MolProbity.[51]*

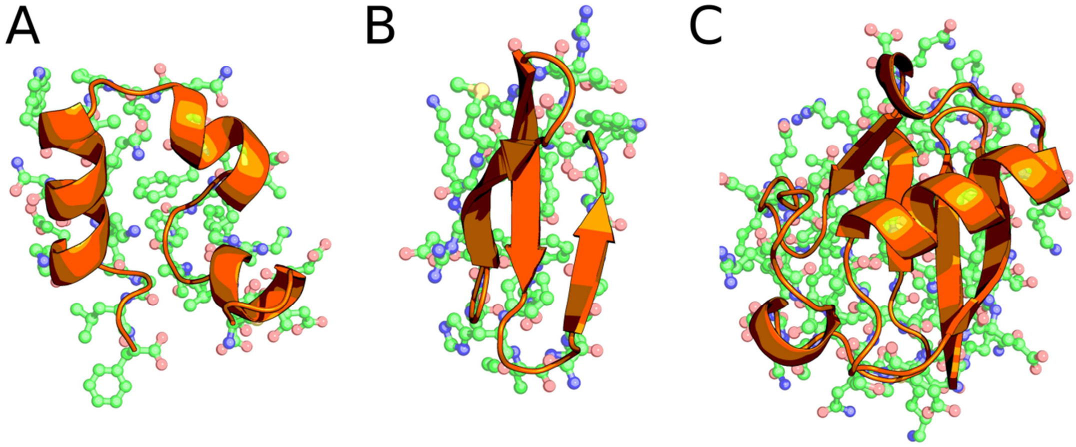


*Figure 2.6. Cartoon representation (orange) of three different proteins in addition to a translucent ball-and-stick representation without hydrogen atoms. The identification of the most common secondary structure elements such as helices and beta sheets is simpler in the cartoon representation. Panel A shows the Villin headpiece (PDB code 1VII[52]) that only contains alpha helices. Panel B shows the WW domain protein (PDB code 2F21[53]) that contains a beta sheet. Panel C shows Ubiquitin (PDB code 1UBQ[49]), a protein containing a mixture of secondary structure elements.*

### *Biological Membranes and Phospholipids*

Biological membranes create a necessary permeability barrier between cells and their environment or even between cell compartments.[54] About 25% of all proteins in eukaryotic genomes bind or associate to membranes. These membrane proteins constitute 50% of all drug targets, wherefore they are an important subject of pharmaceutical research.[55] To understand the functions of these proteins, one has to account for the influence of the biological membrane on their structure and function. Therefore, I will shortly review the constituents of biological membranes and some of their properties.

Biological membranes are composed of a double layer of phospholipids. In turn, Phospholipids are composed of two or three different parts. Those are the headgroup and one or two tails. There exists a large diversity of phospholipids.[56] For example, one of the most common headgroups in cellular membranes contains a phosphate group, a glycerol, and a choline group.[57] However, other groups can also be attached to a phosphate group via biosynthesis.[58] The phospholipid tails are fatty acids that are bound to the glycerol. Those fatty acids differ in their length and saturation of the hydrocarbons. A wide variety of fatty acids is, for example, present in human cell membranes.[59] Figure 2.7 shows the chemical structure of an exemplary phospholipid.

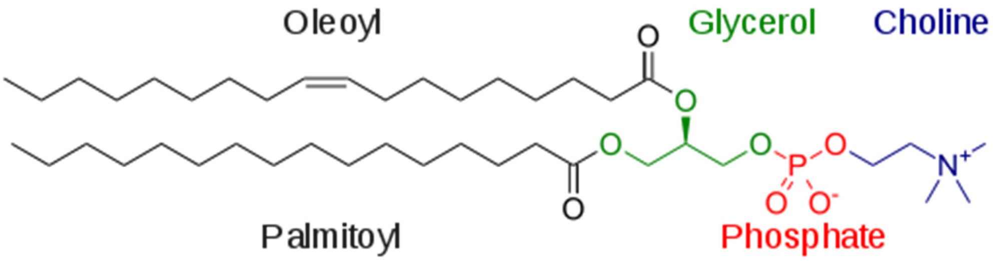


***Figure 2.7. The chemical structure of a POPC phospholipid. The lipid tails are shown in black. The headgroup consists of a glycerol (green), a phosphate group (red), and a choline group (blue).[60]***

Phospholipids are amphipathic molecules, which mean that their long fatty acid tails at one end are hydrophobic, while their headgroups are hydrophilic. This property enables phospholipids to form bilayers in an aqueous environment.[61] The phospholipid tails align in a parallel fashion. The hydrophilic head groups form a layer that shields the polar water molecules from the hydrophobic fatty acid tails. To shield the other end of the fatty acids also from water, a similar second layer can form. The two layers arrange so that the fatty acids face each other while the headgroups face the water. This bilayer is the basic constitutes a biological membrane. Figure 2.8 shows a visualization of such a bilayer.

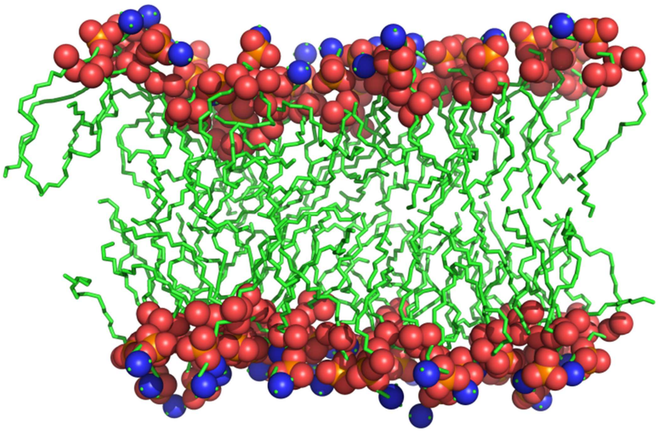


***Figure 2.8. Visualization of a phospholipid bilayer of DOPC lipids.[62,63] The hydrocarbon tails of the fatty acids are shown as green sticks. The nitrogen and phosphorus atoms of the headgroups are shown as blue and orange spheres respectively, while the oxygen atoms of the headgroups and fatty acid tails are shown as red spheres.***

## 2.2 Biomolecular Forcefields

Biomolecular forcefields model the potential energy of biomolecules such as proteins by classical mechanics. They consist of a set of mathematical functions that describe the general form of the interactions present within biomolecules. Their arguments are the coordinates of the molecule's atoms. In addition, these functions contain sets of free parameters. Since quantum mechanics governs the behavior of molecules, these forcefields are only approximations to the real potential energy. Either these parameters are chosen to match the results of elaborate quantum mechanical calculations as closely as possible, or they are determined empirically. In the latter case, the free parameters are optimized so that simulations with this forcefield reproduce specific experimental data.

The significant advantage of this approach is its computational efficiency. It can compute the potential energy for a molecule orders of magnitude faster compared to quantum mechanical methods.[25,64–66] In addition, simple analytical formulas are available to calculate the forces acting on the atoms. Therefore, solving Newton's equations of motion yields the behavior of the molecule. This forms the basis of molecular dynamics, for which these forcefields are typically used.

In this thesis, I will use the AMBER99SB*-ILDN[38–41] forcefield. The reasons for this decision were that it was able to produce repeated folding events of two structurally different small proteins on a rare specialized supercomputer for molecular dynamics simulations.[32] In addition, simulations of larger proteins with this forcefield are able to reproduce experimental NMR data.[67,68] Hence, the forcefield seems to provide a reasonably accurate description of the interactions inside proteins.

The AMBER99SB*-ILDN forcefield consists of the following terms:

$$E = E_{\mathrm{LJ}} + E_{\mathrm{Coulomb}} + E_{\mathrm{bond}} + E_{\mathrm{angle}} + E_{\mathrm{dihedral}} + E_{1-4}. \tag{2.1}$$

The first term contains the Lennard-Jones interactions[69] that model Pauli repulsion due to overlapping electron orbitals and dispersion attraction because of induced electrostatic dipoles. The formula to compute this term is

$$E_{\mathrm{LJ}} = \frac{1}{2} 4 \sum_{i,j=1}^{N} \epsilon_{ij} \left( \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right), \forall\, j \notin \mathrm{excluded}(i). \tag{2.2}$$

The indices $i$ and $j$ denote the atoms of the molecule, $N$ is the total number of atoms in the molecule, $r_{ij}$ is the distance between the two atoms, and $\epsilon_{ij}$ and $\sigma_{ij}$ are given by the two formulas

$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}, \tag{2.3}$$

$$\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j). \tag{2.4}$$

Here, $\epsilon_i$ and $\sigma_i$ are atom type dependent Lennard-Jones parameters for atom $i$. Certain biomolecular forcefields exclude specific interactions between a given atom $i$ and other atoms $j$. These other atoms are contained in the set $\mathrm{excluded}(i)$. The self-interactions $i = j$ also belong to the excluded interactions. For the AMBER99SB*-ILDN forcefield, all Lennard-Jones interactions between atoms that are connected by three or less bonds are excluded. Other forcefield terms are used to model the interactions between those atoms.

Electrostatic interactions caused by the varying electron density of the molecule and the positively charged nuclei are modeled by assigning each atom a partial charge and computing the coulomb interactions between these point charges:

$$E_{\mathrm{Coulomb}} = \frac{1}{2} \frac{1}{4\pi\epsilon_0\epsilon_{\mathrm{s}}} \sum_{i,j=1}^{N} \frac{q_i q_j}{r_{ij}}, \forall\, j \notin \mathrm{excluded}(i). \tag{2.5}$$

Again, $r_{ij}$ is the distance between two atoms, while $q_i$ and $q_j$ are the partial charges, $\epsilon_{\mathrm{s}}$ is the assumed dielectric constant inside the molecule, and $\epsilon_0$ is the vacuum permittivity. One method to compute the partial charges is to compute the electrostatic potential via quantum mechanics and then fit point charges at the position of the nuclei so that the electrostatic potentials agrees with that of the quantum mechanical calculation up to a defined error.[70] The definition of the excluded interactions is usually the same as for the Lennard-Jones term. Together, the Coulomb and the Lennard-Jones interactions form the non-bonded interactions, since the interacting atoms are usually not covalently bound to each other.

The last four terms in Equation (2.1) are short-ranged and describe the bond geometry of the molecular system. These are the so-called bonded interactions. The 1-4 interactions generate the basic torsion potential around the axis of a dihedral angle. The 1-4 interactions include Coulomb and Lennard-Jones interactions between two atoms $i$ and $j$ that are separated by three covalent bonds

$$E_{1-4}(i,j) = f_{\mathrm{LJ}}E_{\mathrm{LJ}}(i,j) + f_{\mathrm{Coulomb}}E_{\mathrm{Coulomb}}(i,j). \tag{2.6}$$

These interactions are scaled with constant factors $f_{\mathrm{LJ}}$ and $f_{\mathrm{Coulomb}}$ respectively. Especially the Lennard-Jones repulsion is responsible for the prohibited regions of the $\Psi$ and $\Phi$ dihedral angles in the Ramachandran plot discussed in section 2.1.

Potential energy changes associated with bond stretching are modeled by $E_{\mathrm{bond}}$. Usually these terms assume a harmonic potential depending on the bond length $d$ with force constant $k$ around the optimal bond length $d_0$

$$E_{\mathrm{bond}}(d) = \frac{k}{2}(d - d_0)^2. \tag{2.7}$$

The same holds for the bond angle term. The potential term has a harmonic dependence on the bond angle $\delta$, a force constant $k_\delta$, and an optimal angle $\delta_0$

$$E_{\mathrm{angle}}(\phi) = \frac{k_\delta}{2}(\delta - \delta_0)^2. \tag{2.8}$$

Dihedral energy terms can be divided into two categories. Improper dihedral angles penalize the deformation of planar chemical groups or rings. In this case, a harmonic dependence on the improper dihedral angle $\theta$ is assumed that has a force constant $k_\theta$ and an optimal dihedral angle $\theta_0$

$$E_{\mathrm{dihedral}}^{\mathrm{improper}}(\theta) = \frac{k_\theta}{2}(\theta - \theta_0)^2. \tag{2.9}$$

Proper dihedral angles define torsion potentials around covalent bonds in addition to the 1-4 interactions of Equation (2.6). For the AMBER99SB*-ILDN forcefield, the corresponding energy terms will be of the form

$$E_{\text{dihedral}}^{\text{proper}}(\theta) = \sum_{n=1}^{4} k_{\theta,n}(1 + \cos(n\theta - \theta_n)). \tag{2.10}$$

The four terms on the right hand side can be interpreted as the first terms of the Fourier series for the torsion potential of the proper dihedral. The coefficients $k_{\theta,n}$ and $\theta_n$ are empirical or semi-empirical parameters. These proper dihedral terms are corrections to the basic torsion potential of the 1-4 interactions. Figure 2.9 provides a sketch of all bonded energy terms, as well as the involved atoms and arguments.

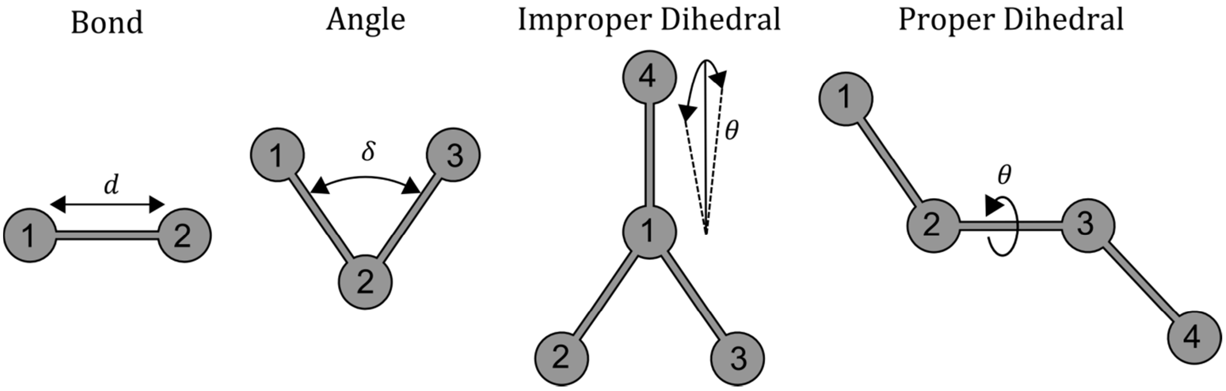


***Figure 2.9. Sketch of the bonded interactions. The bond energy $E_{bond}$ is determined via the distance d between the two atoms 1 and 2. The bond angle energy $E_{angle}$ is defined via the bond angle $\chi$ that is computed from the position of atoms 1 to 3. The improper and proper dihedral angle terms $E_{dihedral}^{improper}$ and $E_{dihedral}^{proper}$ depend on the dihedral angle $\theta$. This dihedral angle is the angle between the planes formed by atoms 1 to 3 and 2 to 4.***

## 2.3 Molecular Surface Definitions

In biomolecular simulations, one usually encounters three definitions of molecular surfaces. They have various applications in biomolecular modeling and simulations. One application is the definition of the boundary between solvent and solute in implicit solvent models, which will be discussed in the next Section 2.4. The different surfaces are the van der Waals surface, the solvent accessible surface and the solvent excluded surface. I will shortly review their definitions and some of their important properties. Figure 2.10A shows a sketch of the differences between the three surfaces.

### *Van der Waals Surface*

The constituents of the van der Waals surface are spheres of radius $r_i = \sigma_i/2$. The parameter $\sigma_i$ is the Lennard-Jones parameter taken from Equations (2.2) and (2.4). These spheres are positioned at the center of every atom. The union of the surfaces of these spheres not located inside any other sphere defines the van der Waals surface. It is visualized in Figure 2.10B. The van der Waals spheres usually do not overlap or only by a small amount. The reason is the strong repulsion of the Lennard-Jones potential in Equation (2.2) at interatomic distances smaller than $\sigma_{ij}$. That is an important property of this surface definition. Fast pairwise methods to approximate the van der Waals surface area exploit this property to account for the overlapping spheres in the computation of the total surface area of a molecule.[71] Another important property is the existence of numerous small cavities between the spheres. Usually these cavities are much smaller than a single water molecule. Therefore, the usage of this surface in continuum electrostatic implicit solvent models results in systematic errors.[72]

### *Solvent Accessible Surface*

Another commonly used molecular surface is the solvent accessible surface proposed by Lee and Richards.[73] As the name already suggests, spherical solvent molecules are excluded from the volume enclosed by this surface. It is generated by adding the probe radius to the radii of the van der Waals spheres. The union of these larger spheres not located inside any other sphere then defines the solvent accessible surface (Figure 2.10C). Each point on that surface will be accessible to the center of a spherical solvent molecule, whose radius is the probe radius. In contrast to the van der Waals surface, this surface only possesses cavities that are large enough to contain at least one spherical solvent molecule.

### *Solvent Excluded Surface*

The third commonly used surface definition is the solvent excluded surface proposed by Richards[74] and Connolly.[75,76] Any point inside this surface is not accessible to any spherical solvent molecule without overlapping with spheres placed at the centers of each atom. For the latter spheres, one can use the Lennard-Jones radii. However, empiric radii are used often to increase agreement with experimental results or explicit molecular dynamics simulations.[77–79] The solvent excluded surface can be generated by taking the volume enclosed in the solvent accessible surface and subtracting all points whose distance to the solvent accessible surface is smaller than the probe radius. The surface of the resulting volume is the solvent excluded surface, which is visualized in Figure 2.10D. This surface is nearly as tight around the solute as the van der Waals surface. However, it does not possess cavities that are smaller than a solvent sphere.

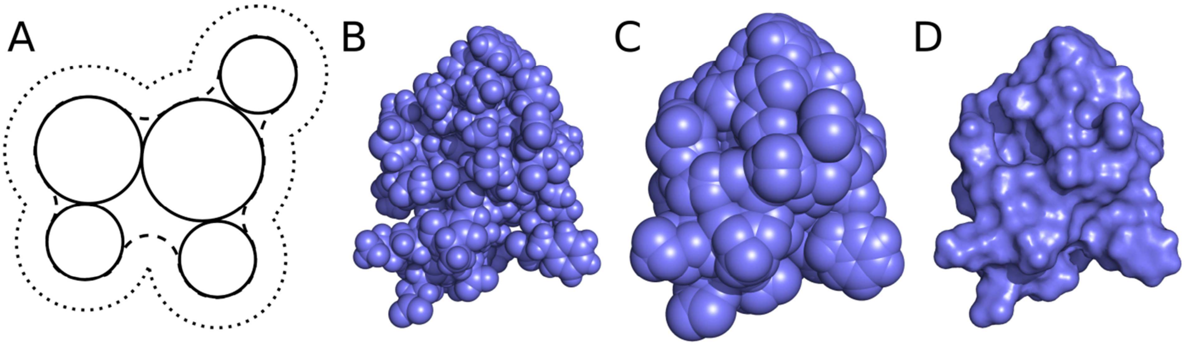


***Figure 2.10. Panel A shows a schematic representation of the three molecular surfaces: van der Waals surface (solid line), solvent accessible surface (dotted line), and solvent excluded surface (dashed line). As an example, the Villin headpiece domain (PDB code 1VII[52]) protein is shown in the van der Waals surface (panel B), the solvent accessible surface (panel C) and the solvent excluded surface (panel D).***

## *2.4 Implicit Solvent Models*

### *Statistical Mechanics Formulation*

Implicit solvent models provide a technique to incorporate the effects of solvent on solutes into simulations without representing every solvent molecule explicitly. As explained in Chapter 1, such an implicit representation is desirable for successful Monte Carlo simulations of biomolecular systems. The formal basis for an implicit solvent description relies on statistical mechanics. I will provide a short overview of these foundations based on the work of Roux and Simonson.[28] They consider a system comprised of a solute U with atomic coordinates $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots\}$ and solvent V with atomic coordinates $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots\}$. Furthermore, they assume that the potential energy $E(\boldsymbol{X}, \boldsymbol{Y})$ of such a system is separable into one term $E_\mathrm{U}$ that only depends on $\boldsymbol{X}$, one term $E_\mathrm{V}$ that only depends on $\boldsymbol{Y}$, and one term for the interaction of solute and solvent $E_\mathrm{UV}$ that depends on both sets of coordinates

$$E(\boldsymbol{X}, \boldsymbol{Y}) = E_\mathrm{U}(\boldsymbol{X}) + E_\mathrm{V}(\boldsymbol{Y}) + E_\mathrm{UV}(\boldsymbol{X}, \boldsymbol{Y}). \tag{2.11}$$

The probability of a microstate in the canonical ensemble with solute configuration $\boldsymbol{X}$ and solvent configuration $\boldsymbol{Y}$ is

$$P(\boldsymbol{X},\boldsymbol{Y}) = \frac{\exp\bigl(-\beta E(\boldsymbol{X},\boldsymbol{Y})\bigr)}{\int d\boldsymbol{X}d\boldsymbol{Y}\exp\bigl(-\beta E(\boldsymbol{X},\boldsymbol{Y})\bigr)}. \tag{2.12}$$

Here, $\beta$ is $\beta = 1/k_BT$, where $k_B$ is the Boltzmann constant and T the temperature. To compute any thermodynamic expectation value $\langle Q(\boldsymbol{X})\rangle$that only depends on the solute coordinates, one has to compute the integral[80]

$$\langle Q(\boldsymbol{X})\rangle = \int d\boldsymbol{X}d\boldsymbol{Y}\, Q(\boldsymbol{X})\, P(\boldsymbol{X},\boldsymbol{Y}). \tag{2.13}$$

According to this equation, the expectation value depends on all solute and solvent configurations $\boldsymbol{X}$ and $\boldsymbol{Y}$. Each microstate contributes $Q(\boldsymbol{X})$ to the expectation value, weighted by the probability $P(\boldsymbol{X},\boldsymbol{Y})$. The goal of implicit solvent models is to create an additional potential term $\Delta G_{\mathrm{S}}(\boldsymbol{X})$ that removes the dependence on $\boldsymbol{Y}$ from Equation (2.13). The name of this term is the solvation free energy. It depends only on the solute coordinates $\boldsymbol{X}$ and not the solvent coordinates $\boldsymbol{Y}$. Together with $E_{\mathrm{U}}(\boldsymbol{X})$ from Equation (2.11) it forms the solute potential of mean force. It is supposed to yield the same expectation values as the original potential. Therefore, Simonson and Roux define a reduced probability function

$$\bar{P}(\boldsymbol{X}) = \frac{\exp\Bigl(-\beta\bigl(E_{\mathrm{U}}(\boldsymbol{X}) + \Delta G_{\mathrm{S}}(\boldsymbol{X})\bigr)\Bigr)}{\int d\boldsymbol{X}\exp\Bigl(-\beta\bigl(E_{\mathrm{U}}(\boldsymbol{X}) + \Delta G_{\mathrm{S}}(\boldsymbol{X})\bigr)\Bigr)}. \tag{2.14}$$

The requirement that no expectation value may change yields the new potential term up to an undefined constant offset $C$

$$\Delta G_{\mathrm{S}}(\boldsymbol{X}) = -\frac{1}{\beta}\ln\Bigl(\int d\boldsymbol{Y}\exp\bigl(-\beta(E_{\mathrm{V}}(\boldsymbol{Y}) + E_{\mathrm{UV}}(\boldsymbol{X},\boldsymbol{Y})\bigr)\Bigr) + C. \tag{2.15}$$

For applications in biomolecular forcefields that use an implicit solvent representation, it is common practice to separate the potential energy of solvent-solute interactions into nonpolar and electrostatic contributions

$$E_{\mathrm{UV}}(\boldsymbol{X},\boldsymbol{Y}) = E_{\mathrm{UV}}^{\mathrm{np}}(\boldsymbol{X},\boldsymbol{Y}) + E_{\mathrm{UV}}^{\mathrm{elec}}(\boldsymbol{X},\boldsymbol{Y}). \tag{2.16}$$

Typically, $E^{\mathrm{np}}$ is the Lennard-Jones interaction of the biomolecular forcefield and $E^{\mathrm{elec}}$ the Coulomb interaction. This differentiation of nonpolar and electrostatic energy terms translates to the solvation free energy according to Roux and Simonson[28]

$$\Delta G_{\mathrm{S}}(\boldsymbol{X}) = \Delta G_{\mathrm{np}}(\boldsymbol{X}) + \Delta G_{\mathrm{elec}}(\boldsymbol{X}). \tag{2.17}$$

The first term on the right hand side describes the reversible work needed to embed the solute in a fixed configuration $\boldsymbol{X}$ into the solvent with all solute charges set to zero. The second term describes the reversible work of charging the solute in a fixed configuration $\boldsymbol{X}$ in the presence of the solvent. These two terms are[28]

$$\Delta G_{\mathrm{np}}(\boldsymbol{X}) = -\frac{1}{\beta}\ln\left(\frac{\int d\boldsymbol{Y}\exp\left(-\beta(E_{\mathrm{V}}(\boldsymbol{Y}) + E_{\mathrm{UV}}^{\mathrm{np}}(\boldsymbol{X},\boldsymbol{Y})\right)}{\int d\boldsymbol{Y}\exp\left(-\beta E_{\mathrm{V}}(\boldsymbol{Y})\right)}\right), \tag{2.18}$$

$$\Delta G_{\mathrm{elec}}(\boldsymbol{X}) = -\frac{1}{\beta}\ln\left(\frac{\int d\boldsymbol{Y}\exp\left(-\beta\left(E_{V}(\boldsymbol{Y}) + E_{\mathrm{UV}}^{\mathrm{np}}(\boldsymbol{X},\boldsymbol{Y}) + E_{\mathrm{UV}}^{\mathrm{elec}}(\boldsymbol{X},\boldsymbol{Y})\right)\right)}{\int d\boldsymbol{Y}\exp\left(-\beta\left(E_{\mathrm{V}}(\boldsymbol{Y}) + E_{\mathrm{UV}}^{\mathrm{np}}(\boldsymbol{X},\boldsymbol{Y})\right)\right)}\right). \tag{2.19}$$

Another reason for this separation into nonpolar and electrostatic contributions relies on the thermodynamic cycle in Figure 2.11. The solvation free energy $\Delta G_{\mathrm{S}}$ is the change in free energy by transferring a solute from a reference environment, e.g. vacuum or a gaseous phase, into the solvent. The solute is kept in a fixed configuration $\boldsymbol{X}$ during this process. One possibility to do this is to first discharge the solute. The associated energy change is $\Delta G_{\mathrm{discharge}}$. Subsequently, the uncharged solute is transferred into the solvent. The required reversible work is $\Delta G_{\mathrm{np}}$. In the last step, the solute is charged again, requiring the reversible work $\Delta G_{\mathrm{recharge}}$.

$$\Delta G_{\mathrm{S}}(\boldsymbol{X}) = \Delta G_{\mathrm{discharge}}(\boldsymbol{X}) + \Delta G_{\mathrm{np}}(\boldsymbol{X}) + \Delta G_{\mathrm{recharge}}(\boldsymbol{X}). \tag{2.20}$$

The work of discharging the solute in vacuum consists only of the negative Coulomb energy $-E_{\mathrm{Coulomb}}(\boldsymbol{X})$ of the solute. On the other hand, charging the solvated solute requires the Coulomb energy $E_{\mathrm{Coulomb}}$ plus an additional term due to the interaction of the solute charges with solvent. Per definition, this additional contribution is the electrostatic part of the solvation free energy $\Delta G_{\mathrm{elec}}$, see Equation (2.17). Thus, the electrostatic contribution to the solvation free energy is also

$$\Delta G_{\mathrm{elec}}(\boldsymbol{X}) = \Delta G_{\mathrm{discharge}}(\boldsymbol{X}) + \Delta G_{\mathrm{recharge}}(\boldsymbol{X}). \tag{2.21}$$

Given the results presented so far, one would need to integrate over all solvent configurations $\boldsymbol{Y}$ to compute the solvation free energy for a single conformation $\boldsymbol{X}$. Moreover, the solvation free energy also depends on the temperature $T$ of the system via the factor $\beta$. Considering the complexity of the Lennard-Jones and Coulomb interactions, it is obvious that there is no trivial solution to arrive at a simple analytic term for $\Delta G_{\mathrm{S}}(\boldsymbol{X})$. However, such an analytic term would allow for an efficient implementation on computers. This, in turn, would enable fast implicit solvent simulations of arbitrary solutes. One solution to this problem is to use approximate implicit solvation models. I will introduce two common approximate models in the remainder of this section, but first, I would like to provide another important definition that I will use throughout this thesis.

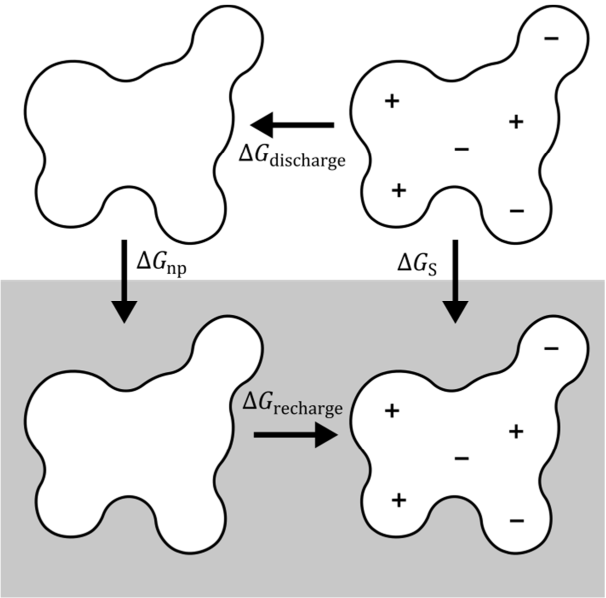


***Figure 2.11. Sketch of a thermodynamic cycle to decompose the solvation free energy $\Delta G_S$ into nonpolar and electrostatic contributions.***

### *Hydration Free Energy*

I would like to point out an important definition that I use throughout this thesis. As explained in the previous subsection, the solvation free energy $\Delta G_{\text{S}}$ is the change in free energy by transferring a solute in a fixed configuration $\boldsymbol{X}$ from a reference environment, e.g. a gaseous state, into the solvent. In contrast, the hydration free energy is the free energy difference between the gaseous state and the solvated state. It does not require the solute to be in a fixed configuration. Therefore, it also accounts for conformational and entropic changes of the solute upon solvation. Experiments are also able to compute this quantity.[81] Thus, the computation of hydration free energies provide a valuable test between theory and simulation on the one hand and experiment on the other hand.[82]

However, the computed hydration free energies form a canonical ensemble with constant particle number, volume and temperature are Helmholtz free energies. In contrast, experiments usually measure the Gibbs free energy because it is easier to control pressure and temperature in laboratories. Nevertheless, these two values can be compared, because the atmospheric pressure under normal conditions is of the order of $10^{-5}$ kcal/(mol Å$^3$), wherefore difference between the Helmholtz free energy and the Gibbs free energy is be negligible according to Roux and Simonson.[28]

Because my thesis is on the theory side, I will give a short overview of the techniques to compute free energy changes between two different states of a solute. I term these states A and B, which are described by the potential functions $U_{\mathrm{A}}(\boldsymbol{X})$ and $U_{\mathrm{B}}(\boldsymbol{X})$ respectively. The thermodynamic coupling parameter $\lambda$ describes the transition of the system from state A to state B, where $\lambda = 0$ corresponds to state A and $\lambda = 1$ corresponds to state B. To be more specific, A will be the vacuum state and B will be the solvated state. I define the potential $U_{\lambda}(\boldsymbol{X})$ as

$$\mathrm{U}_{\lambda}(\boldsymbol{X}) = U_{\mathrm{A}}(\boldsymbol{X}) + \lambda\big(U_{\mathrm{B}}(\boldsymbol{X}) - U_{\mathrm{A}}(\boldsymbol{X})\big) = E_{\mathrm{U}}(\boldsymbol{X}) + \lambda\Delta G_{\mathrm{S}}(\boldsymbol{X}). \tag{2.22}$$

Thermodynamic integration[83] (TI) can yield the free energy difference between the two states by computing

$$\Delta G_{\mathrm{TI}}(A \rightarrow B) = \int_{0}^{1} d\lambda \langle \Delta G_{\mathrm{S}}(\boldsymbol{X})\rangle_{\lambda}. \tag{2.23}$$

The expectation value $\langle \Delta G_{\mathrm{S}}(\boldsymbol{X})\rangle_{\lambda}$ averages over all configurations $\boldsymbol{X}$ using the probability of a microstate $\bar{P}_{\lambda}(\boldsymbol{X})$ at a fixed value of $\lambda$

$$\bar{P}_{\lambda}(\boldsymbol{X}) = \frac{\exp\big(-\beta U_{\lambda}(\boldsymbol{X})\big)}{\int d\boldsymbol{X} \exp\big(-\beta U_{\lambda}(\boldsymbol{X})\big)}. \tag{2.24}$$

Another method to compute the free energy difference between two systems is free energy perturbation (FEP) proposed by Zwanzig[84]

$$\Delta G_{\mathrm{FEP}}(A \rightarrow B) = -\frac{1}{\beta} ln\big(\langle \exp\big(-\beta\Delta G_{\mathrm{S}}(\boldsymbol{X})\big)\rangle_{\mathrm{A}}\big). \tag{2.25}$$

Here, the average runs over the system in state A. However, I note that the definition of the states A and B is exchangeable. For proper convergence of this method, it requires that there be sufficient overlap between the states A and B. This means that there has to be a sufficiently large number of configurations $\boldsymbol{X}$ that have a non-vanishing microstate probability in both states A and B.

A third method to compute the free energy difference between two states is the Bennet acceptance ratio method (BAR).[85] I will use this method in my thesis. The free energy difference is estimated via

$$\Delta G_{\mathrm{BAR}}(A \rightarrow B) = \frac{\langle \min(\exp(\beta\Delta G_{\mathrm{S}}(\boldsymbol{X})), 1)\rangle_{\mathrm{A}}}{\langle \min(\exp(-\beta\Delta G_{\mathrm{S}}(\boldsymbol{X})), 1)\rangle_{\mathrm{B}}}. \tag{2.26}$$

This approach was shown to be near-optimal and highly efficient.[85,86]

### *Physical Properties of Water*

Since the focus of my thesis is the study of biomolecules such as proteins, the solvent will be water because it constitutes the physiological environment of many proteins. To construct an approximate implicit solvent model, one has to understand the physical properties of water and the effects that these properties cause. Thus, I will briefly review some of the physical properties of water.

One remarkable property of water molecules is their high dipole moment of about 3 Debye in solution.[87] Therefore, water is a polar solvent. Consequently, water molecules around polar or charged solutes reorient and shield the electrostatic field created by the solute. The high reported relative dielectric constant of water of 78.3 to 78.5 at 25°C reflects this behavior of water.[88,89] The temperature dependence of the dielectric constant of water on the temperature $t$ in degree Celsius is according to Malmberg and Maryott[88]

$$\epsilon_{\mathrm{w}} = 87.740 - 0.4008t + 9.398 \cdot 10^{-4} t^2 - 1.410 \cdot 10^{-6} t^3. \tag{2.27}$$

Another important property of water is the presence of hydrogen bonds between different water molecules. The two hydrogen atoms of a water molecule can act as hydrogen bond donors, while the oxygen atom acts as an acceptor for two other hydrogen bonds. For example, due to this favorable interaction, only less than 5% of all water molecules are not engaged in hydrogen bonding at any given time at a temperature of 10°C.[90]

Another property of liquid water is the hydrophobic effect, which causes oil-water mixtures not to mix.[91] Nonpolar solutes, such as alkanes, can disrupt the tetrahedral hydrogen bond networks present in water, because they do not possess hydrogen bond donors or acceptors.[92] It was believed that this disruption causes a rearrangement and strengthening of the hydrogen bond pattern around the solute. The strong hydrogen bonds would reduce the translational and rotational degrees of freedom. This results in a decrease of the system's entropy.[90] However, recent experiments indicate that the hydrogen-bonding pattern may not be strengthened, while the reorientation of the water molecules at hydrophobic surfaces still occurs.[93]

### *Electrostatic Continuum Solvation Models*

In the last decades, scientists have developed a wide variety of implicit solvent models that provide approximate descriptions of the effect of the solvent on the solute.[27,28,94–97] In my thesis, I will focus on continuum implicit solvent models. These allow the approximate computation of the solvation free energy with reasonable accuracy at reduced computational cost.[94,98]

Continuum implicit solvent models describe the solvent by a continuous dielectric medium. Dielectric media respond with polarization to an electric field $\boldsymbol{E}(\boldsymbol{x})$, which can be generated by charge distribution $\rho(\boldsymbol{x})$, e.g. the charge distribution of the solvated molecule. Implicit continuum solvent models assume that the solvent's response to that electric field is local, homogenous, isotropic, and linear. Local means that the polarization density $\boldsymbol{P}(\boldsymbol{x})$ of the medium at position $\boldsymbol{x}$ does not depend on the polarization density of the medium at any other position $\boldsymbol{y}$. Homogenous means that the polarization density at both positions is equal if the electric field is equal. Furthermore, in an isotropic solvent the polarization density does not depend on the orientation of the electric field. Finally, linear means that the polarization density is proportional to the electric field via the susceptibility $\chi$. With these assumptions the polarization density is

$$\boldsymbol{P}(\boldsymbol{x}) = \epsilon_0 \chi \boldsymbol{E}(\boldsymbol{x}). \tag{2.28}$$

This yields the dielectric displacement field

$$\boldsymbol{D}(\boldsymbol{x}) = \epsilon_0 \boldsymbol{E}(\boldsymbol{x}) + \boldsymbol{P}(\boldsymbol{x}) = \epsilon_0 \epsilon_\mathrm{r} \boldsymbol{E}(\boldsymbol{x}), \tag{2.29}$$

where the relative dielectric constant is defined as $\epsilon_\mathrm{r} = 1 + \chi$ and $\epsilon_0$ is again the vacuum permittivity. With that, it is possible to compute the energy necessary to assemble a charge distribution within a dielectric medium as[99]

$$W(\boldsymbol{x}) = \frac{1}{2} \int \rho(\boldsymbol{x}) \Phi_\mathrm{e}(\boldsymbol{x}) d^3x = \frac{1}{2} \int \boldsymbol{E}(\boldsymbol{x}) \cdot \boldsymbol{D}(\boldsymbol{x})\, d^3x. \tag{2.30}$$

Using the above assumptions, the electrostatic potential $\Phi_\mathrm{e}(\boldsymbol{x})$ can be found by solving the Poisson equation[99]

$$\Delta \Phi_\mathrm{e}(\boldsymbol{x}) = -\frac{\rho(\boldsymbol{x})}{\epsilon_0 \epsilon_\mathrm{r}}. \tag{2.31}$$

To obtain a unique electrostatic potential, boundary conditions need to be defined and fulfilled. Let $\boldsymbol{x}_\mathrm{b}$ be a position vector on the boundary between two dielectric regions. The normal vector of the boundary surface is $\boldsymbol{n}(\boldsymbol{x}_\mathrm{b})$ and the relative dielectric constants of the two dielectric regions $\epsilon_1$ and $\epsilon_2$. The following boundary conditions for the electric and the displacement fields in the corresponding regions must hold at the interface[99]

$$(\boldsymbol{D_1}(x_\mathrm{b}) - \boldsymbol{D_2}(x_\mathrm{b})) \cdot \boldsymbol{n}(x_\mathrm{b}) = 0, \tag{2.32}$$

$$\boldsymbol{E_1}(x_\mathrm{b}) \times \boldsymbol{n}(x_\mathrm{b}) = \boldsymbol{E_2}(x_\mathrm{b}) \times \boldsymbol{n}(x_\mathrm{b}). \tag{2.33}$$

While a vast number of Poisson-Boltzmann solvers have been developed to compute the electrostatic potential from Equation (2.31) under these boundary conditions, the numerical methods are also computationally demanding.[97] Even at low accuracy, such methods need about 0.3 s to 22 s to compute a single solvation free energy.[100] Considering that tens of millions of such evaluations are necessary for the simulation of biomolecular systems via molecular dynamics or Monte Carlo methods, the simulations would take more than a year to complete. Therefore, the computation time of Poisson-Boltzmann solvers is at least one to two orders of magnitude too high for such simulations. That is the reason the approximate generalized Born model has become so popular. It provides a computationally more efficient alternative while retaining good agreement with Poisson-Boltzmann results.[94,97,100–103]

The generalized Born model is based on the Born model of ion hydration proposed by Max Born.[104] Within that model, the solvation free energy of an ion with charge $q$, the ion's assumed dielectric constant $\epsilon_{\mathrm{s}}$, and water's dielectric constant $\epsilon_{\mathrm{w}}$ is given by

$$\Delta G_{\mathrm{Born}} = -\frac{1}{4\pi\epsilon_0}\left(\frac{1}{\epsilon_{\mathrm{s}}} - \frac{1}{\epsilon_{\mathrm{w}}}\right)\frac{q^2}{R}. \tag{2.34}$$

The Born radius $R$ is an empiric parameter used to match experimentally determined solvation free energies. It is a measure of the amount of polarization induced in the surrounding solvent by the ion's charge. The induced polarization charges can in turn interact with the ion charge.

Still et al.[105] extended this model from ions to molecules

$$\Delta G_{\mathrm{GB}} = -\frac{1}{8\pi\epsilon_0}\left(\frac{1}{\epsilon_{\mathrm{s}}} - \frac{1}{\epsilon_{\mathrm{w}}}\right)\sum_{i,j=1}^{N}\frac{q_i q_j}{r_{ij}}\frac{1}{f_{\mathrm{GB}}\left(r_{ij}, R_i, R_j\right)}. \tag{2.35}$$

The analytical form of the generalized Born model is very similar to the Coulomb term in Equation (2.5). Again $\epsilon_{\mathrm{s}}$ is the assumed dielectric constant inside the solute. However, there are a few notable differences. The sign of the generalized Born term is the opposite of the Coulomb term. The former term also includes self-energies that correspond to the sum of the Born terms in Equation (2.34) for each atom

$$\varDelta G_{\mathrm{self}} = -\frac{1}{4\pi\epsilon_0}\left(\frac{1}{\epsilon_{\mathrm{s}}} - \frac{1}{\epsilon_{\mathrm{w}}}\right)\sum_{i=0}^{N}\frac{q_i^2}{R_i}. \tag{2.36}$$

In addition, the factor $f_{\mathrm{GB}}$ in Equation (2.35) scales the interaction terms $i \neq j$ depending on the distance $r_{ij}$ and Born radii $R_i$ and $R_j$ of the atoms in question

$$f_{\mathrm{GB}}(r_{ij}, R_i, R_j) = \sqrt{1 + \frac{R_i R_j}{r_{ij}^2} \exp\left(-\frac{r_{ij}^2}{4R_i R_j}\right)}. \tag{2.37}$$

These terms model the interaction of induced polarization charges by atom $i$ with the charge of another atom $j$. In conclusion, the generalized Born term results in a shielding of the Coulomb interaction between two atoms due to the induced polarization charges.

The remaining open question is how to compute the Born radii. According to Equation (2.36), one would have to compute the solvation free energy for the entire molecule if only atom $i$ is charged to get the Born radius for that atom. Unfortunately, this method would also require computationally expensive Poisson-Boltzmann calculations. Therefore, approximate methods to compute these Born radii are desired. The so-called Coulomb field approximation assumes that the electric displacement field caused by a solute point charge is of the form

$$\boldsymbol{D}_i \approx q_i \frac{\boldsymbol{x} - \boldsymbol{x}_i}{|\boldsymbol{x} - \boldsymbol{x}_i|^3}. \tag{2.38}$$

With this approximation, Born radii may be estimated by the integral expression[97]

$$\frac{1}{R_i} = \frac{1}{4\pi} \int_{\mathrm{water}} \frac{d^3x}{|\boldsymbol{x} - \boldsymbol{x}_i|^4}. \tag{2.39}$$

For each Born radius, an integral over the whole space outside the solute has to be solved. Although the integrand is very simple, the integration region is non-trivial due to the complex surface of large molecules such as proteins. An example is the solvent excluded surface introduced in Section 2.3. To enhance agreement with Poisson-Boltzmann calculations, Lee et al. introduced corrections to the integral expression of Equation (2.39).[106,107] Grycuk proposed another integral expression to compute the Born radii, which fully agrees with solutions of the Poisson equation for the case of a spherical solute and an infinite dielectric constant of the solvent[108]

$$\frac{1}{R_i^3} = \frac{3}{4\pi} \int_{\mathrm{water}} \frac{d^3x}{|\boldsymbol{x} - \boldsymbol{x}_i|^6}. \tag{2.40}$$

This integral expression was shown to be reasonable accurate also for non-spherical solutes and finite solvent dielectric constants. In addition, it is expected to be the most efficient,[102] wherefore I will use it in my thesis.

Considering the strong assumptions used in the implicit continuum models so far, one should be aware of their implications. Since water forms hydrogen-bonding networks, these may induce correlation between the orientations of different water molecules. Therefore, the local response assumption may not hold. In addition, the dielectric constant of water also varies for very high strengths of the external electric field,[109] wherefore water's to such a field response is not linear anymore.

Some implications of these two effects have been studied by Gong and Freed[110] or Bardhan.[111] They find that both effects lead to smaller penalties of removing ions from the solvent than compared to the simple Born model. However, they only investigated cases where the Born model and the advanced models used the same dielectric surface. As Bardhan explained, nonlocal effects lead to an induced surface charge distribution that is located further away from the ion. That is what causes the lower charge burial penalty.[111] Therefore, I would like to point out that using a different ion radius in the Born model might partly correct these discrepancies, wherefore the real advantage of these models is still unclear. Since they are also computationally very expensive,[111,112] I will not consider them further in my thesis.

### *Implicit Nonpolar Solvation Models*

Implicit nonpolar solvation models should include the Lennard-Jones interactions between solute and solvent as well as enthalpic or entropic changes in the solvent itself, such as the hydrophobic effect. Early models to describe such effects in an efficient manner were proposed by Eisenberg and McLachlan[113] and Ooi et al.[114] These models approximate the nonpolar contribution to the solvation free energy $\Delta G_{\mathrm{np}}$by multiplying each atomic solvent accessible surface area $A_i$ with an atom type dependent surface tension coefficient $\gamma_i$

$$\Delta G_{\mathrm{np}} \approx \Delta G_{\mathrm{SASA}} = \sum_{i=1}^{N} \gamma_i A_i. \tag{2.41}$$

Further support comes from experimental observations that the solvation free energy of hydrophobic hydrocarbons correlates well with the solvent accessible surface area of those molecules.[115,116] This correlation is also present for analogs of hydrophobic amino acid side chains.[117] Early theoretical investigations by Pierotti using scaled particle theory also support solvent accessible surface area models.[118]

Furthermore, Gilson and Honig[119] proposed that attractive dispersion interactions between solvent and solute be negligible in a first order approximation. They argued that these interactions should be of the same order of magnitude as the solute-solute dispersion interactions. Consequently, nonpolar solvation is only modeled by Equation (2.41) in many generalized Born based implicit solvent models.[105–107,120–124]

However, more recent studies showed that such an approximation leads to errors in estimates of the hydration free energy for cyclic alkanes,[125] the solvation free energy of large macromolecules[126] or the differences of solvation free energies for proteins in different conformations.[127] Therefore, nonpolar solvation should also be modeled by taking into account the solvent accessible volume (SAV) for small molecules and an explicitly account for attractive solute-solvent dispersion interactions[125,127–129]

$$\Delta G_{\text{np}} \approx \Delta G_{\text{SASA}} + \Delta G_{\text{SAV}} + \Delta G_{\text{dispersion}}. \tag{2.42}$$

For molecules such as proteins, the volume term will again be negligible since they are macromolecules. This leaves the computation of $\Delta G_{\text{dispersion}}$ for practical applications in molecular simulations as an open question. Using the probability function of Equation (2.12), the averaged solute-solvent dispersion interaction for a solute in configuration $\boldsymbol{X}$ is given by

$$\langle E_{\text{dispersion}}(\boldsymbol{X})\rangle = \int d\boldsymbol{Y}\, E_{\text{dispersion}}(\boldsymbol{X},\boldsymbol{Y})\; P(\boldsymbol{X},\boldsymbol{Y}). \tag{2.43}$$

Following the suggestions of Floris and Tomasi,[128] this expression can be approximated for a given average number density of water molecules at position $\boldsymbol{x}$ around the solute in configuration $\boldsymbol{X}$ $\langle \rho_{\text{w}}(\boldsymbol{x})\rangle_{\boldsymbol{X}}$.

$$\langle E_{\text{dispersion}}(\boldsymbol{X})\rangle \approx \Delta G_{\text{dispersion}} = \sum_{i}^{N} \int d^3x\; E_{\text{dispersion}}(\boldsymbol{x}_i, \boldsymbol{x}) \langle \rho_{\text{w}}(\boldsymbol{x})\rangle_{\boldsymbol{X}}. \tag{2.44}$$

If the Lennard-Jones potential is used to describe solute-solvent dispersion, $E_{\text{disperion}}$ is the attractive component of this potential term. According to the well-established Weeks-Chandler-Anderson (WCA) decomposition, this attractive term is given by[130]

$$E_{\text{WCA}}^{\text{attractive}}(\boldsymbol{x}_i, \boldsymbol{x}) = E_{\text{LJ}}(\boldsymbol{x}_i, \boldsymbol{x}) \theta_{\text{H}}\left(|\boldsymbol{x}_i - \boldsymbol{x}| - 2^{\frac{1}{6}}\sigma_{i\text{w}}\right) - \epsilon_{i\text{w}}\theta_{\text{H}}\left(-|\boldsymbol{x}_i - \boldsymbol{x}| + 2^{\frac{1}{6}}\sigma_{i\text{w}}\right), \tag{2.45}$$

where $\theta_{\text{H}}$ is the Heavyside function and $E_{\text{LJ}}(\boldsymbol{x}_i, \boldsymbol{x})$ is the Lennard-Jones potential between atom $i$ at position $\boldsymbol{x}_i$ and a water molecule located at position $\boldsymbol{x}$, defined in Equation (2.2). The parameters $\epsilon_{i\text{w}}$ and $\sigma_{i\text{w}}$ are taken from Equations (2.3) and (2.4) respectively. The index w denotes the Lennard-Jones parameters of the water molecule's oxygen atom. The contributions of water's hydrogen atoms are neglected. A simpler decomposition of the Lennard-Jones potential into attractive and repulsive terms is given by the so-called 6-12 decomposition, where the attractive term $E_{6-12}^{\text{attractive}}$ is simply the second term of Equation (2.2) according to Gallicchio and Levy[131]

$$E_{6-12}^{\text{attractive}}(\boldsymbol{x}_i, \boldsymbol{x}) = -4\frac{\epsilon_{i\text{w}}\sigma_{i\text{w}}^6}{|\boldsymbol{x}_i - \boldsymbol{x}|^6}. \tag{2.46}$$

Tan et al.[132] have shown that the WCA decomposition yields results in better agreement with explicit solvent simulations than the 6-12 decomposition. Unfortunately, they also found that the models still have problems in reproducing nonpolar attraction between dimers.

Nevertheless, the 6-12 decomposition is simpler and, therefore, better suited for implementation into efficient molecular simulations. Making the same assumptions as in the continuum electrostatics model, namely the uniform distribution of water outside the solute cavity, the integral in Equation (2.44) gives the dispersion contribution to the nonpolar solvation free energy. It is a striking coincidence that the dispersion integrals in Equation (2.44) are of the same form as the Born radii integrals in Equation (2.40), if the 6-12 decomposition is used. Unfortunately, the integration region may differ, since the atomic radii used to construct the dielectric surface are usually empirical parameters, which may not be optimal for the calculation of the dispersion contribution $\Delta G_{\text{dispersion}}$ in Equation (2.42).

Although the dispersion integrals and the Born radii integrals are of similar form, applications would require the estimate of two of these integrals for each atom in molecular simulations. As I will show in Section 3.4, the estimate of these integrals together with the computation of the solvent accessible surface area is the computationally most expensive step in the evaluation of the energy of the system. Therefore, the extension of the nonpolar model beyond the solvent accessible surface area approach is likely to induce severe performance penalties, which will restrict size of representative ensembles that can be generated by Monte Carlo simulations. Thus, I will restrict the simulation of proteins to the standard solvent accessible surface area model. Only for the study of small molecule hydration free energies in Chapter 4, I will take the explicit modeling of the attractive dispersion interactions into account.

## 2.5 Monte Carlo Simulation Techniques

### Metropolis Monte Carlo

The estimate of a physical property of a solute-solvent system by Equation (2.13) is computationally very expensive or even impossible, if the system may access an infinite number of microstates. To solve this problem, Metropolis et al.[133] proposed an algorithm to create a finite set of $N_\mathrm{E}$ representative states for a given system. The average value of a physical observable $Q$ from that representative set converges to the expectation value $\langle Q \rangle$ if the set is large enough

$$\langle Q \rangle \approx \frac{1}{N_\mathrm{E}} \sum_{i=1}^{N_\mathrm{E}} Q_i. \tag{2.47}$$

As can be seen from this equation, all states of the ensemble contribute equally to the expectation value. Their proposed algorithm to create such an ensemble is to start at a random configuration of the system $\{\boldsymbol{X}_0, \boldsymbol{Y}_0\}$ and then perturb the system via a defined transformation and propose this new configuration $\{\boldsymbol{X}_1, \boldsymbol{Y}_1\}$. It will be added to the ensemble with probability

$$p_\mathrm{accept} = \min\big(1, \exp\big(-\beta E(\boldsymbol{X}_1, \boldsymbol{Y}_1) - E(\boldsymbol{X}_0, \boldsymbol{Y}_0)\big)\big), \tag{2.48}$$

where $\beta = 1/k_B T$. Here, $k_B$ is the Boltzmann constant and $T$ is the temperature of the system. If the new configuration is rejected, the old configuration will be added to the ensemble, wherefore it may be present more than once in the ensemble. This process is iterated with the latest configuration in the ensemble. As shown by Metropolis et al., the structures in the ensemble will approach the Boltzmann distribution, if the perturbations follow the detailed balance condition[133]

$$\pi(\boldsymbol{X}_i, \boldsymbol{Y}_i) p_{ij} = \pi\big(\boldsymbol{X}_j, \boldsymbol{Y}_j\big) p_{ji}, \tag{2.49}$$

Here $\pi(\boldsymbol{X}_i, \boldsymbol{Y}_i)$ is the probability of being in configuration $\{\boldsymbol{X}_i, \boldsymbol{Y}_i\}$ and $p_{ij}$ the probability to perturb the system into state $\{\boldsymbol{X}_j, \boldsymbol{Y}_j\}$ from state $\{\boldsymbol{X}_i, \boldsymbol{Y}_i\}$.

In principle, it is now possible to estimate any expectation value. However, Metropolis et al. explicitly stated that it is unknown how fast the ensemble will approach the Boltzmann distribution. Therefore it is unknown, at which point the ensemble will be representative. This speed of convergence will strongly depend on the type of system and the possible chosen set of transformations.[133]

### *Parallel Tempering*

Considering that an arbitrary system may contain many local energetic minima separated by high barriers, the transition of the system between these minima is very unlikely. The reason is the suppressed probability to accept new configurations with higher energy, see Equation (2.48). The low probability for crossing energy barriers is one reason for the before-mentioned possible slow convergence of the algorithm to the Boltzmann distribution. Given a multidimensional system, one could increase this convergence by applying perturbations that take the system directly from one minimum to another. However, guessing such perturbations without prior knowledge about the locations of the barriers and minima is non-trivial. In addition, such perturbations would have to satisfy the detailed balance condition of Equation (2.49). Nevertheless, quite a few algorithms exist that allow a faster convergence of the ensemble to the Boltzmann distribution.

One such algorithm is parallel tempering (PT) that I will use for my simulations. It was first proposed by Swendsen and Wang[134] and extended by Geyer[135] according to Deem and Earl.[136] Hansmann[137] first applied this method to a biomolecular system. He showed that this algorithm could overcome energy barriers between multiple local energy minima successfully. Therefore, the convergence speed of the representative ensemble to the Boltzmann distribution increases considerably.

Parallel tempering considers $N_\mathrm{T}$ identical independent systems called replica in possibly different configurations $\{\boldsymbol{X}^k, \boldsymbol{Y}^k\}$ at different temperatures $T_k$. For each of the systems a Metropolis Monte Carlo simulation is run for a certain number of steps at the corresponding temperature $T_k$. Afterwards, an exchange of the temperatures between two systems $T_k$ and $T_{k+1}$ is attempted and accepted with probability

$$p_\mathrm{PT} = \min\big(1, \exp\big(-\Delta\beta_{k,k+1}\Delta E_{k,k+1}\big)\big), \tag{2.50}$$

where $\Delta\beta_{k,k+1}$ and $\Delta E_{k,k+1}$ are defined as

$$\Delta\beta_{k,k+1} = \frac{1}{k_B T_{k+1}} - \frac{1}{k_B T_k}, \tag{2.51}$$

$$\Delta E_{k,k+1} = E\big(\boldsymbol{X}^{k+1}, \boldsymbol{Y}^{k+1}\big) - E\big(\boldsymbol{X}^k, \boldsymbol{Y}^k\big). \tag{2.52}$$

The probability $p_\mathrm{PT}$ guarantees that the distribution of states for each ensemble $k$ will converge to the Boltzmann distribution. Moreover, the system will converge much faster, since for a given energy barrier, the probability for the system to overcome this barrier will be much higher at high temperatures according to Equation (2.48).

To ensure a reasonable exchange probability between temperatures $T_k$ and $T_{k+1}$, the temperature intervals have to be chosen carefully, so that there is sufficient overlap of the energy distributions at consecutive temperatures. Since the expectation value of the energy and the fluctuations of the energy will be system-specific, they will have to be adapted to each system.

# 3 Development and Implementation of Implicit Solvent Model and Forcefield

This chapter focuses on the development and implementation of methods that enable Monte Carlo simulations of proteins in an implicit solvent model within the SIMONA simulation framework.[37] The first section discusses the challenges of transferring a common biomolecular force field usually used in molecular dynamics simulations to Monte Carlo simulations. Subsequently, the section explains the details of the implementation in SIMONA and presents results on the performance of the implementation. The second section gives details of the parallelization of a method to compute the solvent accessible surface area of proteins, which is used in the nonpolar contribution to implicit solvent model (see section 2.4). The third section introduces an efficient method for the accurate computation of Born radii in the generalized Born implicit solvent model introduced in section 2.4. It explains the underlying algorithm developed by me and provides an assessment of the accuracy of the model. Finally, this section demonstrates the efficiency of my method in comparison to previously published methods. In the last section, I will present results on the Monte Carlo simulation performance that can be achieved with these methods implemented by me into the SIMONA simulation framework.

## 3.1 SIMONA Implementation of the AMBER99SB*-ILDN Forcefield

As explained in Section 2.2, I will use the AMBER99SB*-ILDN forcefield in this thesis because of its proven accuracy. The forcefield is based on the Parm94 parameterization,[138] but uses partial charges created with the RESP[139] scheme and improved torsional potentials to yield significantly improved internal molecular energies in comparison to high level *ab initio* calculations.[38] Hornak et al. further improved backbone dihedral torsional parameters. Their improvement yields a balance between the propensity of secondary structure elements in better agreement with PDB data and experimental NMR data, especially for Alanine and Glycine residues.[39] Best and Hummer further fine-tuned the forcefield with an additional set of backbone dihedral torsion parameters that yield secondary structure propensities in better agreement with experiments.[40] Lindorff-Larsen et al. have improved side chain torsion parameters to yield rotamer distributions in better agreement with PDB statistics and experimental NMR data.[41] All these developments are included in the AMBER99SB*-ILDN forcefield. I note that the recommended water model for this forcefield is the explicit TIP3P[140,141] water model.

To enable Monte Carlo simulations of proteins with common biomolecular forcefields, I will use the SIMONA Monte Carlo simulation framework.[37] Monte Carlo simulations with common biomolecular forcefields such as AMBER99SB*-ILDN were previously not possible with SIMONA, because the framework lacked some of the necessary forcefield terms and a matching implicit solvent model. The first step was to implement the basic terms of the AMBER99SB*-ILDN molecular forcefield into SIMONA in an efficient manner. Therefore, some differences between Monte Carlo and molecular dynamics simulations had to be taken into account. For molecular dynamics, forces are the focus of computation. They are required to solve Newton's equations of motion. However, for Monte Carlo simulations, the total energy of the system is the focus of computation as explained in Section 2.5, while forces are not required.

Another difference is that forces in molecular dynamics have to be computed on a per-atom basis. The total energy in Monte Carlo simulations has to be computed for the whole system. As a result, the pairwise non-bonded interactions described in Section 2.2 require one additional summation for the computation of the total energy compared to the computation of the forces for each atom. Since per-atom energies of Equations (2.2) or (2.5) may have opposite signs and absolute values of different orders of magnitude, the final summation of the total energy may be prone to numerical errors due to the finite precision of floating point numbers on computers (see Appendix A.1). Consequently, my implementation will compute per-atom non-bonded energies in single precision for better performance, but sum these energies in double precision. Goetz et al showed that this scheme provides increased accuracy over summing the per-atom energies in single precision only.[142] This should provide a good compromise between numerical accuracy and computational performance.

A further modification required for the application of the AMBER99SB*-ILDN forcefield to Monte Carlo simulations is the assignment of Lennard-Jones parameter to all atoms. In the original forcefield, some hydrogen atoms have no Lennard-Jones parameters, but do have a partial charge assigned. These hydrogen atoms are covalently bound to much larger atoms such as oxygen. The missing parameters may lead to a Coulomb collapse of these hydrogen atoms with other nearby atoms that have no covalent bond to the hydrogen according to Equations (2.2) and (2.5). The reason is the neglected repulsion of the Lennard-Jones potential due to the missing Lennard-Jones parameters. This neglected repulsion poses no problem for molecular dynamics simulations that start from an energetically minimized conformation without a Coulomb collapse. In typical settings of such a simulation, the repulsion of the larger atom to which the hydrogen is bound will pose a large enough energy barrier for any other approaching charged particle to prevent the Coulomb collapse.

However, the Monte Carlo algorithm may propose a perturbed configuration in which the hydrogen without Lennard-Jones parameters is on top of another atom. In that case, the Coulomb attraction between the hydrogen atom and the other atom may overcome the Lennard-Jones repulsion between the other atom and the nearby larger atom, to which the hydrogen is bound. This event traps the system at a practically infinite negative energy. To prevent this Coulomb collapse, I have assigned Lennard-Jones parameters to all hydrogen atoms missing them. The according parameters are $\sigma = 1.06908$ Å and $\epsilon = 0.00016$ kcal/mol. Here, $\sigma$ is equal to the Lennard-Jones radii of other hydrogen atoms. The small arbitrary value of $\epsilon$ should prevent the Coulomb collapse, but should not modify the forcefield otherwise.

Another important aspect of my AMBER99SB*-ILDN implementation is that SIMONA takes only dihedral degrees of freedom for proteins into account.[37] Therefore, $E_{\text{bond}}$ and $E_{\text{angle}}$ of Equations (2.7) and (2.8) will be constant during the simulations. Thus, these potential terms can be omitted.

The assignment of all parameters, e.g. partial charges, Lennard-Jones parameters, or dihedral terms to a given protein structure is done by the freely available pdb2gmx program of GROMACS.[25] The SIMONA preprocessor reads in the parameter files generated by pdb2gmx and converts the values to the XML input file format of SIMONA.

The implementation of the AMBER99SB*-ILDN dihedral potential into SIMONA is straightforward. Since the number of torsion potential terms depends linearly on the size the protein, this term is not performance-relevant. Therefore, no optimizations of the program code are needed. To check the correctness and accuracy of the resulting dihedral potential term, I among others have compared the implementation in SIMONA to that in GROMACS. I have used an already published test set of 611 native protein structures[100] for this comparison. Figure 3.1 shows a histogram of the relative errors between the dihedral energies of the two implementations. The average relative error is $2.5 \cdot 10^{-6}$ and the maximum relative error is $2.0 \cdot 10^{-5}$. These errors are acceptable when taking into account the limited precision of floating point computations. One reason for the small errors may be a different implementation of the cosine function used in the GROMACS package compared to default the implementation of the C++ standard library, which I use in SIMONA (see Equation (2.10)).

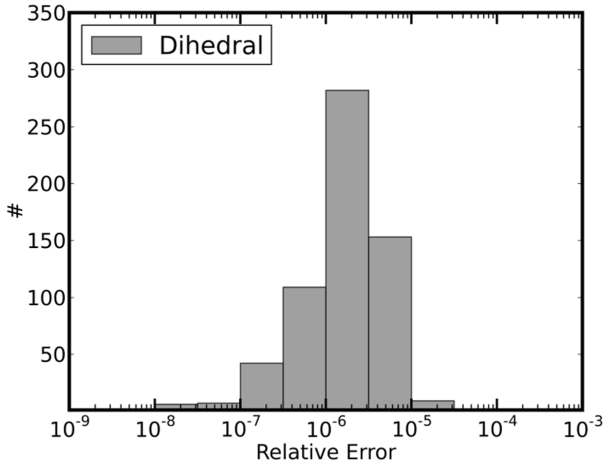


***Figure 3.1. Histogram of relative errors of the dihedral potential energy between implementations in GROMACS and SIMONA for the AMBER99SB*-ILDN forcefield. I used a set of 611 native protein structures[100] for this comparison.***

Now I turn to the implementation of the non-bonded interactions of the AMBER99SB*-ILDN forcefield described by Equations (2.2) to (2.5). These interactions are long range and the number of them is proportional to $N^2$, where $N$ is the number of atoms in the system. As a result, their computation is a performance-critical step in the evaluation of the total energy of the system. With increasing size of the simulated system, their computation becomes extremely expensive. This issue becomes even more pressing in molecular simulations with explicit solvent. The high number of solvent atoms dramatically increases the computational cost. To decrease the computational effort, a number of schemes to treat long-range interactions have been developed. For example, Sagui and Darden[29] or Sutmann et al.[30] have published overviews of these schemes, which include Ewald summation, particle mesh, multipole expansion, and truncation.

However, these schemes can introduce errors to the total energy of the system and the forces acting on each atom. As a result, the errors may lead to artifacts in the simulation. Truncation schemes are especially prone to this problem as shown by several recent studies.[143–147] However, Smith and Pettit also observed artifacts with Ewald schemes.[148] Their observed artifacts vanish if the size of the periodic system is increased or for high dielectric constants of the system. The latter condition may pose problems to the application of Ewald methods to simulations of biological membranes, because the lipid tail regions exhibit a very low permittivity (see Section 2.1). Furthermore, Piana et al. reported that the truncation of Lennard-Jones interactions in biomolecular simulations could also cause artifacts.[149]

To avoid these issues, I will not use any truncation schemes. Due to my employed implicit solvent representation, the number of atoms in the simulated system reduces significantly. Therefore, I expect the direct evaluation of the $N^2$ terms to be very efficient anyway. The number of interaction energies that have to be computed can be further reduced by noting that Equations (2.2) and (2.5) are symmetric under the exchange of atoms $i$ and $j$. This reduces the number of interactions that have to be computed by a factor of two.

The first problem in the efficient implementation of these terms is the data layout in computer memory of the coordinates and forcefield parameters. The architecture of modern CPUs dictates the answer to this problem. They achieve their high performance of floating point computations by using vector instructions. Vector instructions are instructions to the CPU that perform the same operation, e.g. a multiplication or an addition, on multiple data items such as floating point numbers. For more details on these instructions, see the Appendix A.2. By using these vector instructions, the CPU can perform an operation on two, four or eight floating point numbers instead of just one. Ideally, the performance will increase up to a factor of eight. However, these instructions require that the respective data items be arranged in a specific way in the memory.

This requirement defines the data layout for the computation of the non-bonded interactions. Three separate blocks of continuous memory store all x, y, and z coordinates respectively. Storing the x, y, and z coordinates of the first atom, then that of the second and so on in one continuous memory block will in general prevent the usage of vector instructions. The forcefield parameters $q_i$, $\sigma_i$ and $\epsilon_i$ of Equations (2.2) to (2.5) are also stored in such a fashion. I note that due to the nature of proteins being polypeptide chains, many of these atom type dependent parameters will be equal. To lower memory consumption, it was, and still is common in older simulation codes to store just the atom type as an array index. Then there are three smaller arrays of the length of the number of different atom types $N_{\text{type}}$ instead of $N$ to store the partial charges and Lennard-Jones parameters. However, such a memory layout will also in general prevent the usage of vector instructions, which would result in a decreased performance. In addition, memory is no more a sparse resource in modern computers, wherefore I will not use this memory layout.

Another important aspect is the implementation of the excluded interactions in Equations (2.2) and (2.5). Checking for every interaction if it is excluded and then skipping the computation is no option. First, this would again prevent the usage of vector instructions, because not all data items are treated equally. Second, this check also requires computation time and therefore decreases performance. Finally, it would create a branch in the execution of the program. A branch is a point in the execution of the program, where depending on the input value, different instructions will be executed afterwards. Such branches are very costly. The reason for that cost is that CPUs are able to start a new operation although the last operation is not yet finished. For example, an addition or multiplication of two floating point numbers takes three or five CPU cycles respectively for the result to be available with the SSE instruction set. However, the CPU may start an addition or multiplication each cycle if the operands are available.[150] A branch in the program execution prevents such overlapping instructions. In conclusion, compute intensive parts of the program should avoid these branches for good performance.

My solution for the excluded interactions is as follows. Temporary arrays store all the interaction energies of an atom $i$ with all other atoms $j > i$. Afterwards, the entries in the temporary arrays corresponding to excluded interactions are set to zero. In addition, the entries corresponding to the 1-4 interactions in Equation (2.6) are scaled with the appropriate factors. Subsequently, the entries in the temporary arrays are converted to double precision floating point numbers and summed. This gives the interaction energy of an atom $i$ with all other atoms $j > i$. The last step sums all per-atom energies in double precision.

A last important point for the implementation to consider is if two atoms have zero distance. In that case, the energies cannot be evaluated, because division by zero is not allowed. For floating point values, this would result in not a number (NAN). The occurrence of NAN in computations also dramatically decreases performance. To avoid this performance penalty, I modified the calculation of the distance $r_{ij}$ between all atom pairs

$$r_{ij}^2 = \left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|^2 + r_0^2. \tag{3.1}$$

Here, $r_0^2 = 0.000001$ Å is a small arbitrary constant. Given the usual distance between nearby atoms of a few Angstrom in native protein structures, this change should not modify the low energy region of the potential. However if two atoms are closer than their Lennard-Jones radii, the potential will be modified due to the very rapid variation of the $r^{12}$ term in the Lennard-Jones energy of Equation (2.2). I have considered all these facts, when implementing the Coulomb and Lennard-Jones interactions into my new single energy term in SIMONA, which I will refer to as Nonbonded.

Figure 3.2 shows a comparison of the accuracy for the Coulomb and Lennard-Jones energies of my implementation in SIMONA and that of GROMACS. I have used the same set of 611 protein structures as for the dihedral potential comparison. For the Lennard-Jones comparison, the structures were subject to an energy minimization before the comparison, which enables a comparison of the relative errors for the low energy regions of the potential.

The average relative error for the comparison of the Coulomb energies is $1.5 \cdot 10^{-6}$ and the maximum relative error is $6.1 \cdot 10^{-6}$. These results are within the expected range of the floating point precision. For the Lennard-Jones potential, the average relative error is $2.5 \cdot 10^{-6}$, which slightly larger than for the Coulomb energies. However, there is one large outlier. The maximum relative error between Lennard-Jones energies is $4.6 \cdot 10^{-4}$ (not visible in Figure 3.2). The reason for this large error is the previously described modification of the distance computation in Equation (3.1). The relatively short energy minimization procedure was not able to remove all overlaps between atoms, wherefore this example demonstrates the expected deviations at low interatomic distances. The second largest relative error is $1.0 \cdot 10^{-6}$, wherefore the average relative error is mainly because of that one large outlier. Altogether, I observe good agreement between the energies computed with my Nonbonded term in SIMONA and the corresponding energies computed with GROMACS.

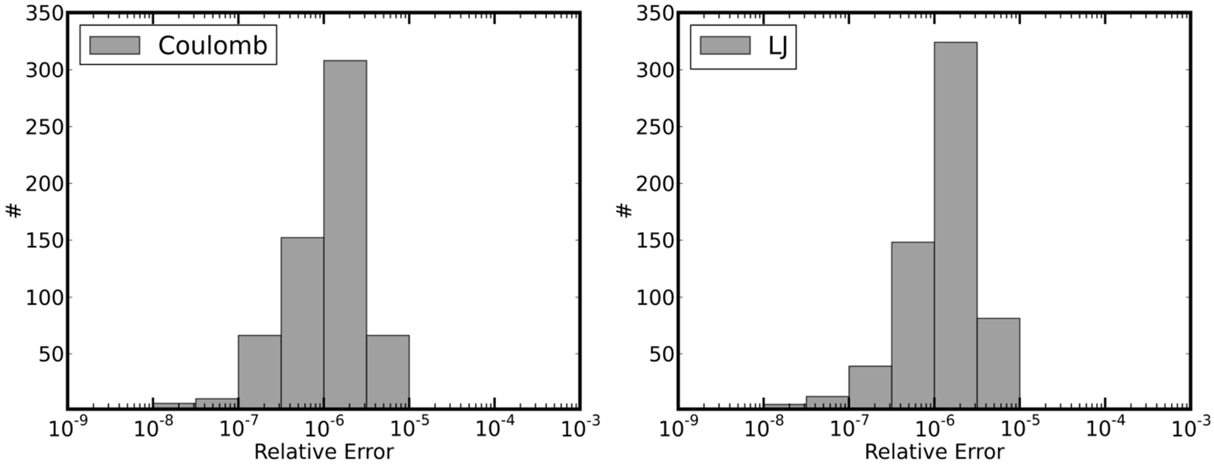


***Figure 3.2. Histograms of relative errors for Coulomb (left panel) and Lennard-Jones (LJ, right panel) energies between the GROMACS and SIMONA implementations. The same set of 611 native protein structures as in Figure 3.1 was used. For the Lennard-Jones comparison, the structures were minimized energetically with GROMACS to enable a comparison of the low energy regions of the potential.***

The goal of my implementation of the Nonbonded term was not only to have an accurate implementation, but also a very efficient one. I have done a performance comparison of the required computation time between my Nonbonded term and the old Lennard-Jones and Coulomb terms previously present in SIMONA. Figure 3.3 shows the computation time for each of the three potential terms during short Monte Carlo simulations of 10,000 steps. I used a small set of twelve native protein structures ranging from 267 to 5164 atoms.

The combined computation time for the old Coulomb and Lennard-Jones terms ranges from 134 s up to 52,843 s for the largest protein. The latter is equivalent to 14 hours and 41 minutes. In contrast, the computation time for my Nonbonded term ranges from 4.3 s to only 1,315 s. Thus, the largest system requires only a computation time of 22 minutes instead of more than 14 hours. The speedup in computation time by using my new implementation increases from a factor of 31.4 for the smallest protein up to 40.2 for the largest. This speedup demonstrates the increased computational efficiency of my new Nonbonded term.

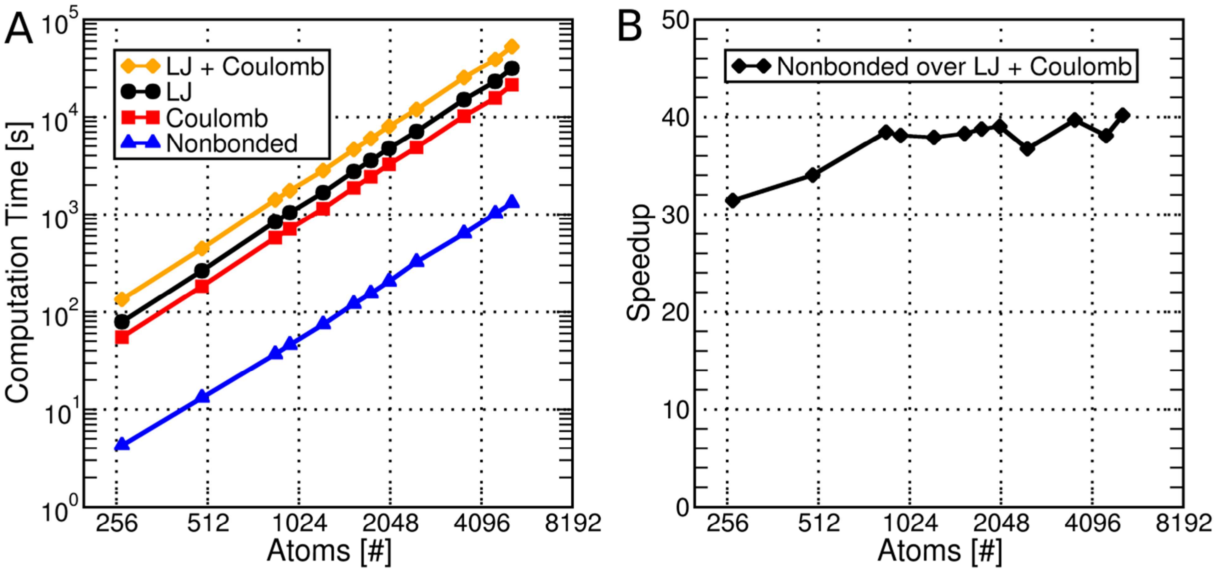


***Figure 3.3. Comparison of computation time as a function of the number of atoms in the system for old Lennard-Jones (LJ) and Coulomb potentials in SIMONA and my new implementation of these two terms, which is labeled as Nonbonded (panel A). The computation time was measured during a short 10,000 step Monte Carlo simulation. The speedup in computation time achieved by using my Nonbonded term instead of the old Lennard-Jones and Coulomb terms is also graphed (panel B).***

Another feature of modern CPU chips is that they consist of multiple CPU cores. These cores may process data in parallel. To make use of this feature, I have parallelized my Nonbonded term by using the OpenMP standard.[151] The OpenMP standard provides access to threads. These threads can process data in parallel while running on different CPU cores. To achieve the parallelization, the work of computing the Nonbonded term is split into small work packages. The available threads then process these work packages independently.

An important requirement to guarantee reproducibility of the simulation results is that the computed energies must not depend on the number of available threads or the scheduling of the work packages to the different threads. The reason is that the finite precision of floating point numbers invalidates the associative property of adding real numbers. Taking this into account, a single work package consists of computing the interaction of a single atom $i$ with all other atoms $j > i$, storing the computed energies in the temporary arrays, setting excluded and 1-4 interactions and summing the temporary arrays. Since the amount of work in one package depends on the index $i$, the work packages contain a varying amount of work, wherefore they are scheduled dynamically to the threads. This means that each thread may request a new work package after it has finished its previous one.

I have tested the implementation, and the energies obtained are binary invariant under the number of available threads in all cases. Subsequently, I have carried out speedup measurements, where I ran the same simulation with an increasing number of available threads. I used five proteins with an increasing number of atoms for the measurements. I have measured the computation time $t$ for the Nonbonded term during a 10,000 step Monte Carlo simulation. More details of these measurements are described in the Appendix A.3. The speedup $s_n$ by using $n$ threads is given by the computation time $t_1$ for using one thread and $t_n$ for using ten threads

$$s_n = \frac{t_1}{t_n}. \tag{3.2}$$

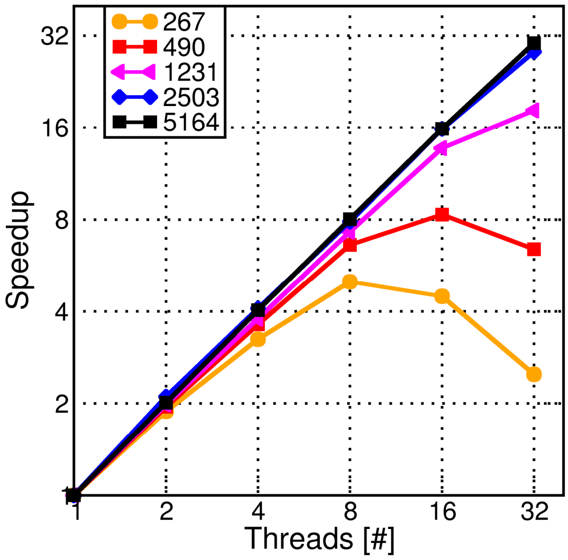


*Figure 3.4. Speedup in computation time of the Nonbonded term over the number of available threads for five proteins with an increasing number of atoms.*

The obtained speedups in Figure 3.4 show that for the smallest system they increase up to eight threads and then start to decrease again. The reason is that the work packages are too small. As a result, the available threads block each other while waiting for a new work package being assigned to them. This effect vanishes for larger systems. The implementation parallelizes well for a system of 1231 atoms up to 16 threads, reaching a speedup of 13.7. Even larger system scale well up to 32 threads with a speedup of 28.3 and 30.3 for 2503 and 5164 atoms respectively.

In summary, I have created an implementation of the AMBER99SB*-ILDN forcefield in SIMONA suitable for use in Monte Carlo simulations. Therefore, a number of modifications outlined above were necessary to guarantee proper behavior of the system even in edge cases. My implementation can compute total energies of that forcefield with sufficient numerical precision, high efficiency and good scaling behavior for multiple available threads.

## *3.2 Parallel Computation of the Solvent Accessible Surface Area*

As described in Section 2.4, I will model nonpolar solvation effects via a solvent accessible surface area (SASA) term. Each atom $i$ is assigned a sphere of radius $r_i$. The SASA $A_i$ of that atom is the surface area of the respective sphere not covered by the spheres of any other atoms. The method I will employ is based on the work of Connolly[76] and an implementation of Klenin et al.[152,153] This method, called PowerSASA, estimates the SASA based on analytical formulas. These formulas can be evaluated for each atom separately and their computation is therefore trivial to parallelize.

However, the evaluation of these formulas requires the knowledge of the so-called surface vertices for each atom. These surface vertices are points where the spheres of three different atoms intersect and those intersection points are not within any other sphere of an atom. A power diagram can yield these points.[152] Unfortunately, the algorithm to construct the power diagram proposed by Klenin et al. is inherently serial. Given that a power diagram for a set of $n$ spheres exists, they describe how the power diagram for $n + 1$ spheres can be constructed.[152]

This non-parallelizable algorithm poses a problem. According to Amdahl's law, the maximum speedup $s_n^{\max}$ of a computation parallelized with $n$ processes or threads is[154]

$$s_n^{\max} = \left(f_{\mathrm{s}} + \frac{1 - f_{\mathrm{s}}}{n}\right)^{-1}. \tag{3.3}$$

Here, $f_{\mathrm{s}}$ is the fraction of serial computation time. Considering that my Nonbonded interactions scale well up to 32 threads or processes as shown in Section 3.1, even a non-parallel fraction of 5%

in the computation would limit the maximum speedup to 12.5. As a result, a lot of the parallelization capability of my Nonbonded term would be wasted. Moreover, this performance bottleneck would strongly limit the application of the Monte Carlo simulations to investigate biomolecular processes, because the amount of sampling performed on the process would reduce significantly. To remove this bottleneck, I have also developed and implemented a parallel algorithm to construct the power diagram.

A power diagram consists of the power cells $\Pi$ belonging to each atomic sphere. The power cell $\Pi_i$ of a sphere $i$ at position $\boldsymbol{x}_i$ consists of all points $\boldsymbol{p}_i$ within a cubic bounding box for which the following condition is true

$$|\boldsymbol{p}_i - \boldsymbol{x}_i|^2 - r_i^2 < \left|\boldsymbol{p}_j - \boldsymbol{x}_j\right|^2 - r_j^2, \forall j \neq i. \tag{3.4}$$

By definition, the power cell has a convex shape. Its boundary consists of planar polygons. The corners of these polygons are termed vertices. These vertices are not to be confused with the surface vertices. The surface vertices of a sphere $i$ required for the computation of the SASA $A_i$ are the intersections of the edges of the power cell $\Pi_i$ with the sphere $i$. To enable parallelization, my algorithm computes these power cells independently of each other instead of computing the whole power diagram.

To construct a single power cell for a given sphere $i$, all other spheres possibly intersecting with that sphere have to be determined. Therefore, I have implemented a neighbor search method. The space is separated into cubes with edge length

$$s_\mathrm{c} = 2r_\mathrm{max}, \tag{3.5}$$

where $r_\mathrm{max}$ is the largest radius of all spheres. Subsequently, all spheres at positions $\boldsymbol{x}_i$ are sorted into these cubes. All other spheres possibly intersecting with sphere $i$ have to be either located in the same cube or any of the 26 neighboring cubes. This method is similar to that by Onderik.[155]

After all neighbors have been resolved, the algorithm starts by constructing a bounding cube around the sphere $i$. This cube is the preliminary power cell that is established by having six additional spheres with zero radii located in each direction along the axes of the coordinate system at distances $r_i$. Subsequently, the algorithm constructs the final power cell by iteratively adding all other possibly intersecting spheres $j$ to the power cell $\Pi_i$. Let us consider that we have a power cell $\Pi_i(j-1)$ where all $j-1$ possibly intersecting spheres have already been added. Now we want to add the $j$-th sphere to this power cell.

For all vertices of the current power cell $\Pi_i(j-1)$, the condition in Equation (3.4) is tested. If all of the vertices fail this test, the corresponding sphere $i$ is completely covered by other spheres. Thus, the SASA is zero and the algorithm continues with the next power cell. If not all, but one or more of the vertices fail this test, the intersection plane between spheres $i$ and $j$ is determined. This plane is orthogonal to the line connecting $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The plane includes the point $\boldsymbol{c}_{ij}$, which lies on the connecting line and fulfills the condition

$$\left|\boldsymbol{c}_{ij}-\boldsymbol{x}_i\right|^2-r_i^2=\left|\boldsymbol{c}_{ij}-\boldsymbol{x}_j\right|^2-r_j^2. \tag{3.6}$$

The intersecting plane cuts through the power cell $\Pi_i(j-1)$ and separates it into two regions. For one region, the condition in Equation (3.4) is true and for the other the condition is false. The former region is the new power cell $\Pi_i(j)$. The vertices of $\Pi_i(j)$ are all vertices of $\Pi_i(j-1)$ which passed the test of the condition in Equation (3.4). In addition, new vertices are those points where the intersection plane between spheres $i$ and $j$ intersected with the edges of the power cell $\Pi_i(j-1)$. These points can be computed easily from the intersection of a line and a plane.

To complete the construction of the new power cell $\Pi_i(j)$, the determination of its edges remains. The question is how to connect the new vertices to those that remained from power cell $\Pi_i(j-1)$ and how to connect the new vertices among each other to form the new edges. To solve this task, let me note a few things about vertices and edges of a single power cell. One vertex is part of exactly three intersection planes. Because each intersection plane is generated by an intersecting sphere $j$, each vertex can be uniquely labeled by the indices of these three spheres. I will denote these three spheres as the generators of the corresponding vertex.

Furthermore, each possible pair out of the three intersection planes of one vertex forms an edge of a single power cell. In turn, each of these edges can be labeled by one pair of generators corresponding to two intersection planes. It also follows that each vertex is part of exactly three edges of a power cell. As a result, the vertices of a single power cell form a ternary net. Consequently, an edge between two vertices exists, if and only if the two vertices have two common generators. This is the criterion how to connect the vertices to form a convex power cell.

After this procedure has been repeated for all intersecting spheres $j$, the construction of the power cell is complete. The computation of the surface vertices and the SASA is then analog to that of Klenin et al.[152,153] Figure 3.5 summarizes the algorithm to construct a single power cell for the two-dimensional case.

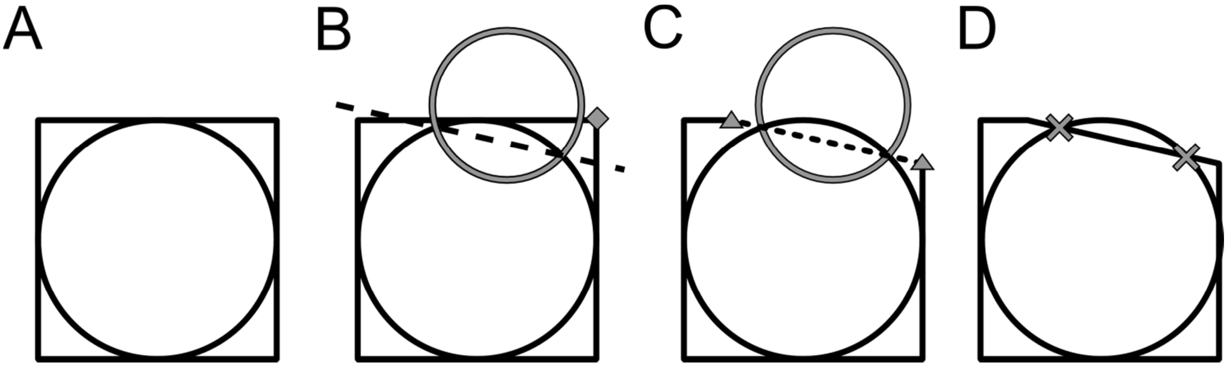


*Figure 3.5. Sketch of the algorithm to compute the surface vertices via a power diagram for the two-dimensional case. At first, a cubic bounding box around the sphere of interest forms the preliminary power cell (panel A). An intersecting sphere (grey circle) is added to the power cell by finding any vertices that do not satisfy the condition in Equation (3.4) (grey diamond) and computing the intersection plane (dashed line) defined by Equation (3.6) (panel B). The power cell is reduced by the region cut away due to the intersection plane and the new vertices (grey triangles), and edges (dotted line) are computed (panel C). When all spheres have been added, the surface vertices (grey crosses) are computed as the intersections of the final power cell with the sphere of interest.*

My implementation also takes care of numerical instabilities described by Klenin et al.[153] To assess the accuracy and stability of my new algorithm, I have run a test simulation. This simulation contained a protein with 1231 atoms and 10 million Monte Carlo steps. For each proposed configuration of a Monte Carlo step, I have compared the computed SASAs of each atom between my implementation and PowerSASA of Klenin et. al.[152,153] Figure 3.6 shows histograms of the resulting root mean square and maximum difference between the two sets of atomic SASAs from the two methods. As the histograms show, the two methods agree very well in most cases. However, in 1166 cases the maximum difference is larger than 1.0 Å$^2$.

I have investigated these cases in more detail. Therefore, I have also computed the SASA by a robust but computationally expensive numerical surface integration scheme for the atoms that showed the maximum SASA difference. The Appendix A.4 contains details about the numerical integration scheme. Comparison of the SASAs by PowerSASA and my parallel implementation to the results of the numerical integration scheme shown in Figure 3.7 reveals that, in 1163 cases, my results are closer to that of the numerical integration scheme, while only in three cases the PowerSASA results are closer to the numerical results. In addition, for the 1163 cases the SASA errors of my implementation in relation to the numeric integration scheme are also smaller than 0.1 Å$^2$. For the remaining three cases, the errors are smaller than 1.0 Å$^2$. This demonstrates the good numerical stability of my implementation. The reason for this improved numerical stability is that my algorithm does not need to construct a complete self-consistent power diagram. Only each single power cell needs to be consistent. This is much easier to achieve.

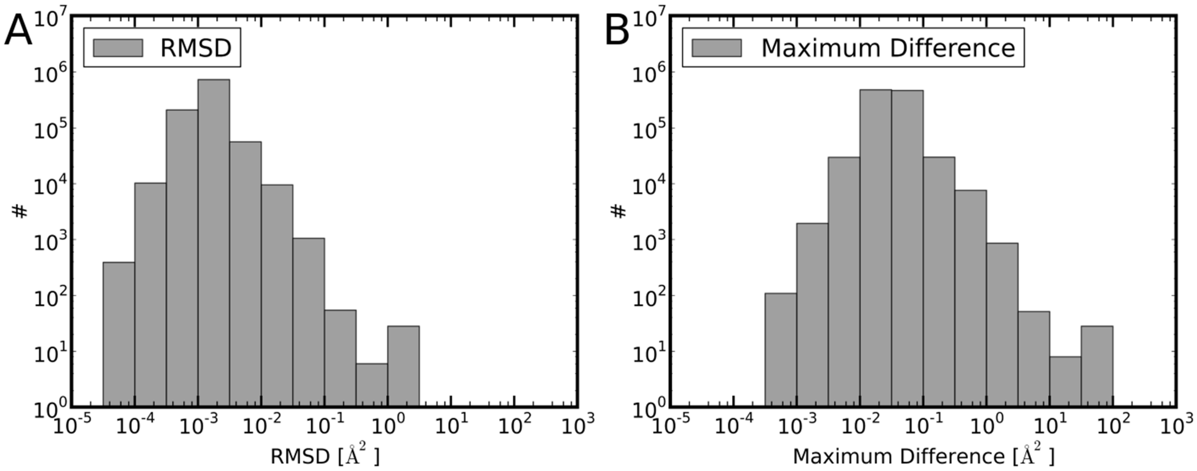


***Figure 3.6. Differences of the computed solvent accessible surface areas (SASA) for each atom with the PowerSASA[152,153] method and my new parallel implementation during a ten-million-step Monte Carlo simulation. Panel A shows a histogram of the root mean square differences (RMSD) between the atomic SASA of each method for each Monte Carlo step. Panel B shows a histogram with the maximum difference of the atomic SASA between the two methods for each Monte Carlo step.***

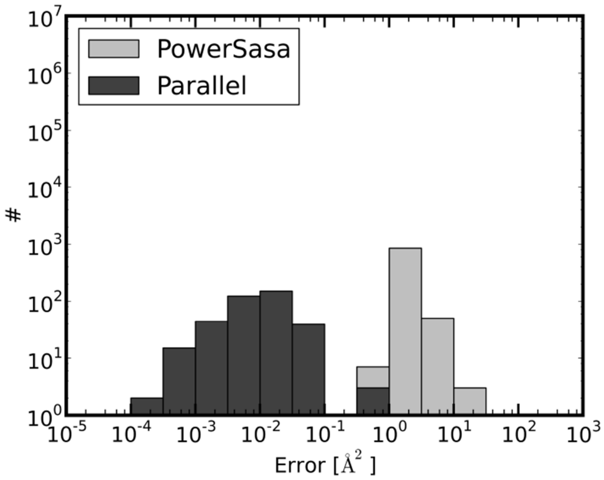


***Figure 3.7. Histogram of the SASA errors of PowerSASA[152,153] and my parallel SASA implementation relative to a robust but computationally expensive numerical SASA integration scheme. The data set contains the 1166 cases of Figure 3.6 where the computed SASA between PowerSASA and my parallel implementation differed by more than 1.0*** Å.

Since my algorithm computes each power cell independently of each other, there is a drawback to my algorithm. Each surface vertex is common to three spheres, because it marks the point where these three spheres intersect. Therefore, my algorithm introduces additional workload, since it requires that every surface vertex have to be computed three times. The timing measurements in Figure 3.8 on a small set of protein structures show that the overhead introduced by the redundant computations increases the computation time by a factor of less than 2.3. This factor decreases down to 1.84 for the largest measured protein. Overall, the increase in computation time due to redundancies is lower than the expected factor of 3.0. However, the potential parallel execution will compensate this drawback. Since this implementation is part of the PowerBorn algorithm described in the next section, I will postpone the speedup measurements to that section.

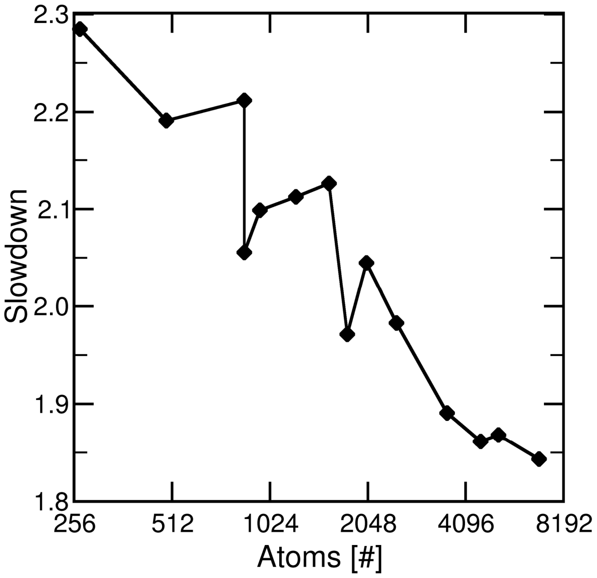


***Figure 3.8. Slowdown in computation time in part due to redundant calculations in the parallel implementation of the SASA computation in comparison to the PowerSASA[152,153] method when using only one thread.***

## 3.3 An Accurate and Efficient Generalized Born Model

*Reproduced in part with permission from Brieg, M.; Wenzel, W. PowerBorn: A Barnes–Hut Tree Implementation for Accurate and Efficient Born Radii Computation. J. Chem. Theory Comput. 2013, 9, 1489–1498. Copyright 2014 American Chemical Society.*

As explained in Section 2.4, I will use the generalized Born model to describe electrostatic solvation effects. Onufriev et al. showed that the accuracy of the generalized Born model in relation to Poisson-Boltzmann methods does strongly depend on having very accurate estimates of the Born radii $R_i$ used in Equation (2.35).[101] According to Mongan et al.,[102] this requires the use of the more elaborate integral expressions of Lee et al.[106,107] or Grycuk.[108] In order to also achieve good agreement between hybrid or explicit water calculations and generalized Born methods,[156,157] one has to use the solvent excluded surface (see Section 2.3).

The problem is that currently available methods and implementations to compute Born radii are unable to comply with all these requirements at a reasonable computational cost. They are either based on the outdated Coulomb field approach,[105,120,121,123,131,158–168] or use the problematic van der Waals surface or another approximate surface.[106,122,169–171] Other methods sacrifice accuracy for having smooth derivatives required for molecular dynamics.[172] Finally, some are just too slow for efficient biomolecular simulations.[107] Moreover, several generalized Born models approximate the Born radii integrals based on a pairwise descreening method.[158,159] This method relies on the fact that the van der Waals spheres of the atoms do not overlap very much. This may be the case for conformations during molecular dynamics simulations, where the Lennard-Jones potential of Equation (2.2) prevents such conformations. However, this may not be true for perturbed configurations in a Monte Carlo simulation.

Based on these facts, I decided to develop a new algorithm that combines the accuracy of Grycuk's R6 integral expression of Equation (2.40) with the solvent excluded surface, and an efficient numerical implementation, that is suited for application in biomolecular Monte Carlo simulations. Together with Wolfgang Wenzel, I have published this algorithm under the name PowerBorn.[173]

### *Algorithm for the Computation of Born Radii*

The PowerBorn algorithm to compute Born radii exploits the fact that the integrand in Equation (2.40) is rather simple while the integration region is very complex. The reason is the complex geometry of the employed solvent excluded surface definition described in Section 2.3. To ease this problem, I split the integral into two parts. Outside a bounding box around the molecule, PowerBorn uses analytical formulas to evaluate the integral of Equation (2.40).[173] This is possible due to the simple geometry of the bounding box. Inside the bounding box, I use an efficient numerical integration procedure. This numerical procedure exploits the fact that the integrand in Equation (2.40) decrease very rapidly with the distance to the atom in question.

PowerBorn employs an octree method based on the proposal of Barnes and Hut.[174] The space inside the bounding box is recursively separated into nested cubes of decreasing size. This tree structure of nested cubes is an octree. The numerical integration procedure first computes those cubes that are outside the solvent excluded surface in a recursive manner. It separates the bounding box into eight equal sized cubes. If any of the cubes is not completely inside or outside the solvent excluded surface, the algorithm again separates that cube into eight equal sized cubes. This procedure is recursively continued. The procedure stops when the cube size reaches a defined minimal size. These smallest cubes are either completely inside or outside the solvent excluded surface. In the former case, the center of the cube must be inside the solvent excluded surface. Otherwise, such a cube is considered completely outside the solvent excluded surface.

For this decision, the PowerBorn algorithm has to approximate the solvent excluded surface efficiently. It uses a finite number of approximate equidistant spaced sampling points on the solvent accessible surface (see Section 2.3). Subsequently, it places so-called water spheres with the radius of the probe radius onto these points. A given point is inside the approximate solvent excluded surface if this point is inside any of the spheres of the solvent accessible surface and not inside any water sphere. For an infinite density of water spheres, this approximation converges to the solvent excluded surface. For a finite number of water spheres placed at distances smaller than the probe radius, this algorithm will provide a sufficient and efficient approximation to the volume not enclosed by the solvent excluded surface.

To decrease the number of necessary sampling points, I reuse the power diagram representation[152,153] from the computation of the solvent accessible surface area. I will place water spheres at each surface vertex computed from the power diagram. This increases the accuracy of the approximated solvent excluded surface.[173] My parallel implementation of the power diagram described in Section 3.2 is also suitable for this application.

Subsequently, PowerBorn calculates the volume $V_k^{\text{water}}$ outside the solvent excluded surface and inside a cube $k$, as well as the centroid of that volume $\boldsymbol{c}_k$. For any cube that is completely outside the solvent excluded surface, the volume $V_k^{\text{water}}$ is equivalent to the volume of the cube and the centroid is equal to the center of the cube. If the cube is completely inside the surface, $V_k^{\text{water}}$ is zero. For any cubes that have smaller nested cubes, $V_k^{\text{water}}$ is given by the sum of the volumes of the smaller nested cubes

$$V_k^{\text{water}} = \sum_{l=1}^{8} V_l^{\text{water}}. \tag{3.7}$$

Here, the index $l$ iterates over all eight cubes nested inside cube $k$. The corresponding centroid is the sum of the centroids of the nested cubes weighted by the fraction of corresponding volume of water

$$\boldsymbol{c}_k = \frac{1}{V_k^{\text{water}}} \sum_{l=1}^{8} \boldsymbol{c}_l V_l^{\text{water}}. \tag{3.8}$$

Subsequently, the PowerBorn algorithm uses these data to perform the numerical integration of Equation (2.40) inside the bounding box. If a given cube $k$ located at centroid $\boldsymbol{c}_k$ with edge length $s_k$ is sufficiently small or far away from an atom $i$ at position $\boldsymbol{x}_i$, it will fulfill the condition

$$|\boldsymbol{x}_i - \boldsymbol{c}_k|^2 < \frac{s_k^2 f}{4}. \tag{3.9}$$

Here, $f$ is the so-called integration factor that defines what is meant by sufficiently small or far away. If this condition holds, the integral of Equation (2.40) over the cube $k$ is approximately

$$\int_{\text{cube } k} \frac{d^3x}{|\boldsymbol{x} - \boldsymbol{x}_i|^6} \approx \frac{V_k^{\text{water}}}{|\boldsymbol{c}_k - \boldsymbol{x}_i|^6}. \tag{3.10}$$

Taylor expansion of the integrand to zeroth order and then performing the integral yields this approximation.

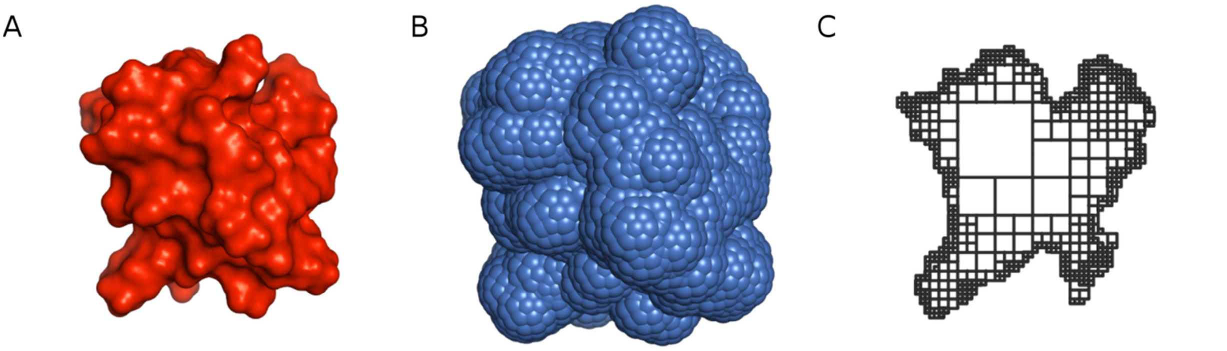


***Figure 3.9. Solvent excluded surface for a protein (panel A). Water spheres located on the solvent accessible surface that are used to approximate the solvent excluded surface (panel B). Slice of the octree structure showing only cubes that are located inside the solvent excluded surface (panel C).[173]***

To find the cubes that fulfill the condition in Equation (3.9), PowerBorn performs a recursive walk through the cubes of the octree, starting with the largest cube. Whenever the visited cube fulfills the condition of Equation (3.9), the contribution in Equation (3.10) is added to the Born radius integral of Equation (2.40). Otherwise, PowerBorn proceeds with the eight smaller nested cubes and tests them until it finds suitable cubes. If a cube does not fulfill Equation (3.9) and has no nested cubes either, a numerical grid integration is performed for that cube. For more details on the algorithm and its implementation, the reader may refer to Brieg and Wenzel.[173]

### *Accuracy Assessment of the Generalized Born Model*

To assess the accuracy of PowerBorn algorithm, I calculated reference Born radii for three native protein structures and compared them to the PowerBorn radii. The reference Born radius for an atom is given by setting all other atoms' partial charges to zero. Consequently, the electrostatic solvation free energy $\Delta G_{\text{elec}}$ will be equal to the self-polarization $\Delta G_{\text{self}}$, see Equations (2.17) and (2.36). With the help of numerical Poisson-Boltzmann solvers such as APBS,[175] $\Delta G_{\text{elec}}$ can be computed. Afterwards, Equation (2.36) can be solved for the reference Born radius.

I have compared these reference Born radii to those computed by the PowerBorn method. The comparison contains two different PowerBorn parameter sets, a more accurate one termed ACC, and a faster one termed FAST.[173] Figure 3.10 shows the results of this comparison. The comparison reveals a very high correlation between the reference Born radii and PowerBorn radii for the three protein structures and both parameter sets. However, the linear fit of the PowerBorn radii to the reference radii shows a systematic deviation.

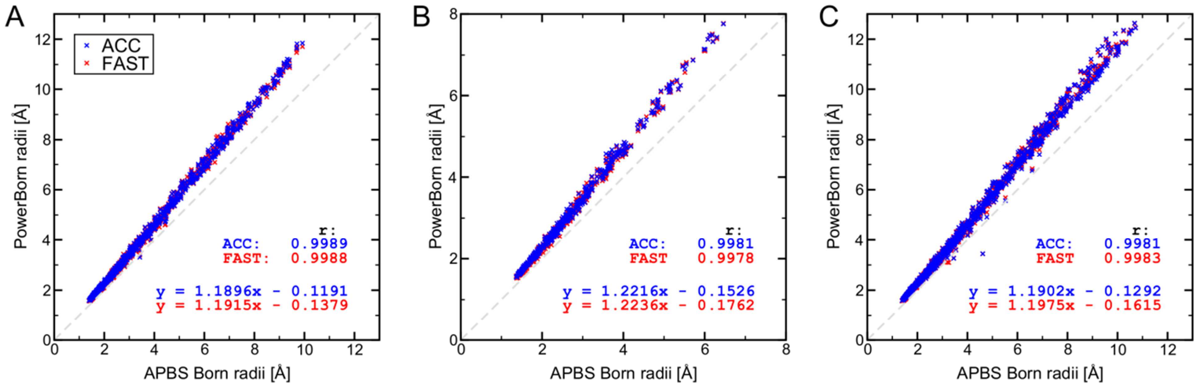


***Figure 3.10. Comparison of reference Born radii computed with the Poisson-Boltzmann solver APBS[175] to PowerBorn radii for three different protein structures. Two different PowerBorn parameter sets are used, ACC and FAST. The plots also show the Pearson correlation coefficient r and the linear fit of the PowerBorn radii to the APBS radii.[173]***

The source of this deviation is the approximation of the integral in Equation (3.10), which systematically underestimates the integral. This deviation can be corrected by using modified Born radii $\tilde{R}_i$[173]

$$\frac{1}{\tilde{R}_i} = \frac{a}{R_i} + b. \tag{3.11}$$

The free parameters $a$ and $b$ are fitted to reproduce electrostatic solvation free energies $\Delta G_{\text{elec}}$ of a training set of protein structures computed with the numerical Poisson-Boltzmann solver APBS.[175] I note that the reported values of $a$ and $b$ differ for the two PowerBorn parameter sets ACC and FAST.[173] Since the systematic deviation will depend on the integration factor $f$ of Equation (3.9), this behavior is to be expected. It also implies that changing any of the PowerBorn parameters will likely require a refitting the parameters $a$ and $b$ of Equation (3.11).

Furthermore, I observed that the non-vanishing parameter $b$ results in a decreased agreement between the corrected PowerBorn radii $\tilde{R}_i$ of Equation (3.11) and the reference Born radii.[173] Nevertheless, the agreement between electrostatic solvation free energies computed via Equation (2.35) and those computed with Poisson-Boltzmann calculations increases when Born radii corrected by Equation (3.11) are used. Figure 3.11A shows the relative errors between the electrostatic solvation free energies for a test set of 611 protein structures.

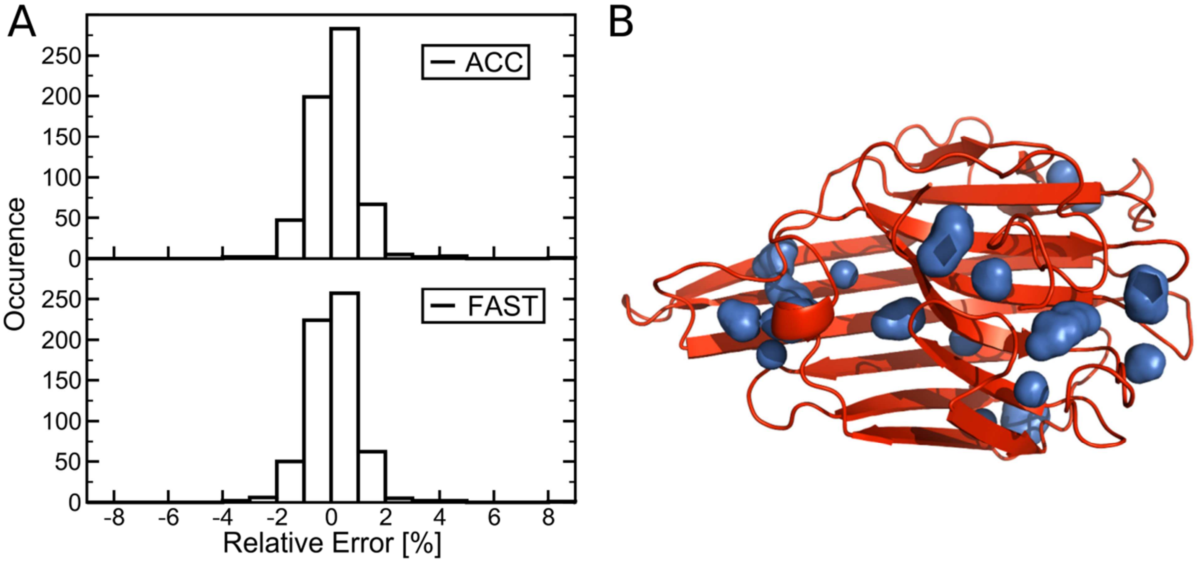


***Figure 3.11. Histogram of relative errors between solvation free energies computed with the generalized Born model using PowerBorn radii with parameter sets ACC and FAST, and the numerical Poisson-Boltzmann solver APBS[175] (panel A). Visualization of the structure with the largest relative error, PDB code 1NLS,[176] in the red cartoon representation with water cavities highlighted in blue (panel B).[173]***

The relative root mean square error is below 1%. However, there are a few outliers with errors of electrostatic solvation free energies of up to 8.2%. The reasons for the largest outlier are numerous water-filled cavities present in the PDB structure, as visualized in Figure 3.11B. The generalized Born model is known to show systematic deviations in such cases. Thus, the error is due to the generalized Born model itself, and not the PowerBorn method for the calculation of the Born radii.[173]

In conclusion, the low root mean square error of the electrostatic solvation free energies shows that the accuracy of the PowerBorn method is as good as the best other published method GBMV2.[107] My result also extends the findings of Onufriev et al. They demonstrated the importance of good agreement between estimated Born radii and reference Born radii for obtaining accurate electrostatic solvation free energies from the generalized Born model.[101] The results for the fit parameters $a$ and $b$ of Equation (3.11) show that even more accurate electrostatic solvation free energies can be achieved by using Born radii corrected by Equation (3.11).

Since the PowerBorn algorithm solves a part of the integral in Equation (2.40) by discretizing the space via the octree data structure, this scheme will introduce discretization errors. There are several reasons for these errors. The first reason is the finite size of a smallest octree cube. The second is the condition in Equation (3.9). The third reason is the approximation of the Born radii integral in Equation (3.10). The fourth reason is the finite number of water spheres used to approximate the solvent excluded surface. I estimated the relative root mean square discretization errors to be 0.11% and 0.15% of the electrostatic solvation free energy for the ACC and FAST parameter set respectively.[173] However, these discretization errors are averaged out when computing physical observables based on ensembles according to Equations (2.13) and (2.14).[173]

### *Performance Assessment of the Born Radii Computation*

To assess the performance of the PowerBorn algorithm, I have carried out computation time measurements. The details of these measurements are explained by Brieg and Wenzel.[173] As shown in Figure 3.12, my implementation of the PowerBorn algorithm performs much better than the GBMV2[107] method implemented in CHARMM.[65] It yields speedups in the range of 4.2 to 14.2, depending on the PowerBorn parameter set and the number of atoms in the system. In comparison to the GBOBC[120] provided by GROMACS,[25] which is based on the Coulomb field approximation, the PowerBorn method is slower for systems with approximately less than 1500 atoms, but much more accurate. For larger systems, PowerBorn outperforms the GBOBC method in terms of speed and accuracy.

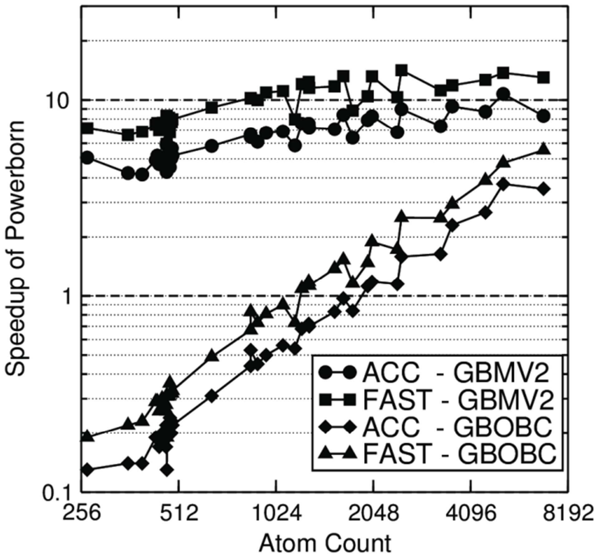


***Figure 3.12. Speedup of PowerBorn's ACC and FAST version in comparison to GBMV2[107] in CHARMM[65] and GBOBC[120] in GROMACS[25] for different sized protein structures.[173]***

### *Parallelization of the Born Radii Computation*

To exploit modern multicore CPU architectures and further enhance performance, I have also parallelized the PowerBorn method using the OpenMP standard.[151] Previous attempts with other parallelization methods showed no satisfying results.[177] Here I will outline the parallelization strategy for the PowerBorn method.

The generation of the sampling points on the solvent accessible surface used to approximate the solvent excluded surface can be done for each sphere of the solvent accessible surface independently. Hence, this step is trivial to parallelize. The same is true for the placing of the water spheres onto these sampling points. For each sphere of the solvent accessible surface, water spheres are placed at the corresponding surface vertices extracted from the power diagram. Therefore, my parallel version of the power diagram can be used (see Section 3.2). In the final step, all generated water spheres are combined in one set in serial.

I have parallelized the construction of the octree in the following way. The parallel algorithm separates the bounding box into $8^{N_{\text{level}}}$ equal sized cubes. Each of these cubes corresponds to a cube of the octree at level $N_{\text{level}}$. Subsequently, the octree can be constructed in parallel within each of these cubes. Since the actual workload for constructing an octree within such a cube can differ significantly, the cubes are scheduled dynamically to the available threads. In addition, a sufficient number of work packages should be present to allow for an efficient load balancing. However, if $N_{\text{level}}$ is too large, it decreases the efficiency of the PowerBorn algorithm, because it transforms the octree data structure into a grid data structure. In tests with a few native protein structures, $N_{\text{level}} = 3$ provided the best average performance. Thus, there are 512 work packages in the parallel algorithm. After the octrees for all these 512 cubes are finished, the algorithm combines them into a single octree in serial with very low computational effort.

In the final step, the algorithm carries out the numerical integration inside the bounding box with the help of the octree as previously explained. This can be done for each atom in parallel. The computation of the analytic formulas for the integral outside the bounding box and the conversion of the final integral value to the Born radius for every atom can be done in parallel too.

To demonstrate the performance increase due to the parallelization of the PowerBorn method, I have performed speedup measurements similar to those in Section 3.1. Because the PowerBorn method requires a power diagram for the construction of the octree, the speedup measurements include the construction of a parallel power diagram as well as the time to compute the solvent accessible surface area described in Section 3.2. The results are shown in Figure 3.13A. In contrast to the parallelization of the Nonbonded term in Section 3.1, the dependence of the achieved speedups on the protein's size is much smaller. For 16 available threads, the minimally achieved speedup is 11.9 for the smallest system and 14.0 for the largest. Only with 32 threads, the speedups start to show a higher dependence on the system size. The smallest system reaches 47% of the maximum expected speedup of 32, while largest system achieves 72%.

The reason for this behavior is the parallelization of the octree construction, which takes up most of the computation time. The number of work packages does not depend on the number of atoms in the system. Therefore, this parallelization scales reasonably well also for systems with few atoms. In addition, the work packages are on average larger, since the overall effort to compute the Born radii and solvent accessible surface areas is larger in comparison to computing the non-bonded interactions.

However, with an increasing number of atoms, the workload contained in one work package can differ significantly. Depending on the conformation of a protein, some of these cubes may not contain any protein at all, while others cubes may be filled with the protein. For a high number of threads, a balanced distribution of the work packages to the threads will be difficult. On average, each thread will only process few work packages. Those may all contain very low or high amount of workload. The different amount of work results in a high load imbalance and reduced parallel efficiency. To improve parallelization further, future efforts may introduce a better workload scheduling.

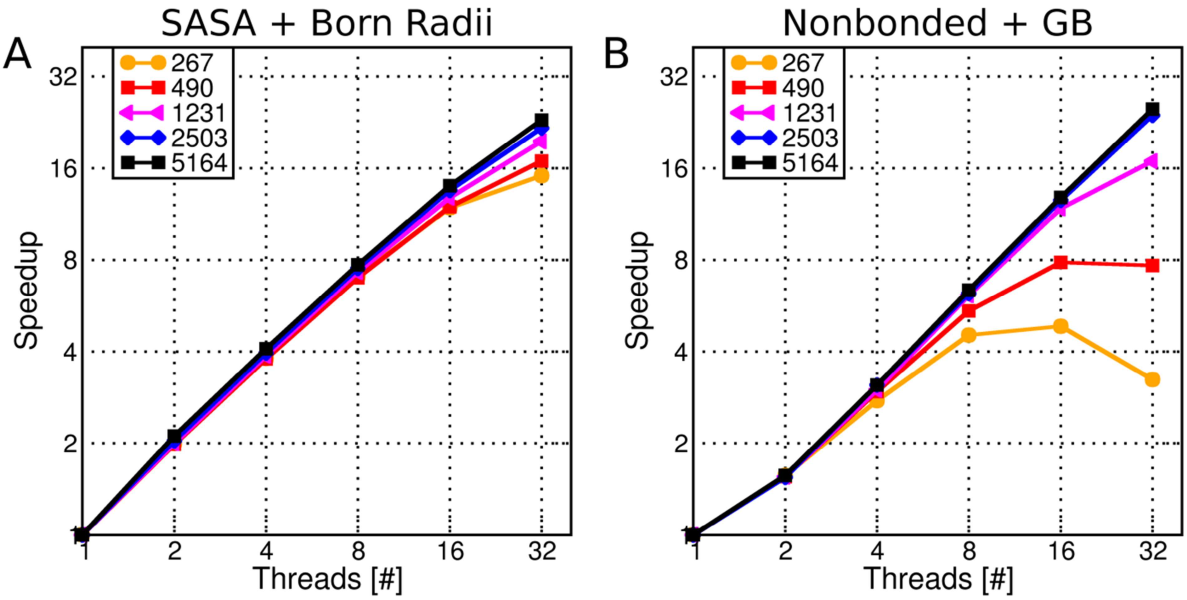


***Figure 3.13. Speedup in computation time of the solvent accessible surface area (SASA) and Born radii with the parallelized PowerBorn method over the number of available threads. Five proteins with an increasing number of atoms were used (panel A). Speedup in computation time for the Nonbonded and generalized Born (GB) term of Equation (2.35) over the number of available threads (panel B).***

### *Optimization and Parallelization of the Generalized Born Implementation*

To be able to compute electrostatic solvation free energies within the generalized Born model for Monte Carlo simulations, an efficient implementation of Equation (2.35) is necessary. Due to the similarity to the Coulomb energy in Equation (2.5), I implemented the generalized Born formula in the same efficient way as the Nonbonded term described in Section 3.1. Thus, the extended Nonbonded+GB term computes the Coulomb, Lennard-Jones, and generalized Born energies. To enable the use of vector instructions, a vectorizable version of the exponential function in Equation (2.37) is required. The Eigen library[178] provides such a function for the SSE vector instruction set. For the faster AVX vector instruction set, I have ported the Eigen version to the new instruction set.

Using this implementation, I have carried out computation time measurements in analogy to Section 3.1. The addition of the generalized Born formula to the Nonbonded term increases the computation time on average by 68% for the single threaded version. The speedups for the multithreaded version are shown in Figure 3.13B. Similar to the Nonbonded term only, the speedups for small systems saturate at a medium number of threads and then decrease again. This problem is not observed for larger systems with more atoms. In contrast to the Nonbonded term only, the maximal speedups for larger systems stay clearly below the maximum expected value of 32. It is interesting to observe that the speedup for two threads reaches only 75% to 80% of the possible speedup of 2.0. Increasing the number of threads for medium to large systems then yields similar fractions of the maximum expected speedup. The reason for this behavior is not understood yet.

Nevertheless, the presented results demonstrate that the PowerBorn algorithm yields accurate Born radii and solvent accessible surface areas with high efficiency compared to similar accurate methods and good scaling behavior. In addition, the optimized parallel implementation of the generalized Born formula allows the efficient computation of electrostatic solvation free energies without the requirement for approximate long-range interaction schemes.

## *3.4 Monte Carlo Simulation Performance*

Here I present an overview of the current performance of the SIMONA simulation package to simulate proteins in an aqueous environment or a biological membrane with my implemented methods. These data is necessary to estimate the resources required for future simulations of a given protein and environment. I emphasize that no cutoffs for long-range interactions such as the Coulomb, Lennard-Jones, or the generalized Born interaction are applied.

Let me focus on the aqueous environment first. Table 3.1 lists the forcefield terms necessary for such a simulation. To run such a simulation, one has to choose the number of available threads used to evaluate the energy. In general, the more threads are used, the faster the simulation will run, and the earlier the results will be available. However, doubling the number of threads will not always result in a simulation running two times faster as shown in Sections 3.1, 3.2, and 3.3. Therefore, the decision how many threads to use, has to weigh up the available amount of computer resources against the time required to complete the simulation. In general, an achieved speedup of 50% of the maximum obtainable speedup is not a worthwhile investment of computational resources. Therefore, the actual speedup of the simulation should not be below half the number of available threads. According to Figure 3.14A, this means that proteins with less than approximately 1250 atoms should not be simulated with more than 16 threads. Larger systems may use up to 32 threads. If the decision on the number of available threads is done, Figure 3.14B shows how many Monte Carlo steps per day the simulation will complete. I note that those numbers may vary depending on the employed hardware.

***Table 3.1. The forcefield terms are necessary for the simulation of proteins in an implicit water environment. SASA denotes the solvent accessible surface area.***

| No. | SIMONA Forcefield term | Description | References or Equations |
|---|---|---|---|
| **1** | Dihedral potential | proper and improper dihedral potentials | Equations (2.9) and (2.10) |
| **2** | Nonbonded | Lennard-Jones, Coulomb and 1-4 interactions plus generalized Born term | Equations (2.2)-(2.6), and (2.35) |
| **3** | PowerBorn | computation of SASA and Born radii | chapters 3.2 and 3.3 |
| **4** | NPSasaEnergy | nonpolar solvation free energy | Equation (2.41) |

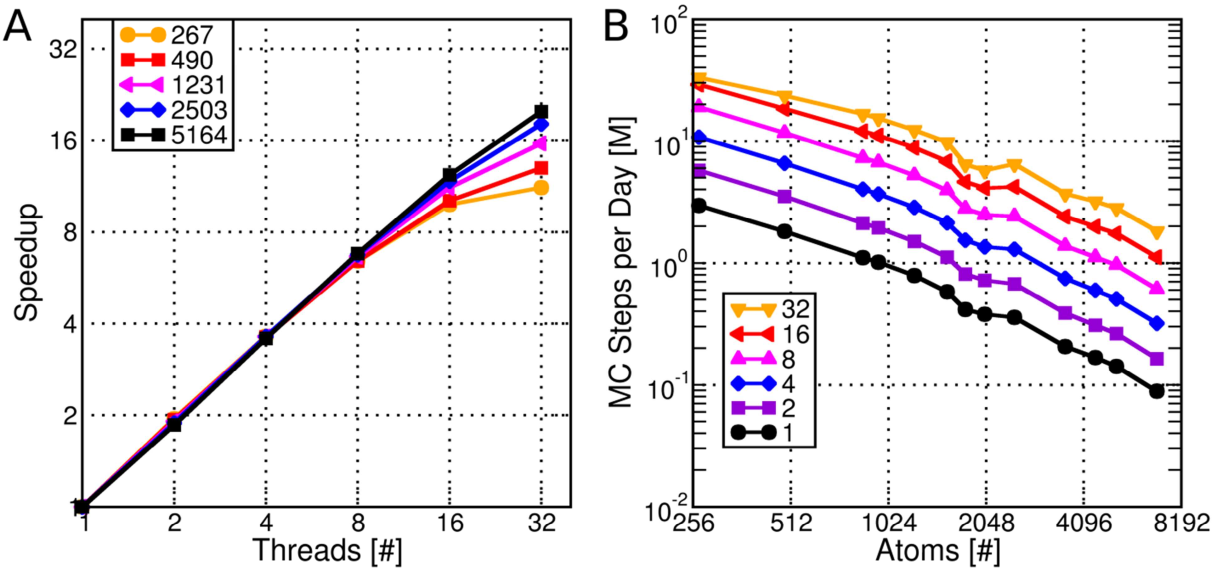


***Figure 3.14. Overall simulation performance measures for protein simulations in an aqueous environment. Panel A shows the speedup of the computation time over the number of threads for the complete Monte Carlo simulation for five different proteins with an increasing number of atoms. Panel B shows how many million Monte Carlo (MC) steps per day can be computed as a function of the number of atoms in the protein and the number of available threads.***

I have further analyzed how the computation time during a simulation is distributed between the different forcefield terms. As Figure 3.15 shows, the computation of the Born radii and solvent accessible surface areas takes up the largest fraction of the runtime. For the smallest proteins, this fraction is as large as 96%. When the size of the protein increases, the fraction of runtime spent in computing the non-bonded interactions and generalized Born formula of Equations (2.2) to (2.5) and (2.35) increases. The reasons is that these terms scale with$N^2$, while the solvent accessible surface area and Born radii computation scales approximately linearly for the considered proteins. Even for the largest protein with 5164 atoms, the fraction of runtime spent for the computation of non-bonded and generalized Born interactions is still smaller than that for the computation of the Born radii and solvent accessible surface area.

In addition, the Nonbonded energy term with the addition of the generalized Born energy parallelizes better with respect to the available threads (see Section 3.3). This better parallelization results in a decrease of the percentage of runtime spent in the Nonbonded and generalized Born term for larger proteins.

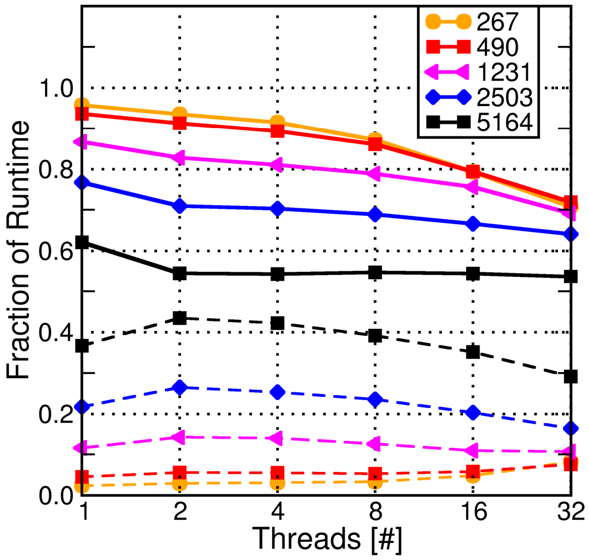


***Figure 3.15. Fraction of computation time for the two most compute intensive forcefield terms relative to the total computation time. The forcefield terms are the combined solvent accessible surface area and Born radii computation (solid lines), and the combined Nonbonded and generalized Born interactions (dashed lines). Five different proteins of increasing size were used for the measurements.***

This high number of Monte Carlo steps per day and the low fraction of computation time spent in the non-bonded interactions clearly demonstrate that it is not necessary to resort to any approximate long-range interaction schemes discussed in Section 3.1. As a result, the errors due to these schemes will not be present in SIMONA simulations using the forcefield terms implemented by me.

I also note that several other factors will influence the simulation performance. This is the capability of the hardware employed. Details about the hardware I employed can be found in Appendix A.3. Faster or slower hardware will likely influence the simulation performance. Another important aspect is the compactness of the simulated structure. The octree for the PowerBorn method described in Section 3.3 has to be constructed with high resolution near the surface of the protein.

Since this construction is one of the most time-consuming steps, it strongly affects the overall simulation performance. Therefore, the simulation of a protein will progress faster during a compact folded state than an extended unfolded state.

This also affects the performance of the parallel tempering method described in Section 2.5. Since the protein is more likely to unfold at high temperatures, the simulation will on average progress slower at high temperatures than at lower temperatures. The current implementation of the parallel tempering algorithm in SIMONA requires that all temperatures have to complete a given number of Monte Carlo steps before a temperatures exchange can happen. Therefore, all simulations have to wait for the slowest progressing simulation. The waiting time reduces the computational efficiency of the algorithm.

In summary, I have implemented the force field terms and implicit solvent models necessary for a Monte Carlo simulation in implicit solvent with common molecular force fields in an efficient manner into the SIMONA Monte Carlo simulation framework. None of my implemented force field terms relies on a special scheme to treat long-range interaction, wherefore the Monte Carlo simulations are not prone to artifacts caused by such schemes as discussed in Section 3.1. The parallelization of my implemented methods increases the simulation performance significantly, allowing for better sampling of the investigated processes or the study of larger systems.

# 4 Improving Small Molecule Hydration Free Energies Estimates of Implicit Solvent Models

In the previous chapter, I introduced the methods necessary for the efficient simulation of proteins in an implicit water model using an accurate generalized Born model and a solvent accessible surface area term. As stated in Section 2.4, the solvent accessible surface term may have deficiencies in modeling nonpolar solvation effects. Thus, extended models of nonpolar solvation effects have to be investigated to improve the accuracy of approximate implicit solvent models. As a start, I have carried out an assessment of three different implicit solvent models in cooperation with Julia Setzler and Wolfgang Wenzel.[179] The first section of this chapter presents some background information that sets the results of this assessment into context with prior work on this subject and motivates our approach that enables a fair comparison of the models. The second section introduces the investigated models. The third section explains how I parameterized them to enable a fair comparison of them. The fourth section reviews our achieved results by comparing computed hydration free energies for small neutral molecules from a large database to experimental data. In the last section, I present my analysis of the hydration free energy data based on the classification of the molecules in the database into chemical groups.

## 4.1 Background and Motivation

As explained in Section 2.4, the hydration free energy is the free energy difference between gaseous and solvated states. Recent advances in simulation techniques and computational resources allow the determination of these free energy differences with very low statistical uncertainties from computer simulations of small neutral chemical compounds.[82] The high-throughput determination of these free energy differences is of high relevance to pharmaceutical research. The small chemical compounds may be drug candidates that are supposed to bind to target proteins. The prediction of the binding affinities is the goal of computational methods of drug discovery.[180] These methods scan large databases of compounds to identify possible drug candidates which show a high predicted binding affinity for a given target protein.[181] Since the binding affinity depends on the free energy difference between the bound state and the solvated state, errors in describing the solvated state affect the binding affinity prediction.[182] Thus, accurate solvent models are an essential requirement of methods for computational drug discovery. Due to the large size of the compound databases, these solvent models should also be computationally efficient, wherefore implicit solvent models promise candidates for this task.

Unfortunately, a recent study on a large set of small neutral molecules showed that the estimated hydration free energies of many common implicit solvent models are less accurate than the estimates of the explicit TIP3P water model.[183] This poses a large problem to the application of implicit solvent models to the prediction of binding affinities. The authors that more elaborate nonpolar contributions to the solvation free energy could increase the accuracy of the estimated hydration free energy.[183] As a result, an assessment of implicit solvent models with different nonpolar terms would provide an important basis for their improvements and future applications.

Although there are already some studies in the literature that assessed the accuracy of one or two of these models,[167,184–187] they used different molecule sets or atom type definitions. That makes it difficult to compare these studies among each other, or to compare the performance of the underlying nonpolar models independent of their parameterization. Together with Julia Setzler and Wolfgang Wenzel, I have carried out an assessment of the standard nonpolar solvation model based on the solvent accessible surface area and two advanced models.[179] We have chosen a set of small neutral molecules[188,189] over a set of proteins for the database of our assessment, because the small molecules contain a larger variety of chemical groups. Thus, they should provide a more challenging test for the models. This molecule set was already used to investigate how accurately the explicit TIP3P water model[189] or many common implicit solvent models[183,187,188] can reproduce experimental hydration free energies. In our study, we have chosen an approach that enables a fair comparison of the accuracy of the models. It is unbiased by the model's parameterization In addition, we closely examined the computed hydration free energies, which provide insights into the reasons why one model performs better than others do.

## 4.2 Investigated Implicit Solvent Models

Each of the three investigated implicit solvent models in our study consists of the same generalized Born term to describe electrostatic solvation effects, and one of three different terms to model nonpolar solvation effects. Therefore, the models are abbreviated by GBNP1, GBNP2, and GBNP3. The generalized Born term is given by Equation (2.35). The Born radii of the generalized Born model are determined by the R6 integral expression of Equation (2.40). The integration region is defined by the solvent excluded surface defined in Section 2.3. This surface requires atomic radii $r_i$ and a probe radius $p_\mathrm{r}$ for its construction. These are the free parameters of the generalized Born model.

The nonpolar term of GBNP1 is based on Equation (2.41) and uses only a single surface tension parameter $\gamma$ that is multiplied by the sum of the atomic solvent accessible surface areas $A_i$

$$\Delta G_{\mathrm{NP1}} = \gamma \sum_{i=1}^{N} A_i. \tag{4.1}$$

The nonpolar term of the GBNP2 model is also based on Equation (2.41) and uses atom type specific surface tension coefficients $\gamma_i$[113,114]

$$\Delta G_{\mathrm{NP2}} = \sum_{i=1}^{N} \gamma_i A_i. \tag{4.2}$$

Finally, the nonpolar term of the GBNP3 model is based on Equation (2.42)

$$\Delta G_{\mathrm{NP3}} = \gamma \sum_{i=1}^{N} A_i + p \sum_{i=1}^{N} V_i - \sum_{i=1}^{N} \frac{d_i}{(R_i + B)^3}. \tag{4.3}$$

GBNP3 additionally uses solvent accessible volumes $V_i$ to model the cost of cavity formation in the solvent and explicitly models dispersion interactions with the solvent via the dispersion coefficients $d_i$ and the Born radii $R_i$ plus a constant offset $B$. Since this nonpolar term also uses the Born radii of the electrostatic generalized Born term, it also depends on the atomic radii $r_i$ and the probe radius $p_{\mathrm{r}}$ that define the integration region for the Born radii.

The elements present in the molecules of the data set define the atom types. The data set contains ten different elements, wherefore we use ten different atom types. Although defining more atom types is possible, we decided to start with this minimal set of atom types. We also investigated the models GBNP1*, GBNP2* and GBNP3*. They have one additional atom type, because they differentiate between nitrogen atoms with positive and negative partial charge. Table 4.1 summarizes all three models and their freely adjustable parameters.

***Table 4.1. Overview over the free model parameters contained in the three different investigated implicit solvent models GBNP1, GBNP2, and GBNP3. The number of free parameters is also given. This number is either one or the same as the number of atom types. GBNP* refers to the three implicit solvent models GBNP1*, GBNP2* and GBNP3*, which differentiate between nitrogen atoms with positive and negative partial charge.[179]***

| Free model parameter | Description | GBNP1 | GBNP2 | GBNP3 | parameter count GBNP/GBNP* |
|---|---|---|---|---|---|
| $\boldsymbol{r_i}$ | atomic radii | X | X | X | 10/11 |
| $\boldsymbol{p_{\mathrm{r}}}$ | probe radius | X | X | X | 1/1 |
| $\boldsymbol{\gamma}$ | global SASA tension | X | | X | 1/1 |
| $\boldsymbol{\gamma_i}$ | atomic SASA tension | | X | | 10/11 |
| $\boldsymbol{p}$ | global SAV pressure | | | X | 1/1 |
| $\boldsymbol{d_i}$ | atomic dispersion coefficient | | | X | 10/11 |
| $\boldsymbol{B}$ | Born radii offset | | | X | 1/1 |

## 4.3 Model Parameterization

My first task was to generate a parameter set that allows a fair comparison of the models unbiased by their parameterization. One parameter set that allows such a comparison is simply the best possible parameter set. Given a set of molecules with experimentally determined hydration free energies as reference data, one can determine the best possible free parameter set by optimizing all free parameters to minimize an accuracy measure with respect to the reference data.

As the accuracy measure, I used the root mean square error between the experimental hydration free energies and computed solvation free energies for a single conformation of the molecule. The reason I use only single conformation solvation free energies instead of the hydration free energies is that the calculation of the former requires much less computational effort than the latter. This is necessary to enable the optimization of the large number of free parameters within a reasonable amount of computation time.

According to Mobley et al.,[188] the single conformation solvation free energies are in good agreement with experimental data, if the lowest energy snapshot from a vacuum simulation trajectory of the respective molecule is used as the single conformation. Thus, we will use these single conformation snapshots to compute the single conformation solvation free energies. We have acquired the vacuum trajectories from Mobley et al.[188] Julia Setzler has computed the vacuum energies of each snapshot in all trajectories with AMBER 10.[66] I have extracted the best vacuum energy conformation from each vacuum trajectory.

The next step in generating the required parameter sets for each model is to have a small computer program that reads in an arbitrary set of free parameters, the molecule files with the corresponding coordinates, atom types and AM1-BCC[190,191] partial charges, assigns the free parameters to each atom where necessary, and computes the solvation free energies. I have implemented these methods in a small C++ program. The program uses the PowerBorn method[173] to compute Born radii and the PowerSASA method[152,153] to compute solvent accessible surface areas and volumes. The solvation free energies for a conformation of a given molecule are then computed via Equations (4.1)-(4.3), (2.35) and (2.37) by the program.

The final step in generating the parameter sets is to carry out the optimization of the free parameters. To enable a fair comparison of the models, the optimized parameters have to represent the global minimum of the accuracy measure and not any local minimum. For that reason, I use a particle swarm global optimization implementation by Kondov.[192] In this method, a swarm of $N_\mathrm{s}$ individuals searches through the parameter space. The swarm's current best location as well as each individual's best location influences the search directions of the individuals. To ensure proper sampling of the parameter space, I have run the optimization procedure with different sets of swarm parameters for each model. This procedure resulted in 81 parameter sets for each model. For faster convergence, a local Powell optimization is carried out after 200 iterations of the particle swarm optimization for each parameter set. The valid ranges of all free parameters for the optimization procedure are given in Table 4.2.

***Table 4.2. Overview over the valid parameter ranges for all free model parameters during the parameter optimization procedure.***[179]

| **Parameter** | $\boldsymbol{r_i}$ [Å] | $\boldsymbol{p_\mathrm{r}}$ [Å] | $\boldsymbol{\gamma}$ [kcal/(mol Å$^2$)] | $\boldsymbol{\lvert\gamma_i\rvert}$ [kcal/(mol Å$^2$)] | $\boldsymbol{p}$ [kcal/(mol Å$^3$)] | $\boldsymbol{d_i}$ [(kcal Å$^3$)/mol] | $\boldsymbol{B}$ [Å] |
|---|---|---|---|---|---|---|---|
| **Minimum** | 0.5 | 0.5 | $10.0^{-6}$ | $10.0^{-6}$ | $10.0^{-6}$ | $10.0^{-6}$ | 0.0 |
| **Maximum** | 5.0 | 3.0 | 10.0 | 10.0 | 10.0 | $10.0^{6}$ | 5.0 |

The resulting root mean square errors $RMSE_\mathrm{FIT}$ from each run of the optimization procedure are shown in Figure 4.1. This includes the models GBNP1, GBNP2, and GBNP3 as well as the models with one additional atom type for nitrogen atoms with positive partial charge, GBNP1*, GBNP2*, and GBNP3*. I observe that the $RMSE_\mathrm{FIT}$ for GBNP1 and GBNP1* show a relative narrow distribution in comparison to the other models. For further data analysis, we only have considered the best of 81 parameter sets of each model, e.g. the parameter set with the lowest $RMSE_\mathrm{FIT}$.

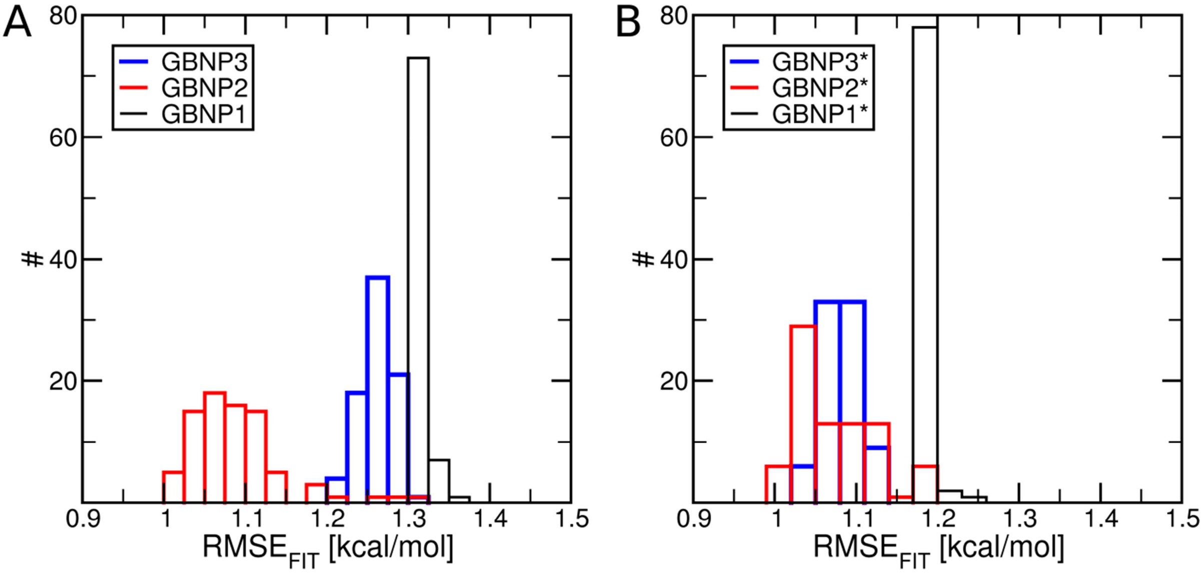


***Figure 4.1. These histograms show the root mean square errors $RMSE_{FIT}$ between experimental hydration free energies and single conformation solvation free energies after the parameter optimization procedure. For each model, 81 parameter sets were generated. Panel A shows the results for the GBNP1, GBNP2, and GBNP3 models. Panel B shows the results for the GBNP1*, GBNP2* and GBNP3* models.***

## *4.4 Comparison of Computed Hydration Free Energies*

To enable comparison of our data to the work of others, especially that of Knight and Brooks,[183] we decided to use hydration free energies instead of the single conformation solvation free energies for the assessment of the models. The hydration free energies are computed from the vacuum and implicit solvent trajectories of the molecules provided by Mobley et al.[188] with the help of the Bennett Acceptance Ratio method (see Section 2.4) as implemented in pyMBAR.[86] Julia Setzler computed the necessary molecular energies of each conformation with AMBER 10.[66] She used the general AMBER forcefield (GAFF)[193,194] and AM1-BCC partial charges.[190,191] I computed the solvation free energies for each conformation with my C++ program, which I extended to read the trajectories also. Afterwards, I used pyMBAR to compute the hydration free energies from the molecular energies and solvation free energies.

We have compared the computed hydration free energies of the models among each other and to other published results.[179] The computed hydration free energies in Figure 4.2 show that the GBNP2 model performs much better than the GBNP3 or GBNP1 models. In comparison to the results of Knight et al.,[183] we observed that the combined optimization of polar and nonpolar model parameters can provide significant improvements over just optimizing nonpolar parameters.[179] In comparison to the explicit water TIP3P model results of Mobley et al.,[189] the GBNP2 model has a lower root mean square error $RMSE_{\mathrm{HFE}}$ and a higher squared Pearson correlation coefficient $R^2$ to experimental data. This demonstrates that implicit models are in principle able to compute hydration free energies with the same or higher accuracy as explicit models, even with a very limited set of only ten atom types.[179]

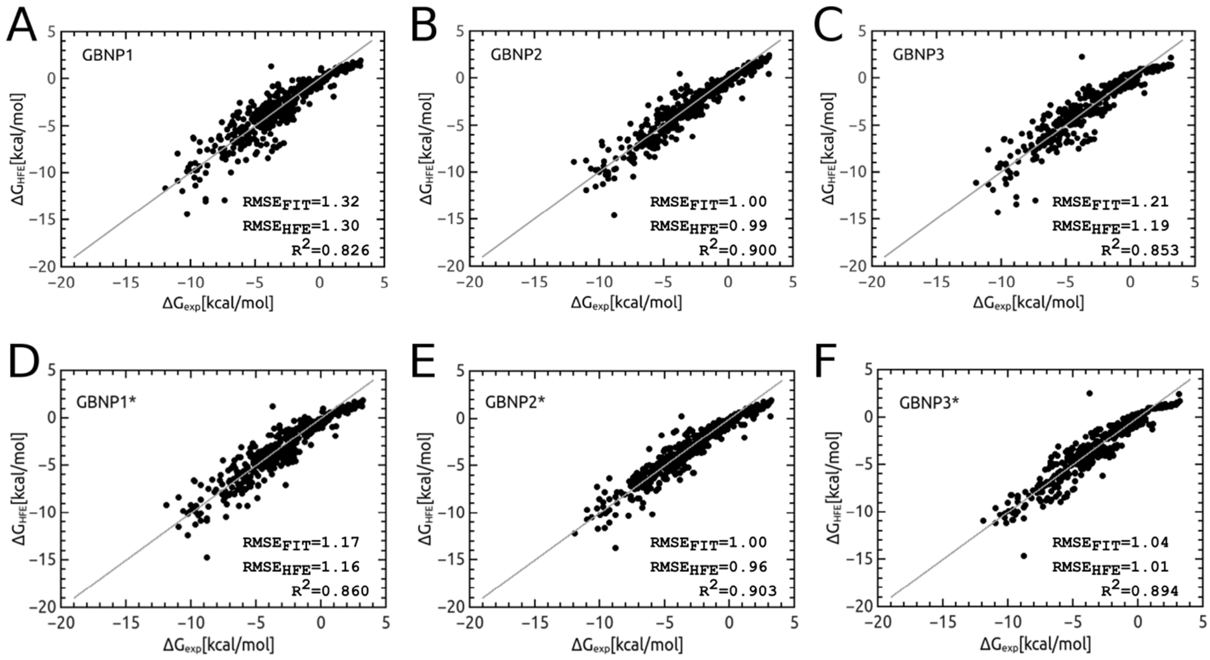


***Figure 4.2. Scatter plots show the computed hydration free energies over the corresponding experimental data for the GBNP1 (panel A), GBNP2 (panel B), GBNP3 (panel C), GBNP1* (panel D), GBNP2* (panel E), and GBNP3* (panel F) model. The gray line marks perfect agreement. In the plots, the root mean square errors that resulted from the parameter optimization procedure $RMSE_{FIT}$ are also given. In addition, the root mean square errors between experimental and computed hydration free energies $RMSE_{HFE}$ as well as the corresponding squared Pearson correlation coefficients $R^2$ are also given. Panels A to C are taken from Brieg et al.[179] Panels D to F were generated by Julia Setzler.***

To investigate the reasons for the moderate performance of the GBNP1 and GBNP3 models, Julia Setzler has grouped the data set into one subset for each atom type. A molecule is contained in such a subset, if it contains at least one atom of the respective atom type. Table 4.3 lists the size of these subsets and the respective root mean square errors $RMSE_{\mathrm{AT}}$ for each subset corresponding to one atom type. We concluded from the relatively large size of the nitrogen subset and the respective large $RMSE_{\mathrm{AT}}$ for the GBNP1 and GBNP3 models that the parameterization of nitrogen atoms is the source of these model's moderate performance.[179]

A closer investigation of the errors for the nitrogen atoms by Julia Setzler revealed that the GAFF nitrogen atom type “no” shows large systematic deviations.[179] We further found that only this nitrogen atom type has positive partial charge. The known asymmetric behavior of water around oppositely charged ions usually causes large differences in their respective hydration free energies.[195–202] Because all nitrogen atoms are assigned the same parameters in our GBNP1, GBNP2, and GBNP3 models, this behavior is not accounted for in the parameterization. Thus, the good performance of GBNP2 over GBNP1 or GBNP3 is partly due to its ability to cope well with this asymmetric behavior of water.[179]

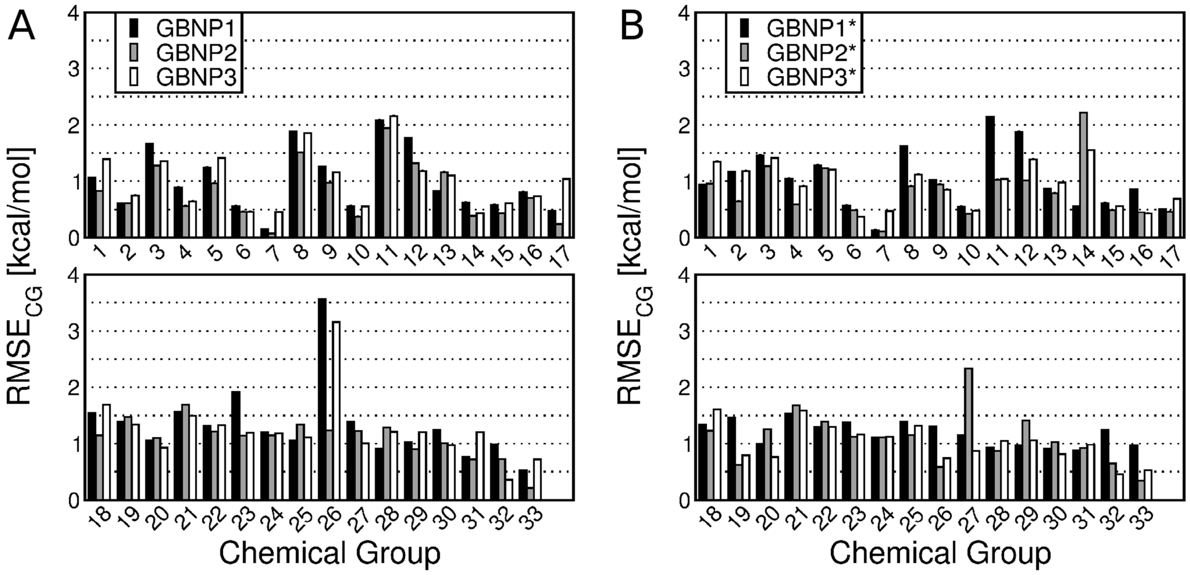
***Table 4.3. Root mean square errors $RMSE_{AT}$ in kcal/mol for subsets of molecules containing at least one respective atom type. Atom types C and H are excluded, because they are contained in nearly every molecule in the data set. The values in parentheses for fluorine exclude hexafluoropropene, for which the experimental hydration free energy was in error as became apparent during the review process of our work.[179]***

| **atom type** | **All** | **O** | **N** | **F** | **Br** | **S** | **I** | **Cl** | **P** |
|---|---|---|---|---|---|---|---|---|---|
| **subset size [#]** | 499 | 227 | 86 | 26 | 23 | 21 | 11 | 8 | 2 |
| **$RMSE_{AT}$ GBNP1 [kcal/mol]** | 1.30 | 1.65 | 1.93 | 1.67<br>(1.37) | 0.69 | 1.08 | 1.21 | 0.66 | 0.74 |
| **$RMSE_{AT}$ GBNP2 [kcal/mol]** | 0.99 | 1.13 | 1.14 | 1.70<br>(1.51) | 0.50 | 0.71 | 1.15 | 0.40 | 0.96 |
| **$RMSE_{AT}$ GBNP3 [kcal/mol]** | 1.19 | 1.41 | 1.76 | 1.56<br>(1.04) | 0.56 | 0.84 | 1.18 | 0.24 | 0.82 |

We have further investigated how an additional atom type for nitrogen atoms with positive partial charge increases the agreement to experimental data. Therefore, we have again carried out the parameterization procedure using the additional nitrogen atom type. We termed the models with the additional atom type GBNP1*, GBNP2* and GBNP3*. The agreement of the computed hydration free energies increased significantly for GBNP1* and GBNP3* over GBNP1 and GBNP3 respectively. The agreement only marginally increased for GBNP2* over GBNP2. The data is visualized in Figure 4.2. The figure also contains the respective root mean square errors $RMSE_{\mathrm{HFE}}$, squared Pearson correlation coefficients $R^2$, and resulting root mean square errors from the model parameterization procedure $RMSE_{\mathrm{FIT}}$. Nevertheless, GBNP2* has still the lowest $RMSE_{\mathrm{HFE}}$ and highest $R^2$, wherefore it is still the best of the three investigated models. However, the GBNP3* model now comes very close to the performance of GBNP2*. GBNP1* is still the worst performing model.[179]

## 4.5 Model Assessment Based on Chemical Groups

I have compared the experimental and computed hydration free energies for each chemical group present in the data set. The classification of the molecules into the chemical groups is taken from Knight and Brooks.[183] There are 33 different chemical groups. Each molecule may be part of more than one chemical group. The root mean square errors between the experimental and computed hydration free energies for each chemical group $RMSE_{\mathrm{CG}}$ are shown in Figure 4.3 for all six investigated models. The resulting average root mean square error and its standard deviation over all groups for a given model are listed in Table 4.4. The largest root mean square error of a chemical group is also listed in that table for each model.



C

| No. | Chemical group | No. | Chemical group | No. | Chemical group | No. | Chemical group | No. | Chemical group |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Acetal | 8 | Amine | 15 | Chloro alkyl | 22 | Heterocyclic | 29 | Other |
| 2 | Acid | 9 | Aromatic | 16 | Chloro aryl | 23 | Hypervalents | 30 | Phenol |
| 3 | Alcohol | 10 | Bromo | 17 | Cyclochydrocarb. | 24 | Iodo | 31 | Sulfur |
| 4 | Aldehyde | 11 | CA amide | 18 | Ether alkyl | 25 | Ketone | 32 | Thioether |
| 5 | Alkane | 12 | CA ester | 19 | Ether aryl | 26 | Nitro | 33 | Thiol |
| 6 | Alkene | 13 | CA ortho | 20 | Fluoro | 27 | Nitrogen | | |
| 7 | Alkyne | 14 | Carbonitrile | 21 | Halogen | 28 | Orthoester | | |

***Figure 4.3. Root mean square errors between experimental and computed hydration free energies by chemical group ($RMSE_{CG}$) for the GBNP1, GBNP2 and GBNP3 models (panel A) and GBNP1*, GBNP2* and GBNP3* models (panel B). The corresponding name for each chemical group number is given in panel C. Carboxylic acid is denoted by CA.[179]***

***Table 4.4. Average root mean square error and its standard deviation for all investigated models over the whole set of 33 chemical groups.***

| Model | Average $RMSE_{\text{CG}}$ [kcal/mol] | Standard deviation of average $RMSE_{\text{CG}}$ [kcal/mol] | Maximum $RMSE_{\text{CG}}$ [kcal/mol] |
|---|---|---|---|
| **GBNP1** | 1.2 | 0.6 | 3.6 |
| **GBNP2** | 0.9 | 0.4 | 1.9 |
| **GBNP3** | 1.1 | 0.5 | 3.2 |
| **GBNP1*** | 1.1 | 0.4 | 2.1 |
| **GBNP2*** | 1.0 | 0.5 | 2.3 |
| **GBNP3*** | 1.0 | 0.4 | 1.6 |

For the models with ten atom types, the GBNP2 model has the lowest average $RMSE_{\text{CG}}$, standard deviation of the average $RMSE_{\text{CG}}$ and maximum $RMSE_{\text{CG}}$, followed by GBNP3 and GBNP1. Furthermore, the GBNP1 and GBNP3 models have two chemical groups with $RMSE_{\text{CG}}$ larger than 2.0 kcal/mol, while there is no such group for the GBNP2 model. Looking at the GBNP* models with eleven atom types, the average $RMSE_{\text{CG}}$, its standard deviation, as well as the maximum $RMSE_{\text{CG}}$ of the GBNP2* model increase over GBNP2. In contrast, the corresponding values for the GBNP1* and GBNP3* models are lower than those for GBNP1 and GBNP3 are respectively. The standard deviation of the average $RMSE_{\text{CG}}$ and the maximum $RMSE_{\text{CG}}$ of GBNP2* are now larger than that of the other two models with eleven atom types. In addition, the GBNP2* model has two chemical groups with $RMSE_{\text{CG}}$ larger than 2.0 kcal/mol, while the GBNP1* model has only one such group and the GBNP3* model no such group. However, the average $RMSE_{\text{CG}}$ of GBNP2* is as low as that of GBNP3*, while that of GBNP1* is slightly larger than that of the previous two models.

The values in Table 4.4 suggest that GBNP3* performs better than GBNP2*. In contrast, the analysis of the hydration free energies based on the single molecules of the data set in Section 4.4 suggested that GBNP2* performs better than GBNP3*. Thus, the two analysis methods weigh the errors of the computed hydration free energies to the experimental data differently, due to the classification of the molecules into chemical groups. However, there is no clear best model according this analysis. The average $RMSE_{\text{CG}}$ and its standard deviation is lower for GBNP2 compared to GBNP3*, but the maximum $RMSE_{\text{CG}}$ of GBNP2 is larger than that of GBNP3*. Nevertheless, the GBNP1 model performs worst in this analysis too.

I note that in Figure 4.3A, the maximum $RMSE_{\text{CG}}$ for the GBNP1 and GBNP3 model is for the same chemical group. This is the nitro group (no. 26) with $RMSE_{\text{CG}}$ of 3.6 and 3.2 kcal/mol respectively. These common errors suggest a systematic problem of the two models. On the other hand, the $RMSE_{\text{CG}}$ of the GBNP2 model for that group is 1.2 kcal/mol, and therefore much smaller. Figure 4.4

shows the structure of a nitro group. The nitrogen is bound to two oxygen atoms. The assignment of the partial charges via the AM1-BCC method[190,191] results in a positive partial charge for the nitrogen atom. In contrast, all nitrogen atoms in the data set not belonging to the nitro group have negative partial charges. Thus, the large errors of the nitro group are due to the already discussed asymmetric behavior of water around oppositely charged ions in Section 4.4. The good performance of the GBNP2 for the nitro group in contrast to the GBNP1 and GBNP3 model suggests that the former model is able to handle this effect well without explicit parameterization.[179] In addition, the increased accuracy of the GBNP1* and GBNP3* models over GBNP1 and GBNP3 respectively, demonstrate the importance of accounting for the asymmetry of water in implicit solvent models to accurately estimate hydration free energies of small molecules.[179]

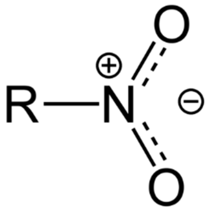


***Figure 4.4. Chemical structure of the nitro group. R denotes residual chemical groups attached to the nitro group. Plus and minus signs mark the distribution of partial charges. The nitrogen atom carries positive partial charge, while the oxygen atoms carry negative partial charge.[203]***

The increase of the standard deviation of the average $RMSE_{\mathrm{CG}}$ and the maximum $RMSE_{\mathrm{CG}}$ for the GBNP2* model over GBNP2 in Table 4.4 is surprising. I have investigated the reasons for this behavior. For the GBNP2 model, a single nitrogen atom type fits all chemical groups containing nitrogen reasonably well. In the GBNP2* model the additional atom type for positively charged nitrogen atoms results in even better agreement for the nitro group. However, the nitrogen atom type for negatively charged nitrogen atoms has to account for all other nitrogen atoms. These consist of many nitrogen atoms with large negative partial charge and only a few nitrogen atoms with small negative partial charge. The latter belong to the carbonitrile group. Since the parameterization procedure tries to reduce the root mean square error $RMSE_{\mathrm{FIT}}$ over all molecules, it may do so by finding a parameter set with slightly better overall $RMSE_{\mathrm{FIT}}$ at the expense of introducing a large $RMSE_{\mathrm{CG}}$ for the few molecules with carbonitrile groups.

In the GBNP2 model, only two atom type dependent nitrogen parameters had for all nitrogen atoms with their wide range of partial charges. Thus, the nitrogen atoms in the nitro group with their large positive partial charge balanced out the few molecules with carbonitrile group that have small negative partial charges against the large number of molecules containing nitrogen atoms with large negative partial charges. The extra nitrogen atom type of the GBNP2* model removes that balance. The positively charged nitrogen atoms of the nitro group now have a separate atom type, wherefore molecules containing a nitro group can no longer balance the molecules containing a carbonitrile group against the large amount of molecules containing any remaining chemical group with nitrogen atoms.[179] Thus, the performance of the GBNP2* model does not increase over that of the GBNP2 model.

Mobley et al.[204] reported that explicit TIP3P water in combination with AM1-BCC charges and the GAFF forcefield has problems in reproducing hydration free energies of molecules containing hypervalent sulfurs (group no. 23). Knight and Brooks[183] reported the same problem for many common implicit solvent models. They argued that it might be necessary to change Lennard-Jones parameters to achieve good agreement with experimental data. The computed hydration free energies by us show that this is not necessary. The $RMSE_{\text{CG}}$ for the hypervalent sulfur group of the GBNP2, GBNP2*, GBNP3 and GBNP3* model is between 1.1 and 1.2 kcal/mol and therefore in good agreement with experimental data. The corresponding errors for GBNP1 and GBNP1* are 1.91 and 1.37 kcal/mol. The GBNP1 and GBNP1* have no nonpolar atom type dependent solvation parameters (see Table 4.1). Therefore, the larger errors for the two latter models suggest that atom type dependent nonpolar solvation parameters are necessary to estimate hydration free energies of compounds containing hypervalent sulfurs correctly.

For GBNP1* amines, carbon amides, and carbon esters still show significant errors with $RMSE_{\text{CG}} > 2.0$ kcal/mol (Figure 4.3). The reason for this error is again the asymmetric behavior of water. The AM1-BCC charges for carbon atoms in these groups are positive, while carbon atoms in other chemical groups like alkanes carry small negative partial charges. Because charge differences are smaller, the induced errors are also smaller than those of the nitro group are. Nevertheless, we expect an additional carbon atom type to reduce these errors further.[179] In addition, the exposition of the carbon atoms to water is very important for the asymmetric behavior of water to have an effect. If the carbon atom is not exposed, neglecting the asymmetric behavior of water will not introduce a large error.

In conclusion, the analysis of the implicit solvent models presented in this chapter provides a solid foundation for future improvements of implicit solvent models. Especially the consideration of the asymmetry of water seems to play a key role in future improvements of implicit solvent models. Atom type dependent nonpolar solvation parameters can also increase the accuracy of estimated hydration free energies for small molecules significantly. However, one has to define atom types carefully to not introduce large errors for sparsely represented entities in the training set.

The next step in the improvement of implicit solvent models will be to see how these results transfer to larger molecules like proteins. These do not contain such a wide variety of chemical groups, e.g. nitro groups are not present in proteins. Therefore, accounting for the asymmetry of water may not be as important for proteins as it is for the small molecules considered in this study. However, proteins can undergo large conformational changes that cause the burial of specific groups inside the protein and the exposition of other groups to water. To improve the description of solvation effects related to these conformational changes, a different approach than the used one by us will be necessary. The reason is that the related free energy changes cannot be measured in experiments, because it would require enforcing a specific conformational change of the protein. Moreover, intramolecular interactions are also important for the thermodynamics of conformational changes.

# 5 Extensions for an Implicit Membrane Model

*Reproduced in part from Setzler, J.; Seith, C.; Brieg, M.; Wenzel, W. "SLIM: An Improved Generalized Born Implicit Membrane Model." J. Comput. Chem.* ***2014****, 35, 2027–2039. Permission granted by John Wiley & Sons, Inc. (License number 3511861064857). Copyright © 2014 Wiley Periodicals, Inc.*

This chapter introduces an extension of the generalized Born implicit solvent model of Section 3.3 to account for some basic properties of biological membranes. These membranes represent another important physiological environment of proteins. In cooperation with Julia Setzler and Carolin Seith, I have developed the so-called SIMONA layered implicit membrane (SLIM) model that enables Monte Carlo simulations in SIMONA with an implicit solvent and membrane representation. The first section explains the properties that this model accounts for and how they are incorporated into the SLIM model. Subsequently, the second section gives details about the implementation of this idea into my PowerBorn algorithm and SIMONA. The third section reviews the parameterization of the SLIM model, its comparison to Poisson-Boltzmann reference calculations, and results of Monte Carlo simulations of small membrane proteins using the SLIM model. To enable the study of larger systems, the fourth section describes the parallel implementation of the SLIM model, and the last section gives an overview of the performance of Monte Carlo simulations with the SLIM model in SIMONA.

## 5.1 Motivation and Basic Idea of the SLIM Model

As explained in Section 2.1, biological membranes represent another important physiological environment for proteins. Similar to implicit solvent models introduced in Section 2.4, implicit membrane models offer the possibility to reduce the computational cost for studies of membrane proteins significantly. However, this requires the incorporation of the membrane into the implicit solvent model. Due to the heterogeneous composition of the membrane bilayer with its headgroup region and the lipid tail region, this task is much more challenging than for homogenous water.

For water, the generalized Born implicit solvent model introduced in Section 2.4 accounts for the polarization of water by the solute charges and the interaction of the induced polarization charges with the solute charges. The strength of this interaction strongly depends on the ratio of the dielectric constants of the solvent and solute regions. If the solvent is water, this ratio is very small and the resulting interaction rather strong.

In a recent study, Nymeyer and Zhou[205] computed effective dielectric constants within a membrane. They find it should be represented by at least two different dielectric regions. These regions correspond to the lipid tails inside the membrane core with a very low dielectric constant and a transition region between the membrane core and the headgroup region with an intermediate dielectric constant. Thus, the induced polarization charges at these interfaces are much smaller due to the larger ratios of the dielectric constants between solute interior, membrane core, and headgroup regions. An implicit continuum membrane model will have to account for these different dielectric regions. Unfortunately, by construction, the computationally efficient generalized Born model is limited to the presence of only two different dielectric regions.

Nevertheless, several attempts in the past have been made to include a membrane implicitly into the generalized Born model. Spassov et al. simply modeled the membrane as a single low dielectric slab.[206] This results again in only two dielectric regions that can be treated with the generalized Born model. Im et al.[207] or Ulmschneider et al[208]. have developed own implicit membrane models based on this idea. Tanizaki and Feig[209] proposed a different method to include the dielectric regions of the membrane into the generalized Born model. They use a position-dependent dielectric profile function that replaces the dielectric constant of water $\epsilon_{\mathrm{w}}$ in Equation (2.35). While this method allows the inclusion of any number of dielectric regions into the generalized Born model, it does not correctly account for the membrane in the interaction terms. For example, the interaction of two ions just outside the membrane will not be altered in this model by the presence of the membrane. Thus, qualitatively correct modeling of interactions with induced polarization charges in the presence of a realistic membrane representation using the generalized Born model is an unsolved problem.

To address this problem, I have developed a new implicit membrane model based on the generalized Born model together with Julia Setzler, Carolin Seith, and Wolfgang Wenzel. We call this model SIMONA Layered Implicit Membrane (SLIM). This model solves the qualitative problems of previous generalized Born implicit membrane models. As a result, it yields electrostatic solvation free energies in better agreement with Poisson-Boltzmann calculations than for previous models.[210]

To motivate the basic idea of the SLIM model, I will shortly review some facts about the electrostatics of dielectric media. According to the boundary conditions at dielectric interfaces in Equations (2.32) and (2.33), the interface causes a jump in the normal component of the displacement field. Together with Equation (2.29) and Gauss law, one can show that this jump is due to induced polarization charges that are located at the interface. These polarization charges are induced by an electric field, e.g. that of a solute due to its charges. These polarization charges will interact with all other charges present in the system. This includes the sources of the external field, e.g. the solute charges, the induced polarization charges themselves, as well as induced polarization charges at other interfaces. An accurate generalized Born model only approximates the interactions between the solute charges and induced polarization charges as well as the induced polarization charges themselves.[211] However, it cannot model the interaction of induced polarization charges at different dielectric interfaces.

Based on these facts, my basic idea for the SLIM model was to decompose an environment consisting of multiple dielectric regions into multiple environments consisting of only two dielectric regions each (see Figure 5.1). The simpler environments can then be treated with established generalized Born models. However, this decomposition neglects the interactions among the induced polarization charges at each interface. Nevertheless, it may be possible to find some empiric correction that can account for the interaction of the induced polarization charge, at least if the system has a fixed simple geometry.

For our SLIM model, we will consider the following geometry of dielectric regions based on the work of Nymeyer and Zhou[205] and Tanizaki and Feig.[209] The region of the protein $V_\mathrm{p}$ will have dielectric constant $\epsilon_\mathrm{p}$. The membrane core region $V_\mathrm{c}$ is modeled by an infinite dielectric slab with dielectric constant $\epsilon_\mathrm{c}$ perpendicular to the z-axis of the coordinate system. We will follow the approach of Spassov et al. and use the same dielectric constant for the protein interior and the membrane core.[206] Thus, the united region is $V_\mathrm{pc} = V_\mathrm{p} \cup V_\mathrm{c}$ and has dielectric constant $\epsilon_\mathrm{pc} = \epsilon_\mathrm{p} = \epsilon_\mathrm{c}$. However, our model does not require this decision. It can be generalized to an arbitrary number of dielectric regions. See Setzler et al. for a more general formulation how to decompose an environment consisting of an arbitrary number of dielectric regions.[210] The membrane core region is surrounded by another two infinite dielectric slabs. These slabs constitute the region $V_\mathrm{h}$ of intermediate dielectric constant $\epsilon_\mathrm{h}$. We will refer to this region as the headgroup region. However, they only model the transition between the membrane core and the headgroup, wherefore they may not coincide with the positions of the headgroups in a real membrane. Finally, the slabs are embedded in implicit water denoted by the region $V_\mathrm{w}$ with dielectric constant $\epsilon_\mathrm{w}$. This geometry is also depicted in Figure 5.1.

The decomposition of the previously described environment can be translated to the decomposition of the electrostatic solvation free energy $\Delta G_\mathrm{elec}$ into two generalized Born terms $\Delta G_\mathrm{GB}$ of Equation (2.35) with two sets of Born radii $\{R\}_1$ and $\{R\}_2$ respectively[210]

$$\begin{aligned}\Delta G_{\mathrm{elec}}(\epsilon_{\mathrm{pc}}, V_{\mathrm{pc}};\ \epsilon_{\mathrm{h}}, V_{\mathrm{h}};\ \epsilon_{\mathrm{w}}, V_{\mathrm{w}}) \approx \Delta G_{\mathrm{elec}}^{\mathrm{SLIM}}(\epsilon_{\mathrm{pc}}, V_{\mathrm{pc}};\ \epsilon_{\mathrm{h}}, V_{\mathrm{h}};\ \epsilon_{\mathrm{w}}, V_{\mathrm{w}}) = \\ \Delta G_{\mathrm{GB}}(\epsilon_{\mathrm{pc}}, V_{\mathrm{pc}};\ \epsilon_{\mathrm{h}}, V_{\mathrm{h}} \cup V_{\mathrm{w}};\ \{R\}_1) + \Delta G_{\mathrm{GB}}(\epsilon_{\mathrm{h}}, V_{\mathrm{pc}} \cup V_{\mathrm{h}};\ \epsilon_{\mathrm{w}}, V_{\mathrm{w}};\ \{R\}_2).\end{aligned} \tag{5.1}$$

The first generalized Born term of this equation treats the interface between the membrane core or protein interior region $V_{\mathrm{pc}}$, and the headgroup region $V_{\mathrm{h}}$. In this term, the water region $V_{\mathrm{w}}$ is assigned the dielectric constant $\epsilon_{\mathrm{h}}$ instead of $\epsilon_{\mathrm{w}}$. Thus, there are only regions that have dielectric constant $\epsilon_{\mathrm{pc}}$ and $\epsilon_{\mathrm{h}}$. The set of Born radii $\{R\}_1$ is computed via Equation (2.40), but the integration region includes all regions with dielectric constant $\epsilon_{\mathrm{h}}$, e.g. the union of regions $V_{\mathrm{h}}$ and $V_{\mathrm{w}}$. The second term in Equation (5.1) treats the interface between the headgroup region $V_{\mathrm{h}}$ with dielectric constant $\epsilon_{\mathrm{h}}$ and the water region $V_{\mathrm{w}}$ with dielectric constant $\epsilon_{\mathrm{w}}$. To have only two different dielectric constants in the system modeled by this generalized Born term, the dielectric constant $\epsilon_{\mathrm{h}}$ is also assigned to the membrane core and protein interior region $V_{pc}$. For this generalized Born term, the integration region for the set of Born radii $\{R\}_2$ is the region with dielectric constant $\epsilon_{\mathrm{w}}$, e.g. the water region $V_{\mathrm{w}}$.

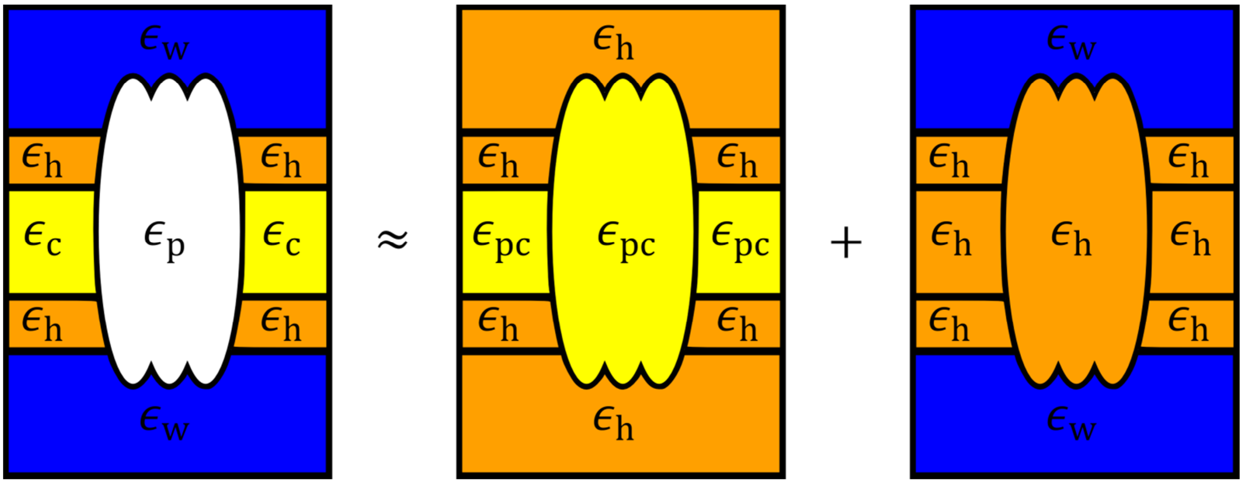


***Figure 5.1.*** *This sketch visualizes the decomposition of a complex environment into two simpler environments of the SLIM[210] model. The protein region $V_p$ (white) with dielectric constant $\epsilon_p$ is embedded in a membrane consisting of core region $V_c$ (yellow) with dielectric constant $\epsilon_c$ and headgroup region $V_h$ (orange) with dielectric constant $\epsilon_h$, which is surrounded by a water region $V_w$ with dielectric constant $\epsilon_w$. The SLIM model assumes the same dielectric constant $\epsilon_{pc}$ for the membrane core and protein regions. The membrane is decomposed into two simpler environments. The first of those has $\epsilon_h$ assigned to the water region. In the second, $\epsilon_h$ is assigned to the protein and membrane core region. As a result, both simpler environments contain only regions with two different dielectric constants. Thus, they can be treated with established generalized Born methods.*

An important property of the decomposition in Equation (5.1) comes to bear if both sets of Born radii are equal. Since all Born radii are computed via the R6 integral expression of Equation (2.40), the sets will be equal if the integration regions are equal. This is for example the case if the headgroup region $V_{\mathrm{h}}$ vanishes. As a result, the membrane will be modeled by only one dielectric slab. It follows from Equation (2.35) that the dielectric constant $\epsilon_{\mathrm{h}}$ cancels out and the two generalized Born terms can be combined into a single term

$$\begin{aligned}&\Delta G_{\mathrm{GB}}\big(\epsilon_{\mathrm{pc}}, V_{\mathrm{pc}};\ \epsilon_{\mathrm{h}}, V_{\mathrm{w}};\ \{R\}_1\big) + \Delta G_{\mathrm{GB}}\big(\epsilon_{\mathrm{h}}, V_{\mathrm{pc}};\ \epsilon_{\mathrm{w}}, V_{\mathrm{w}};\ \{R\}_1\big)\\ &\quad = \Delta G_{\mathrm{GB}}\big(\epsilon_{\mathrm{pc}}, V_{\mathrm{pc}};\ \epsilon_{\mathrm{w}}, V_{\mathrm{w}};\ \{R\}_1\big).\end{aligned} \tag{5.2}$$

The resulting model with only one dielectric slab to represent the membrane is similar to that of Spassov et al.[206] Another case where the sets of Born radii will be equal is when the protein will be far away from the slabs. In that case, all contributions of the slabs to the Born radii integrals of Equation (2.40) will be negligible. This is also the case if both slabs vanish. Since $V_{\mathrm{pc}} = V_{\mathrm{p}} \cup V_{\mathrm{c}}$, the single resulting generalized Born term is

$$\begin{aligned}&\Delta G_{\mathrm{GB}}\big(\epsilon_{\mathrm{pc}}, V_{\mathrm{p}};\ \epsilon_{\mathrm{h}}, V_{\mathrm{w}};\ \{R\}_1\big) + \Delta G_{\mathrm{GB}}\big(\epsilon_{\mathrm{h}}, V_{\mathrm{p}};\ \epsilon_{\mathrm{w}}, V_{\mathrm{w}};\ \{R\}_1\big)\\ &\quad = \Delta G_{\mathrm{GB}}\big(\epsilon_{\mathrm{pc}}, V_{\mathrm{p}};\ \epsilon_{\mathrm{w}}, V_{\mathrm{w}};\ \{R\}_1\big).\end{aligned} \tag{5.3}$$

This generalized Born term consists of one protein region $V_{\mathrm{p}}$ with dielectric constant $\epsilon_{\mathrm{pc}}$ and one water region $V_{\mathrm{w}}$ with dielectric constant $\epsilon_{\mathrm{w}}$. It is the standard implicit solvent generalized Born term. In conclusion, the proposed decomposition contains the limiting cases of the standard generalized Born model of Still et al.[105] and the simple implicit membrane model of Spassov et al.[206]

Another aspect that an implicit membrane model should account for is the absence of the hydrophobic effect introduced in Section 2.4 inside the membrane. In contrast to water molecules, the lipid tails have no large dipole moments. Therefore, they do not form hydrogen bond networks that may be disrupted due to the presence of a solute. To model this effect within the SLIM model, we use the empiric approach of Tanizaki and Feig.[209] They use a solvent accessible surface area term with a z-coordinate dependent profile function $\mathrm{s}(|z_i|)$

$$\Delta G_{\mathrm{np}}^{\mathrm{SLIM}} = \gamma \sum_{i=0}^{N} \mathrm{s}(|z_i|) A_i. \tag{5.4}$$

The profile function is derived from explicit all-atom calculations of the solvation free energy of a neutral oxygen molecule at different positions in the membrane. If the thickness of the explicit membrane differs from that of the implicit membrane, we use a stretched profile function

$$\Delta G_{\mathrm{np}}^{\mathrm{SLIM}} = \gamma \sum_{i=0}^{N} \mathrm{s}\left(|z_i| \frac{h_0}{h_{\mathrm{m}}}\right) A_i. \tag{5.5}$$

Here, $h_0 = 30$ Å is the membrane thickness for the original profile and $h_{\mathrm{m}}$ is the actual membrane thickness. In summary, the solvation free energy of the SLIM model is

$$\Delta \mathrm{G}^{\mathrm{SLIM}} = \Delta \mathrm{G}_{\mathrm{elec}}^{\mathrm{SLIM}}\left(\epsilon_{\mathrm{pc}}, V_{\mathrm{pc}};\ \epsilon_{\mathrm{h}}, V_{\mathrm{h}};\ \epsilon_{\mathrm{w}}, V_{\mathrm{w}}\right) + \Delta G_{\mathrm{np}}^{\mathrm{SLIM}}. \tag{5.6}$$

## 5.2 Implementation of the SLIM Model

The SLIM model requires the computation of two sets of Born radii. For the computation of each set, a different dielectric slab that has the same dielectric constant as the protein interior has to be accounted for. This means that the region of the slab has to be excluded from the integration region in Equation (2.40). Therefore, some changes to the PowerBorn method for the computation of Born radii described in Section 3.3 are necessary. In this section, I will explain the necessary steps to exclude the integration from the slab region.

To implement this feature, the treatment of three different cases is necessary, as illustrated in Figure 5.2. In the first case the bounding box that separates the numerical integration on the inside from the analytical integration on the outside, lies completely inside the slab (Figure 5.2A). In that case, no numerical integration procedure is necessary. The remaining regions are treated analytically. For an atom with z-coordinate $z_i$, the integral of Equation (2.40) over the volume outside the slab with lower and upper boundaries $z_{\mathrm{l}}$ and $z_{\mathrm{u}}$ is[210]

$$I_{\mathrm{slab}}^{\mathrm{outisde}}(z_i, z_{\mathrm{l}}, z_{\mathrm{u}}) = \frac{\pi}{6}\left(\frac{z_i - z_{\mathrm{u}}}{(z_i - z_{\mathrm{u}})^4} - \frac{z_i - z_{\mathrm{l}}}{(z_i - z_{\mathrm{l}})^4}\right). \tag{5.7}$$

If the bounding box is completely outside the slab (Figure 5.2B), the usual PowerBorn integration can be applied. Before the PowerBorn integral is converted to the Born radius via Equation (2.40), the integral over the slab region is subtracted from the usual PowerBorn integral. This contribution is given by

$$I_{\mathrm{slab}}^{\mathrm{inside}}(z_i, z_{\mathrm{l}}, z_{\mathrm{u}}) = -I_{\mathrm{slab}}^{\mathrm{outside}}(z_i, z_{\mathrm{l}}, z_{\mathrm{u}}). \tag{5.8}$$

In the last case, the bounding box touches at least one of the slab boundaries. If it touches only one, then the bounding box is shifted just outside the slab (Figure 5.2C). If it also touches the second boundary of the slab, a second bounding box is constructed at the opposite side just outside the slab. Within these shifted bounding boxes, the usual numerical PowerBorn integration procedure described in section 3.3 can be applied. The integration over the region outside the slab and outside the bounding box is solved analytically by converting the volume integral to surface integrals via Gauss's law. The integrals $I_{\text{square}}$ over the faces of the bounding box are given by Brieg and Wenzel,[173] where the details of the PowerBorn algorithm are explained. The integral $I_{\text{slab}}^{\text{square}}$ over the slab surface excluding the square of the bounding box can then be computed by

$$I_{\text{slab}}^{\text{square}} = I_{\text{slab}}^{\text{inside}} - I_{\text{square}}. \tag{5.9}$$

With that, the Born radii can be computed in the presence of a low dielectric slab, yielding the two sets of Born radii in Equation (5.2). The computation of each generalized Born term in that equation is done as described in Section 3.3.

The implementation of the nonpolar term in Equation (5.4) uses the PowerSASA[152,153] method to compute the solvent accessible surface area $A_i$ of each atom. The scaling function is computed by the formulas given by Tanizkai and Feig.[209] Afterwards, the scaling function is multiplied with the atomic solvent accessible surface area $A_i$ and the surface tension coefficient $\gamma$, and summed.

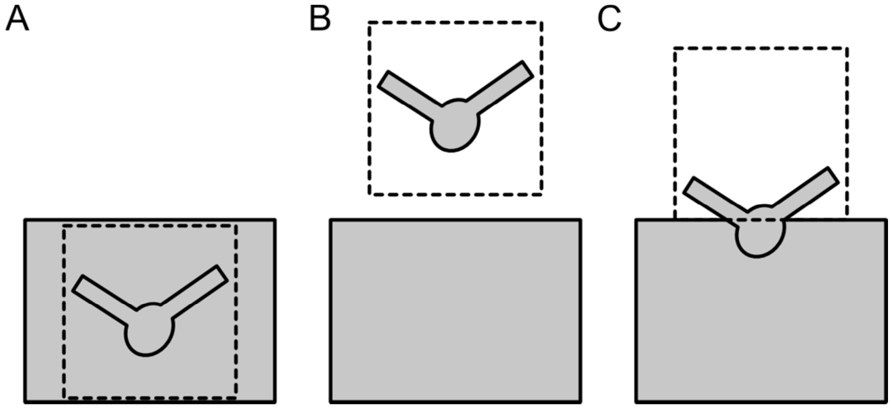


*Figure 5.2. Illustration of the three different cases that need to be treated to incorporate a low dielectric slab in the integration procedure of the PowerBorn[173] method. The protein's bounding box is either completely inside the slab (panel A), completely outside the slab (panel B) or partly inside the slab (panel C). In the last case, the bounding box is shifted to the boundary of the slab, and if it also touches the opposite slab boundary, a second bounding box is constructed. The low dielectric regions of the protein and the slab are shaded grey. The dashed line marks the bounding box inside which the numerical PowerBorn integration procedure is performed.*

## 5.3 Assessment of the SLIM Model

Julia Setzler and Carolin Seith carried out comparisons of the electrostatic solvation free energy of the SLIM model to Poisson-Boltzmann reference calculations including an implicit membrane representation. They first compared the electrostatic solvation free energies of a single ion that is pulled through the membrane. They find that if the SLIM model uses the same thicknesses and dielectric constants as in the Poisson-Boltzmann reference calculations (Figure 5.3A, black line), the SLIM model systematically overestimates the absolute value of electrostatic solvation free energy, especially in the transition region between the headgroup and the membrane core (Figure 5.3A, red dotted line).[210] The reason for this behavior is likely the neglected interaction between the induced polarization charges at the different dielectric interfaces as described in Section 5.1. However, they also showed that this error could be corrected by using optimized thicknesses and dielectric constants (Figure 5.3A, orange dashed line). Usage of these optimized constants results in very good agreement with Poisson-Boltzmann calculations.[210] They also compared the SLIM model with only one dielectric slab, which is similar to that of Spassov et al.[206] to the Poisson-Boltzmann results and find large deviations (Figure 5.3A, blue dot-dashed line). The transition is much steeper, as could be expected from the results of Nymeyer and Zhou.[205]

Julia Setzler and Carolin Seith further compared the interaction term of the electrostatic solvation free energy for two ions by computing the total electrostatic solvation free energy and subtracting the self-energy terms in Equation (2.36). The results in Figure 5.3B also show overestimated absolute values for the interaction terms if the thicknesses and dielectric constants of the Poisson-Boltzmann membrane model are used in the SLIM model (Figure 5.3B, red dotted line). These errors decrease significantly if the optimized thicknesses and dielectric constants are used, however, the absolute values of the interaction term of the electrostatic solvation free energy is still slightly overestimated (Figure 5.3B, orange dashed line). Again, the model similar to that of Spassov et. al. shows significant deviations from the Poisson Boltzmann calculations (Figure 5.3B, blue dot-dashed line). These results demonstrate the improved agreement of the SLIM model to much more computationally expensive Poisson-Boltzmann reference calculations in comparison to the model of Spassov et al.

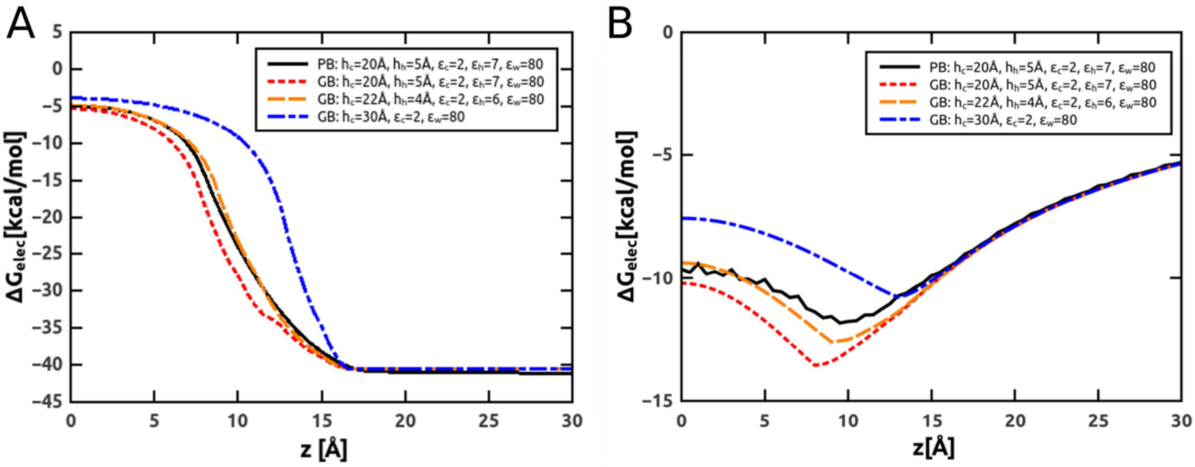


***Figure 5.3. Comparison of electrostatic solvation free energy terms of the SLIM model (GB) to Poisson-Boltzmann (PB) reference calculations using PBEQ.[212,213] Panel A shows the comparison for the total electrostatic solvation free energy of a single Ion with proton charge and radius 2.0 Å. Panel B compares the interaction terms of Equation (5.1) for the case of two ions with radii 2.0 Å. The position of the first ion is fixed in the center of the membrane, while the other ion is pulled through the membrane along the membrane normal with a closest distance of 4.0 Å perpendicular to the membrane normal. In the legend, $h_c$ is the thickness of the core region, $h_h$ the thickness of the headgroup region. The dielectric constants $\epsilon_{pc} = \epsilon_p = \epsilon_c$, $\epsilon_h$ and $\epsilon_w$ are according to Equation (5.1). The red dotted line is the SLIM model with the same parameters as in the PB model. The orange dashed line is the SLIM model with optimized thickness and dielectric constants to reproduce PB results. The blue dot-dashed line shows a model similar to that of Spassov et al,.[206] which uses only one dielectric slab.[210]***

To test the SLIM model for a more complex molecular geometry than a spherical ion, Julia Setzler and Carolin Seith also used the small alpha-helical protein Magainin (PDB code 2MAG[214]) to compare the self-terms of the electrostatic solvation free energy. Therefore, they removed all except for a single partial charge from the protein and set that single charge to that of a proton. Then they pulled the protein through the membrane in three different orientations with the charge located at the same position for every orientation and compared the electrostatic solvation free energies of the SLIM model to Poisson-Boltzmann calculations. As shown in Figure 5.4, they find that the SLIM model, in agreement with the Poisson-Boltzmann reference calculations, results in three different profiles corresponding to the three different orientations of Magainin. In addition, both models predict the orientation where Magainin is mostly inside the membrane to be energetically most favorable, while the orientation with Magainin mostly outside the membrane is least favorable. The quantitative agreement between results from SLIM and Poisson-Boltzmann calculations is also good.

This test demonstrated another important property of the SLIM model in contrast to the model of Tanizaki and Feig.[209] Because the single charge was always located at the same position for all three orientations of Magainin, Tanizaki's and Feig's approach of using a position-dependent dielectric profile function would predict all three orientations to have the same electrostatic solvation free energy. Thus, their approach would yield results that are not even in qualitative agreement with Poisson-Boltzmann calculations. In conclusion, the SLIM model provides more accurate electrostatic solvation free energies than previous generalized Born based implicit membrane models.[210]

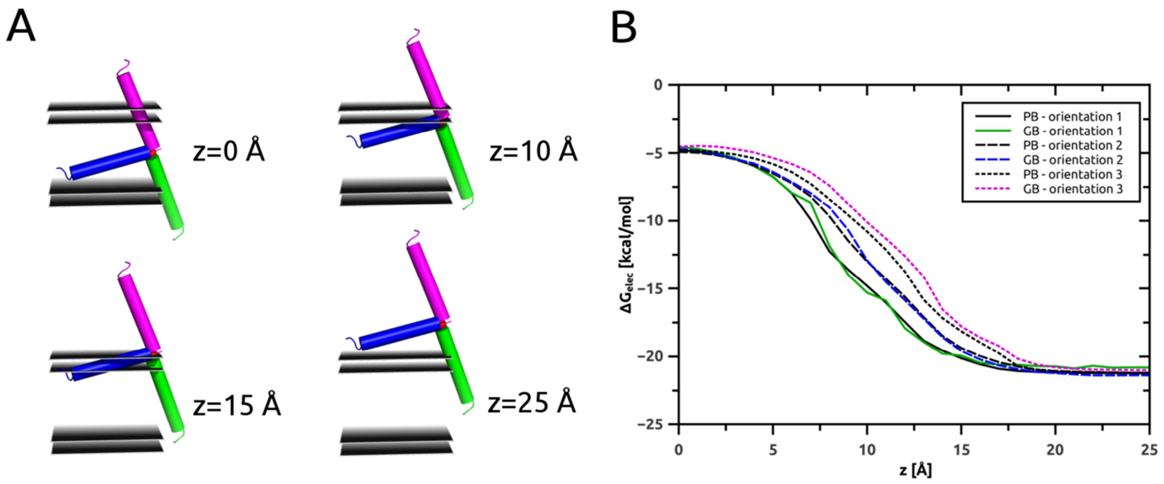


***Figure 5.4. Self-term comparison of the electrostatic solvation free energy between the SLIM model and Poisson-Boltzmann reference calculations for a more complex molecular structure. The small protein alpha-helical Magainin (PDB code 2MAG[214]) is used. Panel A illustrates the three orientations of Magainin's alpha helix shown as cylinders at four different positions relative to the membrane that were used for this comparison. The location of Magainin's single proton charge is shown by a red sphere. Panel B shows the electrostatic solvation free energy profiles of pulling three different oriented Magainin through the membrane. The colors for the SLIM graphs correspond to the orientations in Panel A.[210]***

Moreover, Julia Setzler and Carolin Seith demonstrated that SIMONA Monte Carlo simulations using the SLIM model are able to reproduce established properties of membrane peptides and small proteins. They investigated the distribution of the positions and orientations of the antimicrobial peptide Melittin relative to the membrane using Monte Carlo simulations with SLIM. They found two stable conformations that correspond to experimentally confirmed conformations. In addition, they found one stable set of conformations with a too strong kink in Melittin's alpha helix. This conformation has not been observed experimentally, but in other implicit or coarse-grained membrane simulations.[210]

Furthermore, Julia Setzler and Carolin Seith investigated the tilt angle of a single transmembrane domain of the M2 protein. They find varying tilt angles depending on the total thickness $h_\mathrm{m}$ of the slabs used to model the membrane and the value of the surface tension coefficient $\gamma$ in Equation (5.4). The transmembrane helix of the M2 protein tilts to overcome the unfavorable mismatch between the length of its hydrophobic alpha helical region and the thickness of the membrane. This behavior is in agreement with the concept of hydrophobic mismatch.[215,216] They also find that the SLIM model stabilizes the transmembrane region of the Glycophorin A dimer, with a crossing angle of the two alpha helices of the dimer in good agreement with experimentally observed values.[210]

Thus, the SLIM model provides an improved description of electrostatic solvation effects compared to previous generalized Born implicit membrane models and is able to reproduce some basic properties of small membrane peptides and proteins.[210]

## 5.4 A Parallel SLIM Implementation

Since the SLIM model implementation builds upon the PowerBorn method, its parallelization is rather trivial. The construction of the octree data structures inside the bounding boxes can be done in parallel with the method described in Section 3.3. However, there may be multiple bounding boxes in the SLIM algorithm, depending on the cases discussed in Section 5.2. To decrease load imbalance and idle time of the threads, I have rescheduled some functions of the octree construction method to reduce the number of synchronization points. These are points that all threads have to reach before any thread may continue. This rescheduling increases the efficiency of the parallelization. The analytical formulas in Equation (5.7) to (5.9) can be evaluated for each atom independently, hence they are also trivial to parallelize. In addition, the parallelization of the evaluation of Equation (2.35) for each of the generalized Born terms in Equation (5.1) can be done as described in Sections 3.1 and 3.3.

I have also performed speedup measurements for the SLIM model in analogy to Section 3.3. I have used three membrane proteins with an increasing number of atoms. These are Melittin, which contains 433 atoms (PDB code 2MLT[217,218]), the Glycophorin A dimer with 1322 atoms (PDB code 1AFO[219]) and a bacteriorhodopsin monomer containing 3538 atoms (PDB code 1FBB[220]). I measured the computation time of the SLIM model during a 10,000 step Monte Carlo simulation. The computation time includes the computation of both sets of Born radii and the evaluation of Equation (2.35) for both generalized Born terms in Equation (5.1), as well as the computation of the SASA for each atom as described in Section 3.2. The obtained speedups are shown in Figure 5.5. The algorithm scales well, as expected from the results of the power diagram and PowerBorn parallelization. With 32 available threads, speedups reach from 17.2 for the smallest system to 24.1 for the largest system. These results demonstrate that the parallel SLIM model is well suited for execution on modern multicore CPUs and that this parallelization significantly increases the amount of sampling that can be performed in a given amount of time.

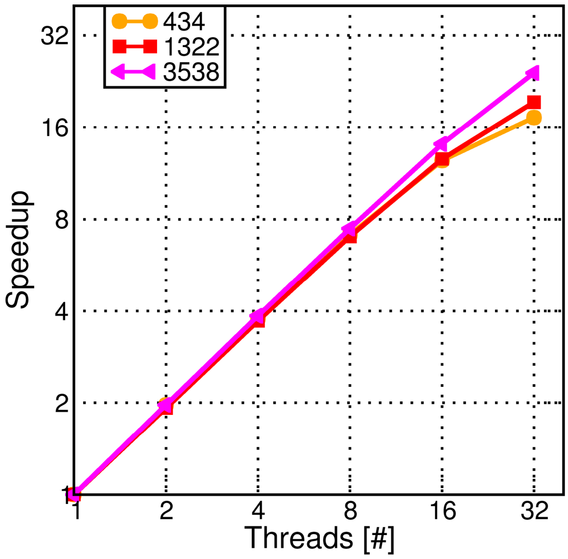


***Figure 5.5. Speedup of the computation time to evaluate the electrostatic solvation free energy in the SLIM model as a function of the number of threads for three different membrane proteins with an increasing number of atoms.***

## *5.5 Monte Carlo Simulation Performance of the SLIM Model*

Here I present results on the Monte Carlo simulation performance of SIMONA with the SLIM model. I have done the performance measurements in analogy to Section 3.4. For the measurements, I have used the same three proteins as in the previous Section 5.4. The required forcefield terms for a SIMONA Monte Carlo simulation with the SLIM model are listed in Table 5.1.

The simulation performance results are shown in Figure 5.6. Similar to the results for the implicit solvent model without the membrane discussed in section 3.4, the achieved speedup for the small system with 434 atoms, is lower than 16, e.g. the efficiency of the parallelization is below 50%. Thus, for such small systems, parallel simulations should not use more than 16 threads, while larger systems may also use more threads due to the achieved speedups being larger than 16.0 (Figure 5.6A). Again, the number of Monte Carlo steps that the simulation completes per day strongly depends on the number of atoms in the protein and the number of available threads. For a single thread, the Monte Carlo simulation of the smallest system completes 1.81 million steps per day, while the simulation of the largest system completes only 0.15 million steps per day (Figure 5.6B).

***Table 5.1. List of the forcefield terms that are necessary for the simulation of proteins in an implicit membrane environment. SASA is the abbreviation for solvent accessible surface area.***

| No. | SIMONA Forcefield term | Description | relevant sections or equations |
|---|---|---|---|
| **1** | Dihedral potential | Proper and improper dihedral potentials | Equations (2.9) and (2.10) |
| **2** | Nonbonded Vacuum | Lennard-Jones, Coulomb and 1-4 interactions | Equations (2.2)-(2.6) |
| **3** | SLIM | Computation of the SASA and Born radii, evaluation of $\Delta G_{\text{elec}}^{\text{SLIM}}(\epsilon_{\text{pc}}, V_{\text{pc}};\ \epsilon_{\text{h}}, V_{\text{h}};\ \epsilon_{\text{w}}, V_{\text{w}})$ | Sections 3.2 and 3.3, Equation (5.1) |
| **4** | NPSasaEnergyMembrane | Nonpolar solvation free energy for implicit membrane model | Equation (5.5) |

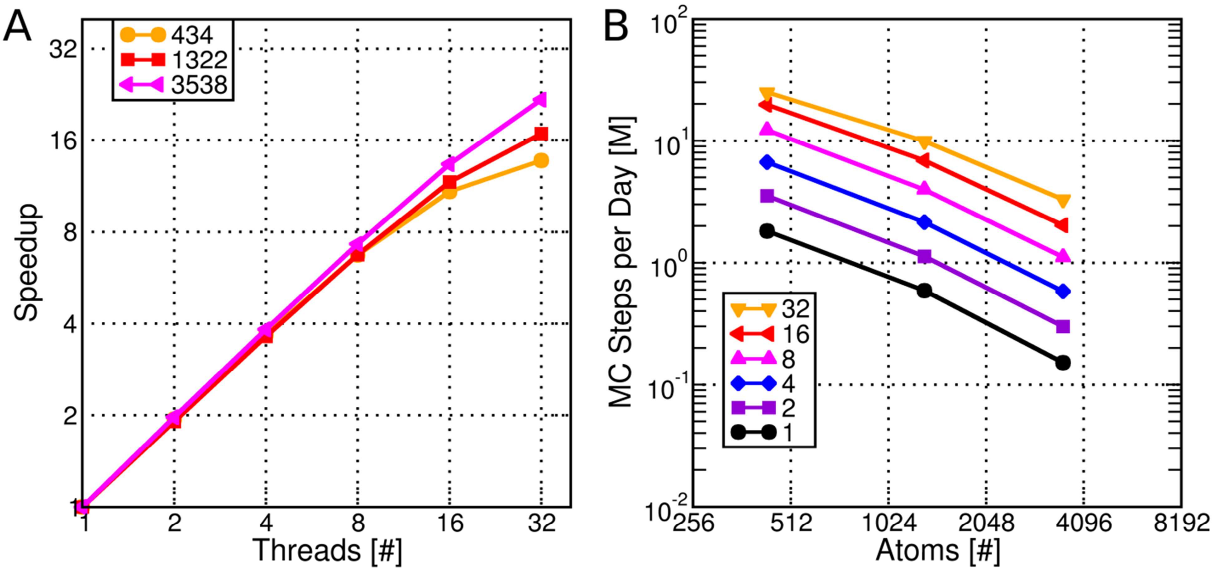


***Figure 5.6. Overall simulation performance measures for protein simulations in an implicit membrane environment. Panel A shows the speedup of the computation time over the number of threads for the complete Monte Carlo simulation for three different proteins with an increasing number of atoms. Panel B shows how many million Monte Carlo (MC) steps per day can be computed as a function of the number of atoms in the protein and the number of available threads.***

Tanizaki and Feig[221] reported molecular dynamics simulation of large integral membrane proteins using a different implicit membrane model and cutoffs on long-range interactions. Although they find that these cutoffs can have dramatic unphysical consequences on the orientation of membrane proteins, they deem simulations without cutoffs unfeasible due to the high computational cost. With cutoffs, their simulation of a bacteriorhodopsin monomer required 12 days for 500,000 molecular dynamics integration steps on two CPUs using CHARMM.[221]

I have also used this system as the largest for my performance measurements reported in Figure 5.6. In contrast to the molecular dynamics performance of Tanizaki and Feig,[221] the SIMONA simulation with my implementation of the SLIM model completes 300,000 Monte Carlo steps per day using two threads and without the need for any cutoffs (Figure 5.6B). Because the employed hardware in my simulation and that of Tanizaki and Feig differed strongly, I have carried out two test simulations with the model of Tanizaki and Feig using CHARMM and SIMONA with the SLIM model running on the same hardware. This test simulation again used the PDB 1FBB[220] as the starting conformation and ran for 1000 molecular dynamics or Monte Carlo steps respectively. More details about this test simulation can be found in the Appendix. A.3. CHARMM required 1021 s to complete the respective simulation, while SIMONA with the SLIM model required only 230 s. Thus, the SIMONA simulation with SLIM achieves about 4.4 times more simulation steps while removing the requirement for cutoffs at the same time. In addition, a SIMONA simulation with the SLIM model scale well up to 32 threads for not too small systems, increasing the simulation performance by another factor of up to 22 (see Figure 5.6A).

As in the case with the aqueous environment, the simulation performance will depend on the extent of the protein conformation. In addition, the position in relation to the membrane will also have a strong influence on the simulation performance. The reason is that, for protein regions inside the slab, no octree construction has to be performed. This saves a lot of computation time. In addition, if the bounding box of the protein is outside both slabs, only one numerical integration procedure inside the bounding box is required. This again reduces the computational cost. Therefore, the computational cost of evaluating the SLIM model for a given conformation varies even more than for the aqueous environment.

I conclude that the performance of my implemented forcefield terms listed in Table 3.1 is very well suited for the investigation of proteins in an implicit water or membrane environment. Especially they do not require any cutoffs of long-range interactions to yield the demonstrated performance. According to Feig and Tanizaki,[221] this is a large step forward in enabling realistic simulations of proteins and membrane proteins in implicit models.

# 6 A Monte Carlo Study of Protein Folding

This chapter contains an application of the methods that I developed and implemented to study the folding of the small protein FSD-EY using Monte Carlo simulations. The first section gives an introduction into the protein folding problem with regard to computational studies. The second section gives details about my Monte Carlo simulation setup. The third chapter presents first results of the simulations with regard to the efficiency of the employed parallel tempering method explained in Section 2.5. In the next section, I determine the folded state of FSD-EY in my simulation data and compare it to experimental NMR data. Afterwards, I determine the critical folding temperature of FSD-EY and a metastable conformation at low temperatures. Finally, I deduce FSD-EY's folding mechanism from the simulation data at the critical temperature.

## 6.1 The Protein Folding Problem and Computer Simulations

The large variety of functions that proteins can carry out relies on their unique feature to fold into clearly defined three-dimensional structures. The question how this three-dimensional structure is dictated by the amino acid sequence is known as the protein folding problem.[222] Nowadays, this large problem has been separated into three smaller problems:[222] What balance of forces determines the native fold? How can the native fold of a protein be predicted from its amino acid sequence? How do proteins fold into their native state? Especially the last question was recognized as one of the 100 biggest questions in science by the Science magazine.[223]

In principle, computer simulations can help to answer these questions. However, to have the computational means to carry out these investigations is a large challenge itself. The problem is the long timescale on which protein folding takes place. Kubelka et al.[224] investigated the lower limit on the folding time of a protein. They assume that the folding process can be described by a one-dimensional reaction coordinate and argue that there are two limiting factors for the folding time. At low temperatures, the limiting factor is the trapping of the system in local free energy minima that do not correspond to the folded state. At intermediate temperatures, the limiting factor is the crossing of the free energy barrier that separates the folded and unfolded state along the one-dimensional reaction coordinate. At high temperatures, the protein does not fold anymore. As a result, Kubelka et al. argue that the speed limit of protein folding is reached in the case where the free energy barrier vanishes, e.g. at a sufficiently high temperature. They further argue that for a protein consisting of $N_{\text{res}}$ residues, the lower folding time limit is $N_{\text{res}}/100$ µs. However, they also note that even the known ultrafast folding proteins take much longer to fold.

Since typical molecular dynamics simulations can only reach the low microsecond range,[24,225] they are unable to thermodynamically characterize the folding process unless the protein is very small. One solution to the problem is to employ rare specialized supercomputers.[32,33] However, even this approach failed to study the folding of a moderate sized protein due to the limiting time the system could be simulated.[34] Another approach is to use replica exchange molecular dynamics.[226] This is the molecular dynamics extension of parallel tempering, see Section 2.5. Since the system is simulated at different temperatures, one such replica may likely be close to the temperature where the free energy barrier vanishes. Thus, the folding time of the protein at this temperature is minimal. This has enabled the study of the folding process of a few small peptides and proteins.[222,227–237] Nevertheless, the folding speed limit still applies, wherefore even replica exchange molecular dynamics will eventually fail to investigate the folding of medium sized proteins with complex topology. In addition, a recent investigation showed that replica exchange molecular dynamics in explicit water only increases the efficiency of conformational sampling by a factor of two over multiple conventional molecular dynamics simulations.[238]

Coarse-grained models have also been used to study protein folding.[26,239,240] Since they average out the fast degrees of motion, such as atomic vibrations, they allow the use of much larger timesteps in molecular dynamics. However, they are usually used in conjunction with Brownian dynamics, which does not allow for the computation of thermodynamic expectation values.[26] In addition, the conversion to an all-atom representation is required to extract the atomistic mechanisms of protein folding.[26] Therefore, I will not consider them further.

Given these circumstances, my Monte Carlo simulation methods promise to fill this gap of a computational method that can investigate the folding process of a protein independent of its folding time at an all-atom level. First studies on a small protein consisting of three alpha helices showed that this promise is well founded.[241,242] Here I investigate the folding of another small protein. In contrast to the previous studies, I will focus on a small protein that contains a mixture of secondary structure elements. This investigation should provide further insights into the folding of small proteins. In addition, the mixture of secondary structure elements provides a larger challenge for my implicit solvent model, as some previous implicit solvent models have been show to favor some secondary structure elements over others.[243,244]

## 6.2 Monte Carlo Simulation Setup for Folding of the FSD-EY Protein

The protein I investigate is FSD-EY (PDB code 1FME[42]). It has a beta-beta-alpha fold. The beta sheet has hydrogen bonds between residue pairs 5 and 12 as well as 7 and 10. I will refer to the residues 5 to 6 as the first beta strand region and the residues 10 to 12 as the second beta strand region. The alpha helix contains the residues 15 to 24. Figure 6.1 shows a cartoon representation of the FSD-EY from three different perspectives. To investigate the folding of this protein, I ran a parallel tempering Monte Carlo simulation using SIMONA[37] with code revision number 3762. I used my implemented forcefield terms and developed implicit solvent model described in Chapter 3 as well as the parallel tempering algorithm introduced in Section 2.5. The parallel tempering algorithm contained 32 different temperatures distributed exponentially between 250 K and 500 K as shown in Figure 6.2A.

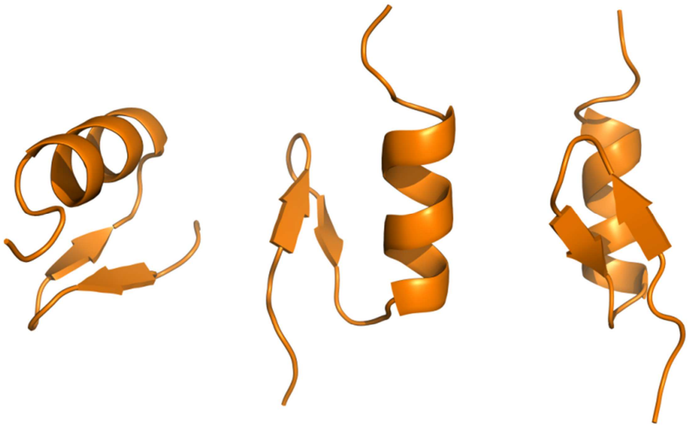


***Figure 6.1. This is the cartoon representation of the protein FSD-EY (PDB Code 1FME[42]) viewed from three different perspectives. The protein has two beta strands forming one beta sheet and an alpha helix at the C-terminus.***

The simulation was run in parallel at each temperature. The parallel tempering algorithm attempted an exchange of the temperatures and saved a snapshot of the simulation after every 10,000 Monte Carlo steps. Every temperature used eight threads to evaluate the energy of the current configuration. In total, the simulation performed 200 million Monte Carlo steps at each temperature. The simulation ran about 9.1 million Monte Carlo steps per day. Thus, the total simulation took about 22 compute days to complete while running on 256 compute cores of the HERMIT cluster at the HLRS Stuttgart.

During the simulation, only dihedral degrees of freedom were considered, while bond lengths and angles held constant. For a new proposal configuration, one dihedral angle was rotated relative to its current position by a value chosen from a Gaussian distribution with 20 degrees width and zero mean. Perturbing any backbone dihedral angle was twice as likely as any side chain dihedral angle. In addition, I used so-called local moves that are implemented in SIMONA[37] during the simulation.

These perturb six succeeding backbone dihedral angles, while leaving the remaining protein unchanged. In addition, the protein was free to perform rigid rotations to average out the discretization errors described in Section 3.3.

The starting structure for the simulation was the first model deposited in the PDB entry 1FME[42]. The program pdb2gmx generated the forcefield parameters of the AMBER99SB*-ILDN forcefield. The corresponding atomic radii of the forcefield for the GBOBC method[120] and a probe radius of 1.4 Å were used to define the solvent excluded and solvent accessible surfaces. For all temperatures, I used the same dielectric constant $\epsilon_{\mathrm{w}} = 80.0$ in Equation (2.35), and I use a global surface tension $\gamma = \gamma_i = 5.42\ \mathrm{cal/mol/Å^2}$ in Equation (2.41).[183] Due to time constraints, I was not able to implement the temperature dependence of these values into the simulation. The dielectric constant for the protein interior was assumed to be $\epsilon_{\mathrm{s}} = 1$. The initial structure of FSD-EY was minimized energetically using GROMACS to relax unusual bond lengths and bond angles. The resulting structure is the starting conformation for my parallel tempering simulation at each temperature.

## 6.3 Parallel Tempering Simulation Characteristics

According to Bittner et al., each replica should spend the same amount of time at each temperature present in a parallel tempering simulation for it to be most efficient.[245] Therefore, the first analysis is devoted to the course of temperatures for the replicas during my parallel tempering simulation of FSD-EY. Figure 6.2A graphs the exponential distribution of the starting temperatures, as well as the average temperatures, sorted from smallest to largest, and their standard deviations of each replica that resulted from the parallel tempering simulation. I observe that the resulting average temperatures do not follow the exponential distribution of the starting temperatures. The 15 lowest average temperatures all have average values below 326 K. The average temperatures for these replica increase moderately for the lowest three temperatures and slowly for the remaining 12 temperatures. Then there is a jump in the average temperature from 326 K up to 379 K. The average temperature for the remaining replica increases moderately up to 444 K. As a result, the simulation seems to be inefficient according to Bittner et al., because if all replicas would have spent the same amount of time at each temperature, they should have the same average temperature.

The probability to exchange two temperatures during the parallel tempering simulation is shown in Figure 6.2B. It lies between 0.7 and 0.9 for all temperatures. The probability is lowest at temperatures that lie between 325 and 375 K. This temperature range coincides with the jump in the average temperature (Figure 6.2A). Nevertheless, the exchange probabilities are very high for all temperatures. According to Deem and Earl,[136] their values should be between 20% and 23% to yield the largest computational efficiency. This suggests that the temperatures could be spaced even more widely to save computational effort.

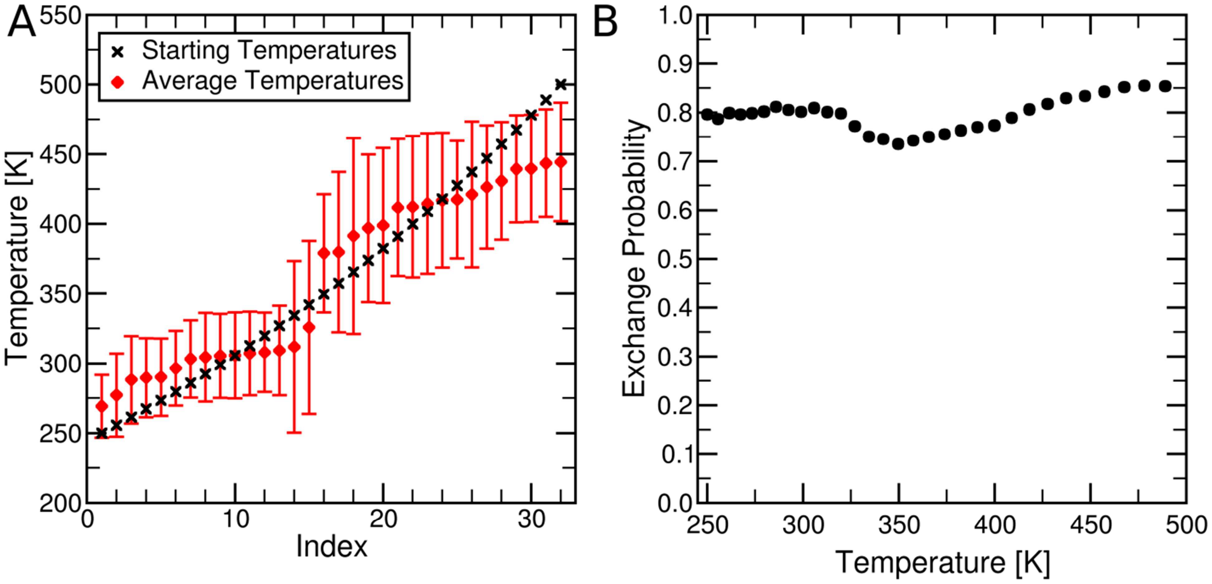


***Figure 6.2. Panel A shows the starting temperatures of the FSD-EY parallel tempering simulation (black crosses) and the resulting average temperatures and their standard deviations sorted by average temperature after 200 million Monte Carlo steps (red diamonds with error bars). Panel B shows the probability to exchange the temperature between two replicas with adjacent temperatures.***

The standard deviation of the temperature in Figure 6.2 is another indicator how much the different replica move in temperature space. It is lower than 32 K for the 13 replicas with the lowest average temperature. It increases up to 70 K for replicas that are close to the jump in the average temperature and then decreases again down to 38 K for replicas with larger average temperatures than 379 K. Figure 6.3 graphs the course of temperature during the parallel tempering simulation for the replica with the lowest and highest standard deviation. In the former case, the temperature shortly increases and then drops towards the lowest temperature during the first 6 million Monte Carlo steps. Afterwards, the temperature fluctuates at low values with short spikes up to 360 K. In the latter case, the temperature strongly fluctuates between 250 and 420K during the first 70 million steps. Then it increases to very high values and fluctuates between 350 and 500 K up to 170 million steps. In the last stage, the temperature drops to a medium range and fluctuates around 350 K. The results from Figure 6.3 show that the round-trip time of a replica from the lowest temperature to the highest temperature and back is very high. Because these round-trip times are another indicator of the efficiency of the parallel tempering algorithm,[136] they also suggest that the parallel tempering simulation seems to be rather inefficient. According to Bittner et al.,[245] this behavior can be caused by phase transitions of the studied systems, where each phase transition corresponds to a barrier that hinders replicas to travel through temperature space. This would also explain the jump in the average temperature in Figure 6.2. Bittner et al. further showed that to lower these barriers, one has to increase the number of Monte Carlo steps between temperature exchanges, or to increase the speed by which the system can move through phase space. Besides an optimized temperature distribution, future studies should investigate how the round-trip times can be reduced by changing the sets of Monte Carlo moves and the number of steps between the attempted exchanges of temperatures. Nevertheless, the simulation was long enough for some replicas to visit all temperatures at least once, which indicates that the conformational space of FSD-EY was sampled thoroughly.

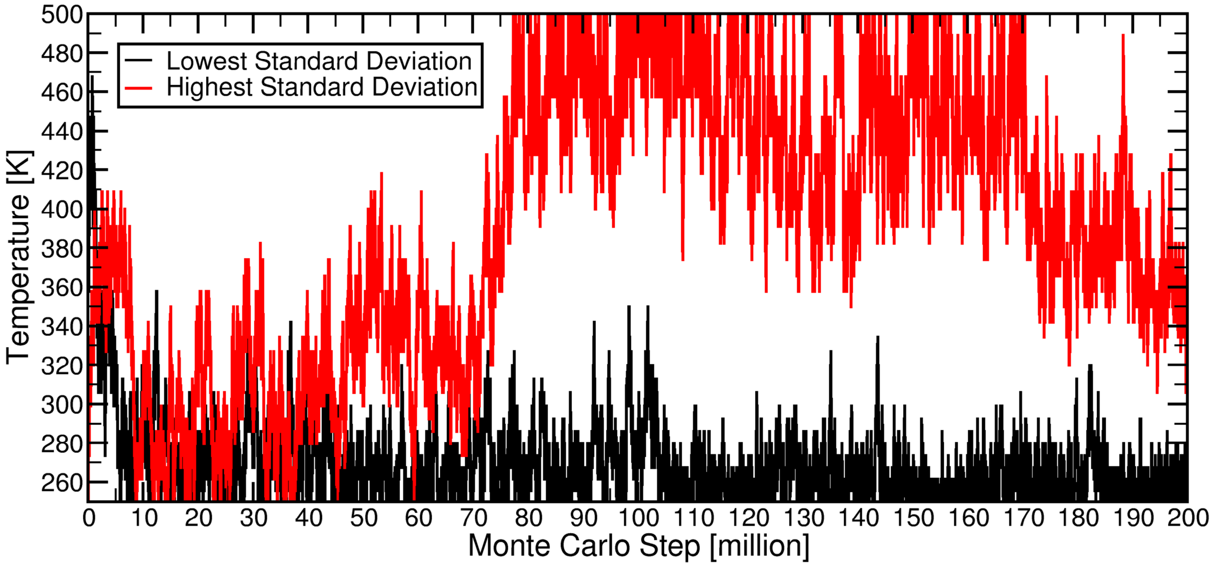


***Figure 6.3. Course of the temperature for two replicas during the parallel tempering simulation of FSD-EY. The black line shows the course of temperature for the replica with the lowest standard deviation of temperature and the red line the corresponding graph for the replica with the highest standard deviation of temperature.***

## 6.4 Comparison of the Simulated and Experimental Folded State

Now I turn to the comparison of the folded state between the NMR ensemble of FSD-EY and the simulated ensemble. This comparison yields insights into the accuracy of the biomolecular forcefield and the implicit solvent model. For this comparison, one requires a measure that is able to differentiate between the folded and unfolded state. The root mean square deviation (RMSD) between two conformations of a protein is the minimum of the root of the mean squared distance between corresponding atoms of the two conformations. To minimize this value, one can rigidly rotate and translate one of the conformations. Kabsch proposed a method to compute this best rotation and translation that minimizes the RMSD.[246,247] I will use this as a similarity measure for different protein conformations. This measure is also used in other studies of protein folding.[32–34,227–230,232–235]

To determine a single representative folded conformation from the NMR ensemble of FSD-EY in the PDB entry 1FME[42], I have performed a cluster analysis of this ensemble. A clustering algorithm finds groups of conformations that have a low RMSD to each other but a high RMSD to conformations of all other groups. It is implemented in the g_cluster program of the GROMACS package.[25] The RMSD considered all non-hydrogen atoms and used a cutoff of 2.0 Å as the minimal distance between two conformations of neighboring clusters. The analysis of PDB entry 1FME[42] resulted in only one cluster. The fifth model in that PDB entry was closest to the center of the cluster according to the program, which means that its conformation has the lowest average root mean square deviation to all other conformations in the cluster. I will refer to its conformation as the NMR reference conformation of FSD-EY.

Subsequently, I have performed the same cluster analysis for my simulated ensemble. Since the temperature dependence of the dielectric constant and hydrophobic effect of water were neglected, I have only considered the replica at 292.36 K. The corresponding dielectric constant of water at that temperature is closest to the used value of $\epsilon_{\mathrm{w}} = 80.0$ according to Equation (2.27). Because the clustering algorithm is very compute intensive, as it scales quadratically with the number of conformations to cluster, only every fourth snapshot in the simulated ensemble was considered for clustering.

This cluster analysis of the simulated ensemble resulted in 15 clusters that contain five or more conformations and only 4 clusters with more than ten conformations out of the 5000 conformations that were considered for clustering. The largest cluster contains 3777 conformations and is 7.1 times larger than the second largest cluster. Since the protein should have a stable folded state at this temperature,[42] I define this largest cluster to represent the folded state of FSD-EY in my simulations. I will refer to the central conformation of this cluster as the simulated folded conformation. Figure 6.4 shows a comparison of the simulated folded conformation to the NMR reference conformation. In general, the agreement between the two conformations is good. The RMSD of the C-alpha atoms ($\mathrm{RMSD}_{\alpha}$) is 2.7 Å, and the RMSD of all atoms including hydrogen atoms is 4.2 Å. I observe some deviations between those two conformations in the C-terminal region at the end of the alpha helix. The helix of the simulated folded conformation is one residue longer than that of the NMR reference conformation according to STRIDE.[47] In addition, the beta sheet is also one residue longer in the simulated folded conformation than in the NMR reference conformation. There are also small deviations in the loop linking the two strands of the beta sheet.

A prominent difference between the two conformations is the conformation of the side chain of residue Tyr$_7$, which is part of the FSD-EY's hydrophobic core (Figure 6.4). This residue is located at the N-terminal end of the loop linking the two beta strands. The side chain flips and does not point towards the side chains of residues Leu$_{18}$ and Ile$_{22}$ as in the reference conformation, but towards the side chain of Phe$_{25}$. The former two side chains show also moderate differences in their conformations between the simulated folded conformation and the NMR reference conformation. The flip of Tyr$_7$ is likely the reason for the different loop conformations between the beta strands, because this residue was shown to be very important for the loop conformation in experiments.[42] The different side chain conformations result in a higher exposition of the hydrophobic residues Leu$_{18}$ and Ile$_{22}$ to water in the simulated ensemble. Because of the hydrophobic effect explained in Section 2.4, the nonpolar term of the implicit solvent model should disfavor such conformations energetically. This suggests that my chosen surface tension coefficient of $\gamma = 5.42\ \mathrm{cal/mol/Å^2}$ in Equation (2.41) may be too small. Another possibility is that the torsion potentials of the corresponding side chains are not accurate enough. This possibility can be checked by using the CHARMM22*, which was able to fold this protein in explicit water up to very high accuracy.[33] In any case, further studies will be necessary to improve the implicit solvent model and force field so that the simulated ensemble of my Monte Carlo method agrees even better with the NMR ensemble.

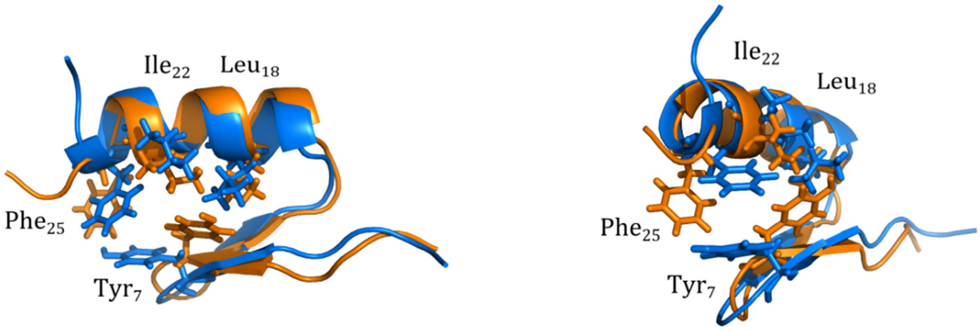


***Figure 6.4. Comparison of the NMR reference conformation of FSD-EY (PDB 1FME[42], model 5) (orange) and the central conformation of the largest populated cluster from my parallel tempering simulation at 292.36 K (blue) viewed from two different perspectives. In addition to the cartoon representation, the hydrophobic core's side chains of residues Tyr$_7$, Leu$_{18}$, Ile$_{22}$, and Phe$_{25}$ are highlighted by a stick representation because of their differences in the two conformations.***

Nevertheless, I would like to point out that I have exchanged the recommended water model TIP3P of the AMBER99SB*-ILDN forcefield with my implicit water model without any other modifications of the forcefield. In addition, the employed set of atomic radii was straightforward available, although there may be other sets such as that by Swanson et al.[77] that may perform better. Taking these facts into account, the agreement between the simulated folding state and the experimentally determined folded state is satisfying.

## 6.5 Determination of FSD-EY's Critical Folding Temperature

Now I will focus on the folding transition of FSD-EY, especially the determination of the critical folding temperature at which the minima of the folded and unfolded states along a given reaction coordinate have equal free energy.[248] Because my simulations do not account for the temperature dependence of the solvation free energy, this investigation will not present a physically and quantitatively correct picture of the folding process. The error of the electrostatic contribution to the solvation free energy can be estimated from Equations (2.27) and (2.35). Assuming $\epsilon_S = 1$, this contribution's absolute value decreases by 0.67% when increasing the temperature from 273.15 K to 373.15 K. The temperature dependence of the nonpolar contribution is more complex and not a monotonic function of temperature,[91,249–251] wherefore no trivial error estimate is possible. Although the relative changes seem to be larger, the absolute value of the nonpolar contributions to the solvation free energy for proteins is usually much smaller than that of the electrostatic contribution. Due the marginal stability of proteins,[252] even small changes of the solvation free energy can have significant effects. Nevertheless, this investigation should suffice to demonstrate that my Monte Carlo methods allow an efficient study the folding process of small proteins.

To find the critical folding temperature, I will use the RMSD$_\alpha$ as a reaction coordinate. A second reaction coordinate that can describe protein folding is the fraction of established native secondary structure $Q_{SS}$ as determined by STRIDE.[47] I will take all secondary structure elements listed in Table 2.1 into account, even coil. Therefore, $Q_{SS}$ decreases due to the formation of additional helices or beta sheets. In contrast to the previous Section 6.4, I will use the simulated folded conformation as the reference for calculating RMSD$_\alpha$ and $Q_{SS}$ instead of the NMR reference conformation, because I expect the RMSD$_\alpha$ and $Q_{SS}$ to provide better reaction coordinates for the folding process with this new reference conformation.

To determine the critical folding temperature, I have projected my simulated ensembles at the different temperatures onto these two reaction coordinates and counted the number of conformations that fall within a given bin of the reaction coordinate. The widths of the bins are 0.5 Å and 0.1 for RMSD$_{\alpha}$ and $Q_{\mathrm{SS}}$ respectively. Since the ensembles generated by the Monte Carlo simulations are representative, the conformation count can be converted to a free energy landscape $\Delta G(Q)$ that is projected onto an arbitrary reaction coordinate Q. This reaction coordinate is separated into bins $Q_i$. The free energy $\Delta G(Q_i)$ for such a bin $Q_i$ is

$$\Delta G(Q_i) = -RT\ln\left(\frac{\mathrm{n}(Q_i)}{\mathrm{n}_{\max}(Q)}\right). \tag{6.1}$$

Here, $\mathrm{n}(Q_i)$ is the number of conformations in the ensemble that fall into bin $Q_i$, $\mathrm{n}_{\max}(Q)$ is the maximum number of conformations in any bin, $R = k_B N_A$ is the gas constant, and $T$ is the temperature of the system. Per definition of this free energy landscape, the most populated bin will have zero free energy, while all other bins have free energies larger or equal than zero.

Figure 6.5 shows the free energy landscapes for RMSD$_{\alpha}$ and $Q_{\mathrm{SS}}$ of four selected temperatures, 273.39 K, 292.36 K, 349.62 K, and 365.61 K. The first one is the lowest temperature of the replicas that is still above the melting temperature of water, while the last temperature is the highest temperature still below the boiling point of water. The second temperature was used in Section 6.4 to compare the experimental and simulated folded conformation. The importance of the third temperature will be discussed later. For the computation of the four free energy landscapes, I have neglected the first 20 million Monte Carlo steps from each ensemble. This should account for the equilibration of the simulation, since all replicas started with the same conformation. In Figure 6.5A, I observe for RMSD$_{\alpha}$ that the free energy minimum for the three lowest temperatures is between 2.0 and 2.5 Å, while the minimum of the highest temperature is at 8.5 to 9.0 Å. For the fraction of native secondary structure, the replicas with the lowest three temperatures have their free energy minimum in the range of 90% to 100% native secondary structure. The minimum for the highest temperature is 60% to 70% native secondary structure. These observations are consistent with the fact that protein folds are stabilized at low temperatures and become unstable at high temperatures. The $Q_{\mathrm{SS}}$ minimum of the highest temperature replica at $Q_{\mathrm{SS}} > 60\%$ shows that there is a considerable fraction of the native secondary structure still present in the unfolded state.

I find that FSD-EY's critical temperature of folding is slightly above 349.62 K wherefore I have selected the data from the corresponding replica to be present in Figure 6.5. For RMSD$_{\alpha}$, the free energy difference between the folded and unfolded state is 0.27 kcal/mol. The corresponding free energy difference for the fraction of established native secondary structure is 0.05 kcal/mol. These free energy differences are higher for any other replicas, wherefore the replica at 349.62 K is closest to the critical folding temperature. The free energy barrier heights between the folded and unfolded states are 1.4 kcal/mol and 0.7 kcal/mol for RMSD$_\alpha$ and $Q_{SS}$ respectively.

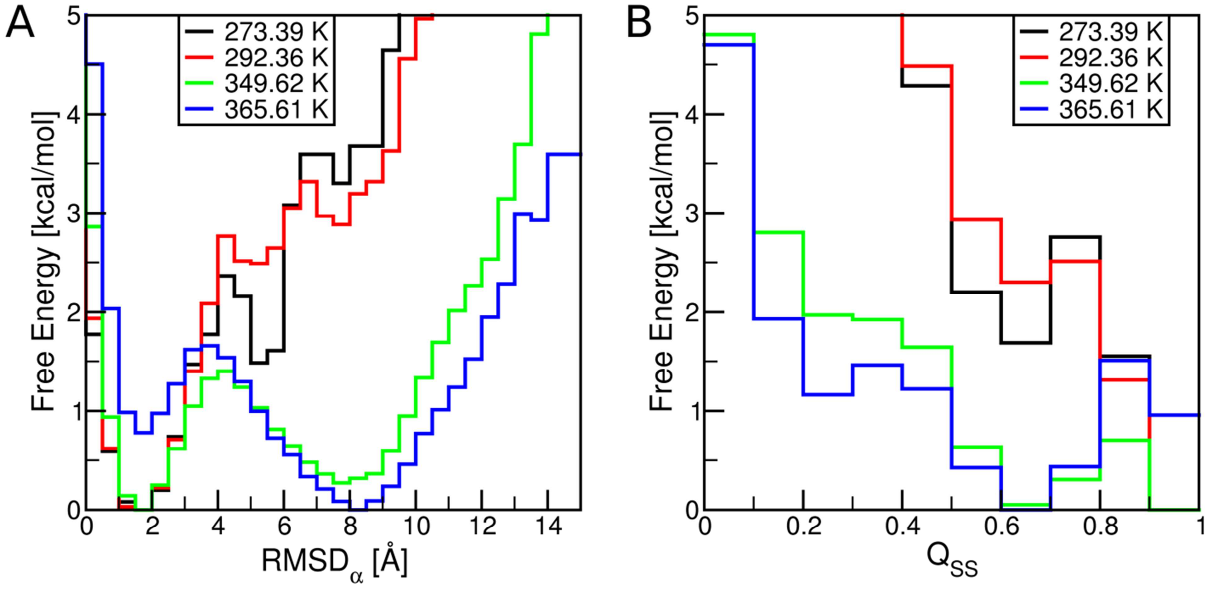


***Figure 6.5. Free energy landscapes generated from parallel tempering Monte Carlo simulation of FSD-EY projected onto the C-alpha RMSD (panel A), and the fraction of established native secondary structure $Q_{SS}$ (panel B) for four selected temperatures.***

Molecular dynamics simulations of this protein in explicit water using the special purpose computer Anton resulted in a folding free energy of 0.7 kcal/mol at 325 K with the unfolded state already being the global free energy minimum.[33] Thus, FSD-EY is more stable in my implicit water model with the AMBER99SB*-ILDN forcefield than in the explicit water simulations of Lindorff-Larson et. al[33] using the CHARMM22*[253,254] forcefield. The explicit water molecular dynamics simulations of Lindorff-Larsen et al. found only very little secondary structure in the unfolded state.[33] This is in contrast to my results. One reason for this discrepancy may be the different forcefield used by Lindorff-Larsen et al. Another likely reason is the neglected temperature dependence of the solvation free energy in my implicit solvent model. Other reasons might include neglected degrees of freedom in my Monte Carlo simulations, such as vibrations of the bond lengths and angles. The matter of helix stability in my implicit solvent simulations should be looked into in further studies. Best and Hummer have shown how to tune forcefields to achieve better agreement to NMR experiments for the fraction of established secondary structure in small peptides.[40] This is likely a good starting point for further improving my Monte Carlo simulation approach.

## 6.6 A Low Temperature Metastable State of FSD-EY

Interestingly, the low temperature replicas, e.g. at 273.39 K, show a metastable conformation with a local free energy minimum at about 6 Å RMSD$_\alpha$ (Figure 6.5A). To find the conformation that corresponds to this metastable state, I have again performed a cluster analysis as described in Section 6.4 of the ensemble from the replica at 273.39 K. The two largest clusters that resulted from this analysis have 3309 and 1166 conformations respectively, and the third largest cluster has 217 conformations. The two former clusters correspond to the folded state because their central conformations have RMSD$_\alpha$ values with respect to the simulated folded conformation of 0.89 Å and 2.3 Å respectively. These values lie in the free energy minimum of the folded state (Figure 6.5A). However, their central conformations differ in the orientation of the N-terminal region. The third cluster's central conformation has a RMSD$_\alpha$ of 5.6 Å. This value is in agreement with the local free energy minimum of the metastable state observed in Figure 6.5A. Thus, I will refer to this conformation as the metastable conformation.

Figure 6.6A shows a comparison of the metastable conformation to the simulated folded conformation. In contrast to the folded state, the residues of the first beta strand are detached from those of the second beta strand, thus disrupting the beta sheet. Instead, residues 3 to 5 of the first beta strand form a tight 3-10 helix in the metastable conformation according to STRIDE.[47] Figure 6.6B visualizes that this tight helix allows the packing of the aromatic rings of residues Tyr$_3$ and Tyr$_7$ against the hydrophobic core of the protein. This packing effectively shields the hydrophobic core from water, wherefore the metastable conformation is energetically favorable.

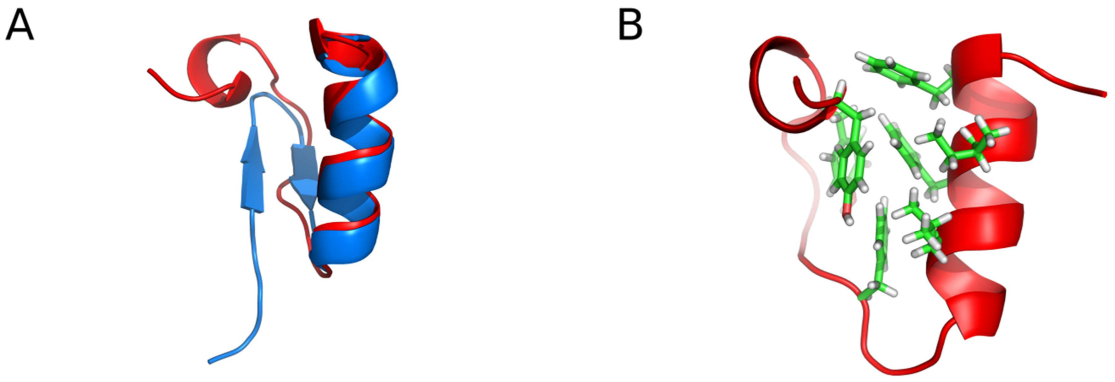


***Figure 6.6** Panel A shows the metastable conformation (red) occurring at low temperatures in comparison to the central conformation of the simulated folded conformation at 292.36 K (blue). Panel B shows the packing of the two residues Tyr$_3$ and Tyr$_7$ against the hydrophobic core in the metastable conformation. The color code of the side chains is according to Figure 2.3. The red oxygen atoms of the Tyrosine side chains mark these two residues on the top left of Panel B.*

However, it disappears at higher temperatures of about 292.36 K (Figure 6.5A). Since my simulations neglected the temperature dependence of the solvation free energy, the only reason for its disappearance can be that the entropy of the simulated folded conformation with its beta sheet must be higher than that of the metastable conformation. The higher entropy of the former conformation results in a larger free energy difference to the metastable conformation at higher temperatures. Because the unfolded state is characterized in general by high entropy, the free energy difference between the metastable conformation and the unfolded state also favors the unfolded state with increasing temperature. Therefore, the occurrence of the metastable conformation vanishes at higher temperatures, in agreement with the data of Figure 6.5.

The explicit water molecular dynamics simulation of Lindorff-Larsen et al.[33] also showed a metastable conformation between the folded and unfolded state. Although they do not investigate this conformation in detail, its RMSD$_\alpha$ is 3.0 Å larger than the free energy minimum of their folded state (Supporting Information by Lindorff-Larsen et al.[33]). This RMSD$_\alpha$ difference agrees with my data, where the folded state minimum is at 2.0 Å to 2.5 Å (see Section 6.5) and RMSD$_\alpha$ of the metastable state is 5.6 Å. However, I note that the different reference structures were used to compute the RMSD. Nevertheless, this agreement indicates that their and mine metastable conformation corresponds to each other.

## *6.7 Deduction of FSD-EY's Folding Mechanism*

While the previous investigation in Section 6.5 yielded some thermodynamic characteristics of the FSD-EY's folding process, it did not yield insights into the structural changes during the folding process. Since the Monte Carlo ensemble does not provide time resolved trajectories of the folding process, the deduction of these structural changes is not straightforward.

I will focus on the secondary structure first. As already noted in Section 6.5, Figure 6.5B suggests that there is a high fraction of native secondary structure left in the unfolded state. The first task is to identify what fraction of native secondary structure remains in the unfolded state. I have computed the number of residues in native beta sheets $N_\mathrm{E}$ and native alpha helices $N_\mathrm{H}$ for the ensemble close to the critical folding temperature at 349.62 K. I also computed these values for subsets of the ensemble that have RMSD$_\alpha$ values larger than 3.0 Å, 5.0 Å, and 9.0 Å respectively. I have converted the probabilities of specific pairs $N_\mathrm{E}$, $N_\mathrm{H}$ in analogy to Equation (6.1) to free energies

$$\Delta G(Q_i, P_j) = -RT\ln\left(\frac{\mathrm{n}(Q_i, P_j)}{\mathrm{n_{max}}\,(Q, P)}\right). \tag{6.2}$$

Here, Q and P are different observables separated into bins $Q_i$ and $P_j$. Again, $\mathrm{n_{max}}\,(Q, P)$ is the maximum number of conformations for any possible pair of bins $Q_i, P_j$ so that the most probable pair has zero free energy. Since the smallest alpha helix and beta sheet consist of four residues each, the pairs with lower respective values but larger than zero are unpopulated in the graphs in Figure 6.7. Moreover, beta sheets can grow only in pairs, wherefore the pairs with $N_\mathrm{E} = 5$ are also not populated.

Figure 6.7 illustrates that with increasing RMSD$_\alpha$, the free energy of conformations with beta sheets in the ensemble significantly increases while the free energy of finding only alpha helices stays approximately constant for RMSD$_\alpha$ cutoffs lower than 9.0 Å. These data proof that the remaining fraction of the native secondary structure in the unfolded state corresponds the native alpha helix of FSD-EY, while the probability of finding native beta sheets in the unfolded state is negligible. Therefore, the alpha helix is already present when the beta sheet is not in conformations of the unfolded state. As a result, these data suggest that the first step of FSD-EY's folding mechanism with my employed forcefield and implicit solvent model is the formation of the alpha helix.

I also observe that the main part of the beta sheet content in the ensemble disappears when the RMSD$_\alpha$ cutoff increases from 3.0 Å to 5.0 Å. According to Figure 6.5A, the free energy barrier of folding is located in this RMSD$_\alpha$ region. This suggests that the free energy barrier of the folding of FSD-EY is due to the formation of the beta sheet.

The obvious question how the folding of FSD-EY continues after the formation of the alpha helix is, how the beta sheet forms and attaches to the alpha helix. There are three possible scenarios. In the first scenario, the beta sheet forms first and subsequently attaches to the alpha helix. Thus, the secondary structure forms before the hydrophobic collapse of the protein happens. In the second scenario, the region of the first beta strand (counting from the N-terminus) aligns to the helix and afterwards the second beta strand region attaches to the helix and forms the beta sheet. In the third scenario, the order of attaching the beta strand regions to the alpha helix is exchanged. The second beta sheet region attaches to the alpha helix first, while the first beta sheet region is still free to diffuse around. Subsequently, the region of the first beta sheet attaches to the helix and second strand to form the folded conformation.

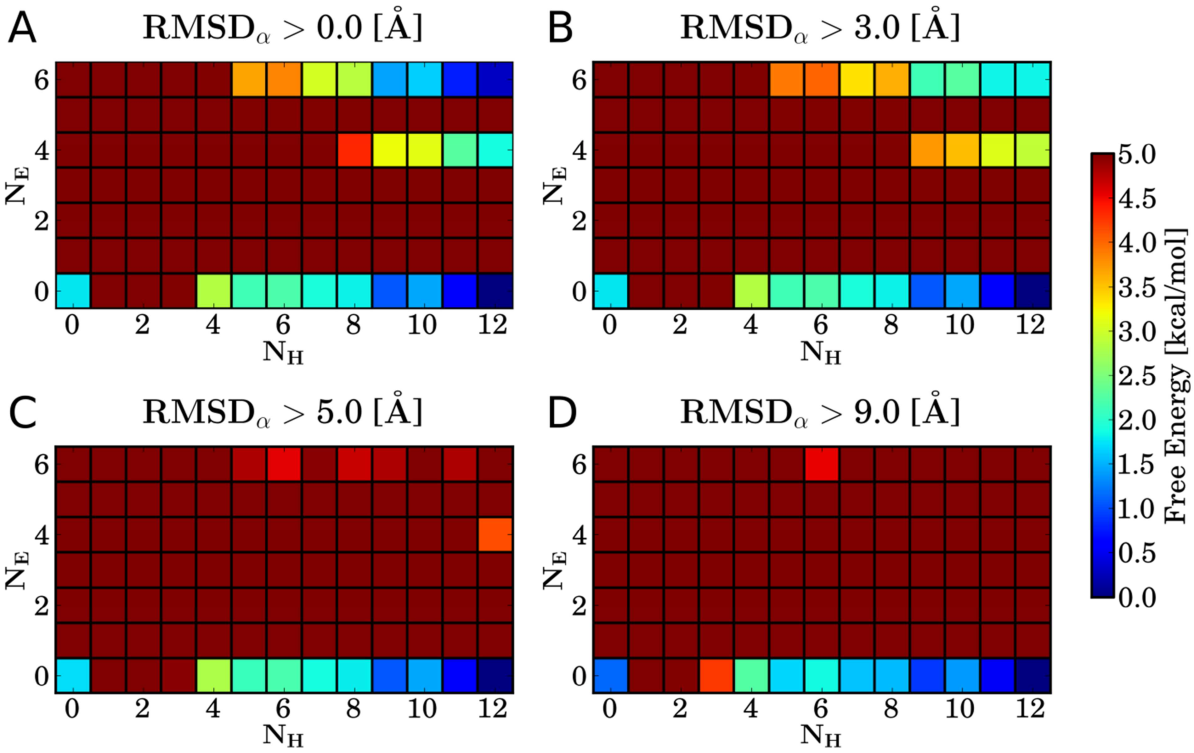


***Figure 6.7. Free energies of having $N_E$ residues forming part of the native beta sheet of FSD-EY and $N_H$ residues forming part of the native helix of FSD-EY. The ensemble for these data is the subset of the simulated ensemble at 349.62 K with $RMSD_\alpha$ larger than 0.0 Å (panel A), 3.0 Å (panel B), 5.0 Å (panel C), and 9.0 Å (panel D).***

To investigate these three scenarios, I have chosen three different atoms pairs of FSD-EY as distance measures within the protein. These three contacts are visualized in Figure 6.8 in the simulated folded conformation. Table 6.1 summarizes the contacts and shows their average distance and standard deviation. The first of these contacts measures the distance between the CB atom of residue $\mathrm{Arg}_{10}$ and the CZ atom of $\mathrm{Phe}_{21}$. In the simulated folded conformation, the former atom marks the N-terminal end of the second beta sheet, and the second atom is at the center of the alpha helix and points towards the former atom. The second contact measures the distance between the CZ atom of $\mathrm{Phe}_{12}$ and the CG atom of $\mathrm{Phe}_{21}$. In the simulated folded conformation, this contact is a measure between the distance of the C-terminal end of the second beta sheet and the alpha helix. The third contact measures the distance between atoms CB of $\mathrm{Ala}_5$ and CG of $\mathrm{Leu}_{18}$. This contact measures the distance between the N-terminal end of the first beta strand and the N-terminal end of the alpha helix.

***Table 6.1. List of contacts between atom pairs that are used to deduce the mechanism of the beta sheet formation during the folding of FSD-EY. The last column gives the average distance and its standard deviation between the pairs of atoms that was computed from largest cluster at 292.36 K, see Section 6.4.***

| Contact | Atom 1 | Residue 1 | Atom 2 | Residue 2 | distance [Å] |
|---|---|---|---|---|---|
| **1** | CB | $\text{Arg}_{10}$ | CZ | $\text{Phe}_{21}$ | $6.6 \pm 1.4$ |
| **2** | CZ | $\text{Phe}_{12}$ | CG | $\text{Phe}_{21}$ | $4.0 \pm 0.9$ |
| **3** | CB | $\text{Ala}_{5}$ | CG | $\text{Leu}_{18}$ | $5.1 \pm 0.6$ |

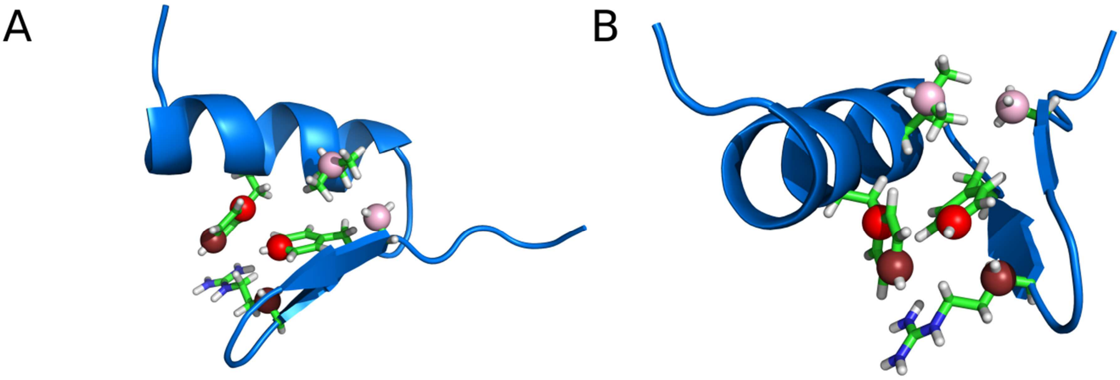


***Figure 6.8. Visualization of the contacts in Table 6.1 used to investigate the formation of the beta sheet during the folding of FSD-EY. The contacts are between the CB atom of residue $\text{Arg}_{10}$ and the CZ atom of $\text{Phe}_{21}$ (dark red spheres), the CZ atom of $\text{Phe}_{12}$ and the CG atom of $\text{Phe}_{21}$ (red spheres), and the CB of $\text{Ala}_{5}$ and CG of $\text{Leu}_{18}$ (bright red spheres).***

Let us consider scenario one. If the beta sheet is formed but not attached to the helix, the distances of contacts 1 and 2 listed in Table 6.1 should show a large variation. I have measured the distance of these two contacts for all conformations in which the beta sheet either is at least partly established or is not present at all according to STRIDE.[47] I used the ensemble at 349.62 K close to the critical folding temperature. Figure 6.9 shows the corresponding two-dimensional free energy landscape projected onto the two contact distances according to Equation (6.2). I observe in Figure 6.9A that for conformations with the native beta sheet, distances with low free energy and therefore high probability are in agreement with the average distances and their standard deviations that are present in the simulated folded ensemble, see Table 6.1. The ensemble without the beta sheet shows a much wider distribution of contact distances with low free energy. Even distances that are larger than 15 Å have free energies below 1.5 kcal/mol. These data do not match the expectation of scenario one, where both contact distances should show a wide distribution even if the beta sheet is formed. Instead, upon formation of the beta sheet, the widths of the contact distance distributions reduce significantly. These data are, therefore, in conflict with scenario one of the beta sheet formation.

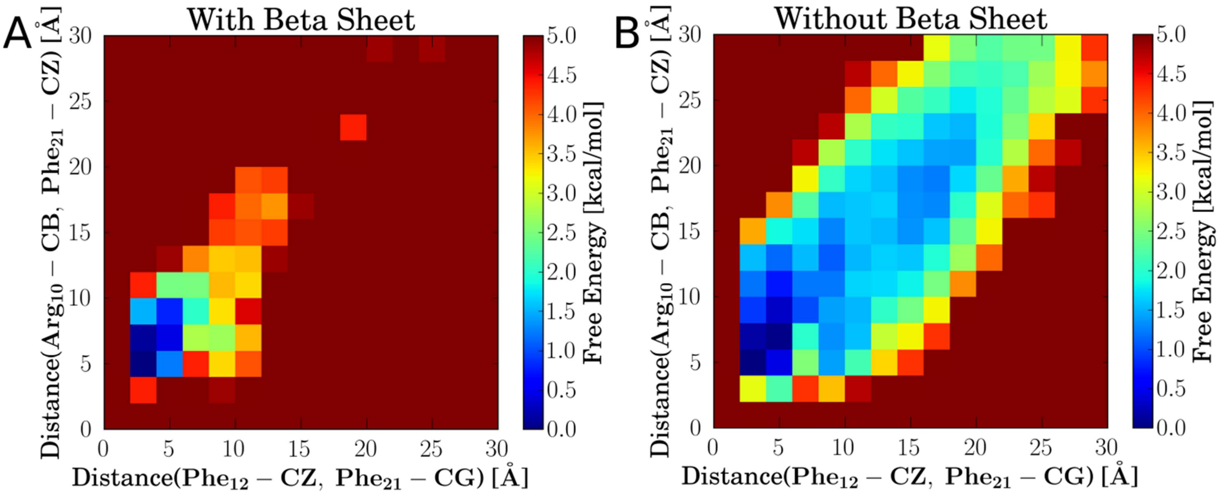


***Figure 6.9. Two-dimensional free energy landscape projected onto the distances between atom pairs of contacts 1 and 2 in Table 6.1 either for the conformations where the beta sheet of FSD-EY is at least partly established (panel A) or for the conformations with no beta sheet (panel B). The data are taken from the replica close to the critical folding temperature at 349.62 K.***

This leaves scenarios two and three for the formation of the beta sheet during the folding of FSD-EY. The second scenario would result in a narrow distance distribution for contact 3 in Table 6.1 peaked at low distances. Equivalently, the projection of the free energy landscape onto this distance should show a free energy minimum at low distances when the beta sheet is not yet established. Since an established contact 3 restraints the positions of those residues between that of the contact, one would expect a distance distribution of moderate width for contacts 1 and 2 in that case.

The third scenario for the formation of the beta sheet should result in a free energy minimum at low distances for the contacts 1 and 2. Because the residue Ala$_5$ is not located between the residues of contacts 1 and 2, the formation of these contacts does not restrain the distance between the atoms of contact 3. Thus, if the beta sheet is not formed, contact 3 should show wide distance distribution at the same time the other two contacts show low distances in scenario three.

I have computed the corresponding distance distributions from the ensemble at 349.62 K excluding all conformations that have at least part of the native beta sheet established. Figure 6.10 shows the corresponding free energy landscapes for the projections onto contacts 1 and 3 as well as onto contacts 2 and 3. Both free energy landscapes show a similar distribution. The free energy minimum is located at low distances of contacts 1 or 2 and high distances of contact 3. Combinations of large distances of contacts 1 and 3, as well as 2 and 3, have also a low free energy. In both graphs, distances of contact 3 smaller than 8 Å have free energies larger than 1.5 kcal/mol. The combination of a small distance for contact 3 with medium or large distance of either contact 1 or 2 have even higher free energies.

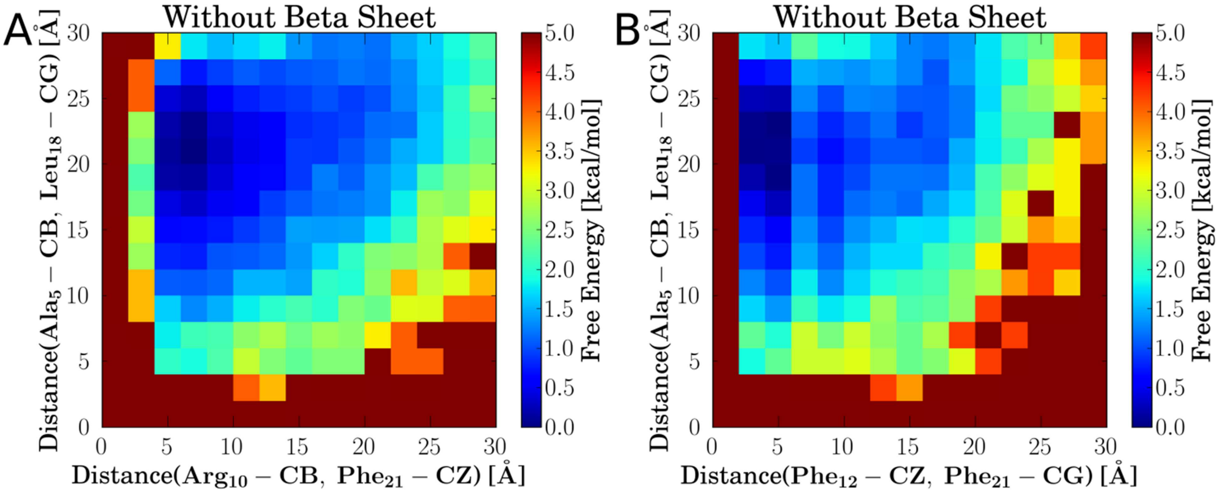


***Figure 6.10. Two-dimensional free energy landscape projected onto the distances between atom pairs of contacts 1 and 3 (panel A) and contacts 2 and 3 (panel B) in Table 6.1 for conformations where no beta sheet is present. The data are taken from the replica close to the critical folding temperature at 349.62 K.***

These data clearly support the third scenario for the formation of the beta sheet. The free energy minima at small distances of contacts 1 and 2 in combination with a large distance of contact three match the expectations of that scenario. The region of the second beta strand is likely to be attached to the alpha helix already, while the beta sheet is not yet formed. This especially includes the packing of the two very hydrophobic Phe$_{12}$ and Phe$_{21}$ against each other. This packing shields both residues from water making this conformation very favorable. Thus, part of the hydrophobic core is already established before the beta sheet forms. In the final step of folding mechanism of FSD-EY, the region of the first beta strand attaches to the alpha helix and second beta strand region to form the folded conformation. This completes the picture of the FSD-EY folding process.

In conclusion, I have carried out parallel tempering Monte Carlo simulations of the FSD-EY protein. The simulations stabilize a folded state at low temperatures. This folded state shows good agreement to the experimentally determined NMR conformation (PDB code 1FME[42]). I observe changes in the packing of specific side chains and the flexible terminal regions of the protein backbone. In addition, the simulations showed a metastable state at very low temperatures, which is characterized by the formation of a 3-10 Helix instead of the beta sheet. Molecular dynamics simulations of FSD-EY in explicit water also found the existence of a metastable state.[33] These results indicate that the implicit solvent model of Chapter 3 correctly balances the propensities of the different secondary structure elements. The simulation replicas at higher temperatures showed a critical folding temperature of 349.62 K with a phase transition between folded and unfolded state. A closer examination of simulation data at this temperature suggested that FSD-EY folds through three steps. In the first step, the alpha helix of FSD-EY folds into its native conformation. In the second step, the region of the second beta strand attaches to the alpha helix, forming part of the hydrophobic core. In the third step, the region of the first beta strand attaches to the other strand and the alpha helix to form the folded state.

# 7 Summary and Outlook

Proteins are an important class of biomolecules, because they take part in fulfilling or regulating nearly all tasks necessary for a cell to function and survive.[1–3] Intrigued by the vast functionality that they provide, scientists have tried to unravel their molecular structure and the atomistic mechanisms that enable this functionality.[1,12,15–18] Besides experimental techniques, biomolecular scientists widely use computational methods for the investigation of proteins and their functions.

A very common approach are molecular dynamics simulations,[19,20] which provide time-resolved trajectories of the underlying molecular mechanisms.[21] Unfortunately, the timescales of many biologically interesting processes are much longer than those that can be reached by these simulations.[21,24–26] Although new algorithms and improved hardware could alleviate this problem to some extent, they are not able to fulfill the demand of biomolecular researchers.[29–34] Thus, how to solve the timescale problem is an open question.

One solution could be to study protein functions by representative ensembles, instead of time resolved trajectories. These ensembles can be generated using Monte Carlo algorithms. However, this strategy is very uncommon in computational biomolecular research. One of the reasons is the lack of an adequate simulation package that can use common molecular forcefields.[36] Another reason is the need to include the physiological environment into the simulation implicitly, because an explicit representation would dramatically reduce the efficiency and success of Monte Carlo algorithms.[36]

The goal of this thesis was to address these problems by developing, implementing, improving and validating the necessary methods, especially an implicit solvent model, for Monte Carlo simulations of proteins with common biomolecular forcefields, as well as demonstrating their success in an exemplary application.

In chapter two, I introduced the basic concepts and theories to understand the results presented in this thesis. These included the composition, properties, and structure of proteins as well as biological membranes. I summarized the potential energy terms of common biomolecular forcefields that were used to model the interactions within biomolecules and reviewed the basic theory of implicit solvent models. The second chapter closed with a brief explanation of Monte Carlo algorithms that I used in this thesis.

In the third chapter, I explained how I implemented the AMBER99SB*-ILDN biomolecular forcefield into the SIMONA[37] Monte Carlo simulation framework. This implementation addressed specific requirements of Monte Carlo algorithms that are not present in molecular dynamics. I showed that the implemented potential terms yielded energies in good numerical agreement with implementations in other molecular simulation packages and performed the computations up to 40 times faster than with previous similar forcefield terms in SIMONA.

To account for the physiological environment of proteins, I implemented in SIMONA a continuum implicit solvent model based on the generalized Born model. Such a model provides solvation free energies that approximate the average interaction of the solvent with the protein. I developed a new algorithm to compute accurate Born radii in the generalized Born model efficiently. This algorithm yielded electrostatic solvation free energies in very good agreement with reference Poisson-Boltzmann calculations with a relative root mean square error of less than 1%, and is therefore one of the most accurate methods available. Computationally, it performed up to an order of magnitude better than similar accurate methods. I published this method together with Wolfgang Wenzel in the Journal of Chemical Theory and Computation.[173] Future improvements should first introduce the correct temperature dependence of the solvation free energy into this model. Currently, it is parameterized at a temperature of approximately 300 K.

With these methods, SIMONA now provides all forcefield terms and an implicit solvent model to carry out Monte Carlo simulations of proteins with the common biomolecular AMBER99SB*-ILDN forcefield. To judge the required resources and feasibility of such Monte Carlo studies of a given protein, I provided an overview of the current simulation performance of SIMONA with these methods. Finally, I showed that the parallelization of all these methods carried out by me allows generating representative ensembles up to 21 times larger in the same time by using up to 32 CPU cores instead of just one.

The fourth chapter focused on improving the accuracy of the approximate description of solvation effects provided by continuum implicit solvent models. I carried out an assessment of three different models together with Julia Setzler and Wolfgang Wenzel.[179] We investigated how accurate these models can estimate experimental hydration free energies for a large database of small chemical compounds. I created an optimized set of freely adjustable model parameters that allowed a fair comparison of these models unbiased by their parameterization. The best model obtained a root mean square error of 1.0 kcal/mol compared to experimental data, while using only ten different atom types with a total of 21 freely adjustable model parameters. We found that this model performed much better than its two competitors do, because it is able to account for the asymmetric behavior of water around oppositely charged ions without explicit parameterization of this effect. Accounting for it in the parameterization of the models significantly improved the accuracy of the other two models, while the best model improved only marginally. These data have highlighted the importance of accounting for this effect in implicit solvent models. In addition, the comparison to other generalized Born based implicit solvent models showed that the combined optimization of all free model parameters together is likely to improve their accuracy further. Our data also indicated that implicit solvent models could yield hydration free energies with better accuracy as explicit solvent models such as TIP3P.

I investigated how the errors of the hydration free energies of these models were distributed among the different chemical groups present in the database. This investigation has also highlighted the importance of accounting for the asymmetry of water. The two models that did not account for this effect showed very large errors for the nitro group, whose nitrogen atoms carry positive partial charges instead of negative partial charges that are present in all other chemical groups containing nitrogen atoms. On the other hand, adding a nitrogen atom type to the model that already accounted for this effect resulted in large errors for sparsely populated chemical groups, because the added atom type destroyed the balance between the different charged nitrogen atoms. I further found that atom-type-dependent parameters for the nonpolar term are sufficient to yield reasonable accurate hydration free energies for compounds containing hypervalent sulfurs. Previous studies had argued that changes of the Lennard-Jones parameters in the general AMBER forcefield would be necessary to achieve this goal.[183,204]

In summary, these results provide a solid basis for the future improvements of continuum implicit solvent models. In the next step, investigations how well these models perform for larger molecules such as proteins are necessary. Investigating how proteins interact with small chemical compounds is of high relevance to pharmaceutical research. Thus, an appropriate implicit solvent model should model solvation effects of small compounds and large biomolecules accurately. However, proteins can undergo large conformational changes, while small molecules cannot. Therefore, the improved modeling of solvation effects for the same protein in different conformations should play a central role in these investigations.

The fifth chapter focused on implicitly modeling biological membranes because they represent another important physiological environment of proteins. I introduced my idea how to decompose an environment consisting of multiple dielectric regions into simpler environments. Each of these can then be treated with an extension of the generalized Born implicit solvent model of Chapter 3. Based on this extension, Julia Setzler, Carolin Seith and I developed the SIMONA layered implicit membrane (SLIM) model. It accounts for the low permittivity inside the membrane due to the presence of the amphipathic phospholipids. We showed that, in contrast to previous models, SLIM captures all qualitative features that are present in Poisson-Boltzmann reference calculations with good quantitative agreement. Thus, SLIM is an important step towards a realistic implicit membrane model. In combination with an already existing nonpolar solvation model that accounts for the absence of the hydrophobic effect inside the membrane, we could study properties of small membrane peptides and proteins with SIMONA Monte Carlo simulations. We found that this model reproduced established properties of these proteins with reasonable agreement and low computational cost. Finally, we have prepared a publication of the SLIM model and the results together with Wolfgang Wenzel.[210] Future efforts to improve the implicit modeling of membranes should focus on accounting for the permanent dipole moments present in the phospholipids as pointed out by Orsi et al.[255] Charged phospholipid headgroups may be taken into account by combining the Gouy-Chapman model (see Mclaughlin[256] for a review) with a generalized Born model that can account for aqueous solutions.[257]

In Chapter 6, I demonstrated the validity and success of my methods for the investigation of proteins with Monte Carlo simulations by studying the folding of the small protein FSD-EY. The native conformation of this protein contains a beta sheet and an alpha helix. I carried out a parallel tempering Monte Carlo simulation of FSD-EY using SIMONA with the forcefield and implicit solvent model of Chapter 3. These methods allowed the simulation of the folding of FSD-EY in only three weeks on conventional supercomputer hardware. In contrast, molecular dynamics required a rare custom-built supercomputer to achieve the folding of this protein.

I found that the simulation successfully stabilized a folded conformation at low to intermediate temperatures. This folded conformation agreed well with that determined by nuclear magnetic resonance spectroscopy. The C-alpha atom root mean square deviation was 2.7 Å. Differences in the conformations were present at the C-terminal end, the loop linking the two beta strands of the beta sheet, as well as in some specific side chains. The alpha helix and the beta sheet of FSD-EY were both one residue longer in the folded conformation of the simulation.

Using the C-alpha atom root mean square deviation and the fraction of established native secondary structure as a reaction coordinate, as well as the fraction of established native secondary structure, I was able to determine the critical temperature of the folding process of FSD-EY from my simulation data. At approximately 350 K, the free energy difference between the minima of the folded and unfolded conformations vanished. Explicit molecular dynamics simulations on a special-purpose supercomputer with a different molecular forcefield resulted in a critical temperature below 325 K.[33] Further investigations will be necessary to determine if this difference is due to the molecular forcefield or the employed implicit solvent model. Unfortunately, no corresponding experimental data is available.

Furthermore, I identified a metastable conformation of FSD-EY. This conformation possesses a different secondary structure. The beta sheet is replaced by a tight 3-10 helix that allows the shielding of the protein's hydrophobic core by the side chains of residues $Tyr_3$ and $Tyr_7$. Although this metastable conformation is energetically favorable, it vanishes at higher temperatures, because it possesses lower entropy than the native beta sheet conformation. The explicit water molecular dynamics simulation of Lindorff-Larsen et al. also showed a metastable state in agreement with my results.[33]

Finally, I studied the mechanism by which FSD-EY folds. My simulation data suggested that the first folding step be the formation of the alpha helix. In the second step, the region of the second beta strand attaches to the alpha helix to form part of the protein's hydrophobic core. Finally, the region of the first beta strand attaches to the second strand and the alpha helix to form the native conformation.

With this simulation, I successfully demonstrated the investigation of the folding of a small protein by using Monte Carlo simulations with the methods developed and implemented by me into the SIMONA Monte Carlo simulation framework. The simulation stabilized a folded state in good agreement with experimental data and identified a metastable conformation in agreement with explicit solvent simulations. Due to the mixed secondary structure elements present in these conformations, these simulation results indicate that the implicit solvent model correctly balances their propensities. Consequently, SIMONA Monte Carlo simulations will allow such protein folding studies on a routine basis in the future. More computational resources with further optimizations of the Monte Carlo and parallel tempering protocol will enable the investigation of larger and biologically more relevant proteins in the future. Thus, the work I presented here will provide a useful toolkit for biomolecular scientists.

# A Appendix

## A.1 Floating Point Numbers

Floating point numbers and operations on them are defined in the IEEE 754 standard.[258] The standard defines different formats of floating point numbers. The most commonly used ones are the so-called single and double precision floating point numbers. A real number $f$ is converted to its binary representation $f_2$ and then represented as

$$f_2 = (-1)^s \cdot m \cdot 2^e. \tag{A.1}$$

Here $s$ determines the sign of $f$, $m$ is the mantissa, whose leading digit is defined to be non-zero, and $e$ is the exponent of the floating point number. In each floating point format, the sign $s$ is represented by a single bit. For single precision floating point numbers, the mantissa has 23 bits and the exponent 8 bits. Thus, a single precision floating point number is 32 bits in size. A double precision floating point number has 52 bits for the mantissa and 11 bits for the exponent, wherefore the size of a double precision number is 64 bits.

The finite number of bits in the mantissa causes that two real numbers are represented by the same floating point number if their difference is small enough. Thus, floating point numbers have a limited precision. For real numbers close to one, this precision is about 7 decimal digits for single precision floating number  and is 16 decimal digits for double precision floating point numbers. The bit size of the exponent determines the range of real numbers that can be represented by floating point numbers.

## A.2 CPU Vector Instructions

CPU vector instructions are specific instructions that can be applied to multiple data items. This scheme is known as single instruction multiple data (SIMD).[259] Many different instruction sets provide vector instructions. Which of these are available depends on the employed hardware and compilers. Some common examples are SSE, AVX, Altivec, and NEON. Information about these instruction sets is available in the manuals and software developer manuals for the CPUs that support them. The sets also differ in the operations they offer and in the number of data items on which a single instruction can be performed. However, this number is usually of power of two.

There are different methods to use these instruction sets. The most convenient method is to let the compiler recognize suitable operations and generate the corresponding vector instructions for the targeted CPU. This process is called auto-vectorization. However, there are strict code requirements for the recognition, wherefore it often fails, and no vector instructions are generated. The compiler manuals explain what code can be vectorized under what conditions, and how to give hints to the compiler.

Another way to use vector instructions is to include them into the code manually by using vector intrinsic functions. These intrinsic functions are translated directly to vector instructions by the compiler. However, each new set of vector instructions requires adaption of the code to the new intrinsic functions. The resulting code is also harder to read and to maintain.

## *A.3 Speedup Measurements*

All speedup measurements were performed on the HERMIT cluster at the HLRS Stuttgart. The HERMIT cluster is a Cray XE6 supercomputer A compute node of this cluster contains a dual socket mainboard equipped with AMD Opteron(tm) 6276 processors. Thus, one node provides up to 32 threads. To ensure that unused resources do not influence the computation time measurements if less than 32 threads are used, I have started $n_\mathrm{p}$ processes of SIMONA using $n_\mathrm{t}$ threads each, so that

$$n_\mathrm{p} \cdot n_\mathrm{t} = 32 \tag{A.2}$$

The timing measurements were always taken from the first SIMONA process. The computation time was measured with the OpenMP[151] omp_get_wtime function, which is part of the SIMONA timers.

The performance comparison between SIMONA with the SLIM model and CHARMM[65] with the HDGB model of Tanizaki and Feig[209] was run on a single node of the BWunicluster at the Steinbuch Centre for Computing with one Intel Xeon E5-2670 processor. Only one thread was used in both cases. The remaining cores of the compute node were empty. GCC compiler suite version 4.8.2 was used to compile SIMONA and CHARMM with architecture specific optimizations and instruction sets enabled in both cases. The CHARMM input was prepared with CHARMM-GUI,[260] whose default settings for the HDGB/GBMV implicit membrane model of Tanizaki and Feig[209] were kept for the simulation. These settings use a rather coarse radial grid for the integration of the Born radii.[260] SIMONA with the SLIM model used the same input as for the performance measurements in Figure 5.6.

## *A.4 Numerical Solvent Accessible Surface Area Computation*

To ensure that the computed solvent accessible surface areas (SASA) based on my parallel power diagram are correct, I have implemented a robust numerical scheme to compute SASA too. This scheme is based on a numerical integration in spherical coordinates $(r, \theta, \phi)$ to determine the SASA $A_i$ of an atom

$$A_i = (r_i + p_\mathrm{r})^2 \int_\mathrm{uncovered} \sin(\theta)\, d\theta d\phi. \tag{A.3}$$

Here, $r_i$ is the radius of atom $i$ and $p_\mathrm{r}$ is the probe radius. The integration region is the part of the sphere that is not inside any other spheres. The implementation approximates the integral by a finite sum over $N_\mathrm{sphere}^2$ points on the surface of the sphere given by

$$\boldsymbol{r}_\mathrm{sphere}(i, k, l) = (r_i + p_\mathrm{r}, \theta_k, \phi_\mathrm{l}), \tag{A.4}$$

$$\theta_k = \frac{\pi}{\mathrm{N}_\mathrm{sphere}}(0.5 + k) \tag{A.5}$$

$$\phi_l = \frac{2\pi}{N_\mathrm{sphere}}(0.5(l \bmod 2) + l), \tag{A.6}$$

The approximate surface are is

$$A_i \approx (r_i + p_\mathrm{r})^2 \sum_{k,l=1}^{N_\mathrm{sphere}} \sin(\theta_k)\, \Delta\theta\Delta\phi\, \mathrm{c}(\theta_k, \phi_l). \tag{A.7}$$

The function $\mathrm{c}(\theta_l, \phi_l)$ is zero if the point $\boldsymbol{r}_\mathrm{sphere}(i, k, l)$ lies inside any other neighboring sphere, otherwise the function is one. The implementation generates these points on the unit sphere, scales them according by $r_i + p_\mathrm{r}$ and then translates them by the position $\boldsymbol{x}_i$ of the atom in question. It finds those points that do not lie inside any other spheres and sums the weight of these points according to Equation (A.7). In the last step, the sum is multiplied by the square of the sum of the atomic radius and probe radius. I used $N_\mathrm{sphere} = 200$ for the comparison of the SASA computation methods in Section 3.2.

# B References

(1) Hong, M.; Zhang, Y.; Hu, F. Membrane Protein Structure and Dynamics from NMR Spectroscopy. *Annu. Rev. Phys. Chem.* **2012**, *63*, 1–24.

(2) Lee, D.; Redfern, O.; Orengo, C. Predicting Protein Function from Sequence and Structure. *Nat. Rev. Mol. Cell Biol.* **2007**, *8*, 995–1005.

(3) Skolnick, J.; Fetrow, J. S. From Genes to Protein Structure and Function: Novel Applications of Computational Approaches in the Genomic Era. *Trends Biotechnol.* **2000**, *18*, 34–39.

(4) Darwin, C.; Wallace, A. On the Tendency of Species to Form Varieties; and on the Perpetuation of Varieties and Species by Natural Means of Selection. *J. Proc. Linn. Soc. Lond. Zool.* **1858**, *3*, 45–62.

(5) Watson, J. D.; Crick, F. H. C. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature* **1953**, *171*, 737–738.

(6) Wilkins, M. H. F.; Stokes, A. R.; Wilson, H. R. Molecular Structure of Nucleic Acids: Molecular Structure of Deoxypentose Nucleic Acids. *Nature* **1953**, *171*, 738–740.

(7) Franklin, R. E.; Gosling, R. G. Molecular Configuration in Sodium Thymonucleate. *Nature* **1953**, *171*, 740–741.

(8) Crack the Code - How the Code was Cracked http://www.nobelprize.org/educational/medicine/gene-code/history.html (accessed Dec 17, 2013).

(9) Matthaei, J. H.; Nirenberg, M. W. CHARACTERISTICS AND STABILIZATION OF DNAASE-SENSITIVE PROTEIN SYNTHESIS IN E. COLI EXTRACTS. *Proc. Natl. Acad. Sci. U. S. A.* **1961**, *47*, 1580–1588.

(10) Matthaei, J. H.; Jones, O. W.; Martin, R. G.; Nirenberg, M. W. CHARACTERISTICS AND COMPOSITION OF RNA CODING UNITS*. *Proc. Natl. Acad. Sci. U. S. A.* **1962**, *48*, 666–677.

(11) Al-Shahib, A.; Breitling, R.; Gilbert, D. R. Predicting Protein Function by Machine Learning on Amino Acid Sequences – a Critical Evaluation. *BMC Genomics* **2007**, *8*, 78.

(12) McLachlan, A. D. Protein Structure and Function. *Annu. Rev. Phys. Chem.* **1972**, *23*, 165–192.

(13) Bernstein, F. C.; Koetzle, T. F.; Williams, G. J.; Meyer, E. F., Jr; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T.; Tasumi, M. The Protein Data Bank: A Computer-Based Archival File for Macromolecular Structures. *J. Mol. Biol.* **1977**, *112*, 535–542.

(14) RCSB Protein Data Bank - RCSB PDB http://www.rcsb.org/pdb/home/home.do (accessed Dec 15, 2013).

(15) Ishima, R.; Torchia, D. A. Protein Dynamics from NMR. *Nat. Struct. Mol. Biol.* **2000**, *7*, 740–743.

(16) Mittermaier, A.; Kay, L. E. New Tools Provide New Insights in NMR Studies of Protein Dynamics. *Science* **2006**, *312*, 224–228.

(17) Markwick, P. R. L.; Malliavin, T.; Nilges, M. Structural Biology by NMR: Structure, Dynamics, and Interactions. *PLoS Comput Biol* **2008**, *4*, e1000168.

(18) Karplus, M.; Kuriyan, J. Molecular Dynamics and Protein Function. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 6679–6685.

(19) Alder, B. J.; Wainwright, T. E. Studies in Molecular Dynamics. I. General Method. *J. Chem. Phys.* **1959**, *31*, 459–466.

(20) Rahman, A. Correlations in the Motion of Atoms in Liquid Argon. *Phys. Rev.* **1964**, *136*, A405–A411.

(21) Klepeis, J. L.; Lindorff-Larsen, K.; Dror, R. O.; Shaw, D. E. Long-Timescale Molecular Dynamics Simulations of Protein Structure and Function. *Curr. Opin. Struct. Biol.* **2009**, *19*, 120–127.

(22) The Nobel Prize in Chemistry 2013 http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/ (accessed Mar 31, 2014).

(23) Warshel, A.; Levitt, M. Theoretical Studies of Enzymic Reactions: Dielectric, Electrostatic and Steric Stabilization of the Carbonium Ion in the Reaction of Lysozyme. *J. Mol. Biol.* **1976**, *103*, 227–249.

(24) Freddolino, P. L.; Schulten, K. Common Structural Transitions in Explicit-Solvent Simulations of Villin Headpiece Folding. *Biophys. J.* **2009**, *97*, 2338–2347.

(25) Pronk, S.; Páll, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; Spoel, D. van der; et al. GROMACS 4.5: A High-Throughput and Highly Parallel Open Source Molecular Simulation Toolkit. *Bioinformatics* **2013**, *29*, 845–854.

(26) Scheraga, H. A.; Khalili, M.; Liwo, A. Protein-Folding Dynamics: Overview of Molecular Simulation Techniques. *Annu. Rev. Phys. Chem.* **2007**, *58*, 57–83.

(27) Feig, M.; Brooks, C. L. Recent Advances in the Development and Application of Implicit Solvent Models in Biomolecule Simulations. *Curr. Opin. Struct. Biol.* **2004**, *14*, 217–224.

(28) Roux, B.; Simonson, T. Implicit Solvent Models. *Biophys. Chem.* **1999**, *78*, 1–20.

(29) Sagui, C.; Darden, T. A. MOLECULAR DYNAMICS SIMULATIONS OF BIOMOLECULES: Long-Range Electrostatic Effects. *Annu. Rev. Biophys. Biomol. Struct.* **1999**, *28*, 155–179.

(30) Sutmann, G.; Gibbon, P.; Lippert, T. *Fast Methods for Long-Range Interactions in Complex Systems*; Forschungszentrum Jülich, 2011.

(31) Shaw, D. E.; Dror, R. O.; Salmon, J. K.; Grossman, J. P.; Mackenzie, K. M.; Bank, J. A.; Young, C.; Deneroff, M. M.; Batson, B.; Bowers, K. J.; et al. Millisecond-Scale Molecular Dynamics Simulations on Anton. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*; SC ’09; ACM: New York, NY, USA, 2009; pp. 39:1–39:11.

(32) Shaw, D. E.; Maragakis, P.; Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Eastwood, M. P.; Bank, J. A.; Jumper, J. M.; Salmon, J. K.; Shan, Y.; et al. Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science* **2010**, *330*, 341 –346.

(33) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. How Fast-Folding Proteins Fold. *Science* **2011**, *334*, 517–520.

(34) Piana, S.; Lindorff-Larsen, K.; Shaw, D. E. Atomic-Level Description of Ubiquitin Folding. *Proc. Natl. Acad. Sci.* **2013**, 201218321.

(35) Laio, A.; Parrinello, M. Escaping Free-Energy Minima. *Proc. Natl. Acad. Sci.* **2002**, *99*, 12562–12566.

(36) Earl, D. J.; Deem, M. W. Monte Carlo Simulations. In *Molecular Modeling of Proteins*; Kukol, A., Ed.; Methods Molecular Biology™; Humana Press, 2008; pp. 25–36.

(37) Strunk, T.; Wolf, M.; Brieg, M.; Klenin, K.; Biewer, A.; Tristram, F.; Ernst, M.; Kleine, P. J.; Heilmann, N.; Kondov, I.; et al. SIMONA 1.0: An Efficient and Versatile Framework for Stochastic Simulations of Molecular and Nanoscale Systems. *J. Comput. Chem.* **2012**, *33*, 2602–2613.

(38) Wang, J.; Cieplak, P.; Kollman, P. A. How Well Does a Restrained Electrostatic Potential (RESP) Model Perform in Calculating Conformational Energies of Organic and Biological Molecules? *J. Comput. Chem.* **2000**, *21*, 1049–1074.

(39) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of Multiple Amber Force Fields and Development of Improved Protein Backbone Parameters. *Proteins Struct. Funct. Bioinforma.* **2006**, *65*, 712–725.

(40) Best, R. B.; Hummer, G. Optimized Molecular Dynamics Force Fields Applied to the Helix−Coil Transition of Polypeptides. *J. Phys. Chem. B* **2009**, *113*, 9004–9015.

(41) Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E. Improved Side-Chain Torsion Potentials for the Amber ff99SB Protein Force Field. *Proteins* **2010**, *78*, 1950–1958.

(42) Sarisky, C. A.; Mayo, S. L. The Bβα Fold: Explorations in Sequence Space. *J. Mol. Biol.* **2001**, *307*, 1411–1418.

(43) Branden, C.; Tooze, J. *Introduction to Protein Structure*; Garland Pub.: New York, 1999.

(44) File:Amidbildung.svg. *Wikipedia, the free encyclopedia*.

(45) Arunan, E.; Desiraju, G. R.; Klein, R. A.; Sadlej, J.; Scheiner, S.; Alkorta, I.; Clary, D. C.; Crabtree, R. H.; Dannenberg, J. J.; Hobza, P.; et al. Defining the Hydrogen Bond: An Account (IUPAC Technical Report). *Pure Appl. Chem.* **2011**, *83*, 1619–1636.

(46) Kabsch, W.; Sander, C. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. *Biopolymers* **1983**, *22*, 2577–2637.

(47) Frishman, D.; Argos, P. Knowledge-Based Protein Secondary Structure Assignment. *Proteins Struct. Funct. Bioinforma.* **1995**, *23*, 566–579.

(48) Ramachandran, G. N.; Ramakrishnan, C.; Sasisekharan, V. Stereochemistry of Polypeptide Chain Configurations. *J. Mol. Biol.* **1963**, *7*, 95–99.

(49) Vijay-Kumar, S.; Bugg, C. E.; Cook, W. J. Structure of Ubiquitin Refined at 1.8Åresolution. *J. Mol. Biol.* **1987**, *194*, 531–544.

(50) Tanford, C. Contribution of Hydrophobic Interactions to the Stability of the Globular Conformation of Proteins. *J. Am. Chem. Soc.* **1962**, *84*, 4240–4247.

(51) Chen, V. B.; Arendall, W. B., 3rd; Headd, J. J.; Keedy, D. A.; Immormino, R. M.; Kapral, G. J.; Murray, L. W.; Richardson, J. S.; Richardson, D. C. MolProbity: All-Atom Structure Validation for Macromolecular Crystallography. *Acta Crystallogr. D Biol. Crystallogr.* **2010**, *66*, 12–21.

(52) McKnight, C. J.; Matsudaira, P. T.; Kim, P. S. NMR Structure of the 35-Residue Villin Headpiece Subdomain. *Nat. Struct. Biol.* **1997**, *4*, 180–184.

(53) Jäger, M.; Zhang, Y.; Bieschke, J.; Nguyen, H.; Dendle, M.; Bowman, M. E.; Noel, J. P.; Gruebele, M.; Kelly, J. W. Structure–function–folding Relationship in a WW Domain. *Proc. Natl. Acad. Sci.* **2006**, *103*, 10648–10653.

(54) Bishop, W. R.; Bell, R. M. Assembly of Phospholipids into Cellular Membranes: Biosynthesis, Transmembrane Movement and Intracellular Translocation. *Annu. Rev. Cell Biol.* **1988**, *4*, 579–606.

(55) Lindahl, E.; Sansom, M. S. Membrane Proteins: Molecular Dynamics Simulations. *Curr. Opin. Struct. Biol.* **2008**, *18*, 425–431.

(56) Cevc, G. *Phospholipids Handbook*; CRC Press, 1993.

(57) Orsi, M.; Michel, J.; Essex, J. W. Coarse-Grain Modelling of DMPC and DOPC Lipid Bilayers. *J. Phys. Condens. Matter* **2010**, *22*, 155106.

(58) Dowhan, W. Molecular Basis For Membrane Phospholipid Diversity:Why Are There So Many Lipids? *Annu. Rev. Biochem.* **1997**, *66*, 199–232.

(59) Phillips, G. B.; Dodge, J. T. Composition of Phospholipids and of Phospholipid Fatty Acids of Human Plasma. *J. Lipid Res.* **1967**, *8*, 676–681.

(60) File:1-Palmitoyl-2-Oleoylphosphatidylcholine.svg. *Wikipedia, the free encyclopedia*.

(61) Alberts, B.; Johnson, A.; Lewis, J.; Raff, M.; Roberts, K.; Walter, P. The Lipid Bilayer http://www.ncbi.nlm.nih.gov/books/NBK26871/ (accessed Mar 14, 2014).

(62) Hub, J. S.; Winkler, F. K.; Merrick, M.; de Groot, B. L. Potentials of Mean Force and Permeabilities for Carbon Dioxide, Ammonia, and Water Flux across a Rhesus Protein Channel and Lipid Membranes. *J. Am. Chem. Soc.* **2010**, *132*, 13251–13263.

(63) Wennberg, C. L.; van der Spoel, D.; Hub, J. S. Large Influence of Cholesterol on Solute Partitioning into Lipid Membranes. *J. Am. Chem. Soc.* **2012**, *134*, 5351–5361.

(64) Genovese, L.; Neelov, A.; Goedecker, S.; Deutsch, T.; Ghasemi, S. A.; Willand, A.; Caliste, D.; Zilberberg, O.; Rayson, M.; Bergman, A.; et al. Daubechies Wavelets as a Basis Set for Density Functional Pseudopotential Calculations. *J. Chem. Phys.* **2008**, *129*, 014109.

(65) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; et al. CHARMM: The Biomolecular Simulation Program. *J. Comput. Chem.* **2009**, *30*, 1545–1614.

(66) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber Biomolecular Simulation Programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.

(67) Lange, O. F.; van der Spoel, D.; de Groot, B. L. Scrutinizing Molecular Mechanics Force Fields on the Submicrosecond Timescale with NMR Data. *Biophys. J.* **2010**, *99*, 647–655.
(68) Lindorff-Larsen, K.; Maragakis, P.; Piana, S.; Eastwood, M. P.; Dror, R. O.; Shaw, D. E. Systematic Validation of Protein Force Fields against Experimental Data. *PLoS ONE* **2012**, *7*, e32131.
(69) Jones, J. E. On the Determination of Molecular Fields. II. From the Equation of State of a Gas. *R. Soc. Lond. Proc. Ser. A* **1924**, *106*, 463–477.
(70) Momany, F. A. Determination of Partial Atomic Charges from Ab Initio Molecular Electrostatic Potentials. Application to Formamide, Methanol, and Formic Acid. *J. Phys. Chem.* **1978**, *82*, 592–601.
(71) Weiser, J.; Shenkin, P. S.; Still, W. C. Approximate Atomic Surfaces from Linear Combinations of Pairwise Overlaps (LCPO). *J. Comput. Chem.* **1999**, *20*, 217–230.
(72) Swanson, J. M. J.; Mongan, J.; McCammon, J. A. Limitations of Atom-Centered Dielectric Functions in Implicit Solvent Models. *J Phys Chem B* **2005**, *109*, 14769–14772.
(73) Lee, B.; Richards, F. M. The Interpretation of Protein Structures: Estimation of Static Accessibility. *J. Mol. Biol.* **1971**, *55*, 379–400.
(74) Richards, F. M. Areas, Volumes, Packing, and Protein Structure. *Annu. Rev. Biophys. Bioeng.* **1977**, *6*, 151–176.
(75) Connolly, M. Solvent-Accessible Surfaces of Proteins and Nucleic-Acids. *Science* **1983**, *221*, 709–713.
(76) Connolly, M. Analytical Molecular-Surface Calculation. *J. Appl. Crystallogr.* **1983**, *16*, 548–558.
(77) Swanson, J. M. J.; Adcock, S. A.; McCammon, J. A. Optimized Radii for Poisson–Boltzmann Calculations with the AMBER Force Field. *J Chem Theory Comput* **2005**, *1*, 484–493.
(78) Nina, M.; Im, W.; Roux, B. Optimized Atomic Radii for Protein Continuum Electrostatics Solvation Forces. *Biophys. Chem.* **1999**, *78*, 89–96.
(79) Nina, M.; Beglov, D.; Roux, B. Atomic Radii for Continuum Electrostatics Calculations Based on Molecular Dynamics Free Energy Simulations. *J Phys Chem B* **1997**, *101*, 5239–5248.
(80) Schwabl, F.; Brewer, W. D. *Statistical Mechanics*; Springer, 2006.
(81) Ben-Naim, A.; Marcus, Y. Solvation Thermodynamics of Nonionic Solutes. *J. Chem. Phys.* **1984**, *81*, 2016–2027.
(82) Shirts, M. R.; Pitera, J. W.; Swope, W. C.; Pande, V. S. Extremely Precise Free Energy Calculations of Amino Acid Side Chain Analogs: Comparison of Common Molecular Mechanics Force Fields for Proteins. *J. Chem. Phys.* **2003**, *119*, 5740.
(83) Kirkwood, J. G. Statistical Mechanics of Fluid Mixtures. *J. Chem. Phys.* **1935**, *3*, 300–313.
(84) Zwanzig, R. W. High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases. *J. Chem. Phys.* **1954**, *22*, 1420–1426.
(85) Bennett, C. H. Efficient Estimation of Free Energy Differences from Monte Carlo Data. *J. Comput. Phys.* **1976**, *22*, 245–268.
(86) Shirts, M. R.; Chodera, J. D. Statistically Optimal Analysis of Samples from Multiple Equilibrium States. *J. Chem. Phys.* **2008**, *129*, 124105.
(87) Silvestrelli, P. L.; Parrinello, M. Water Molecule Dipole in the Gas and in the Liquid Phase. *Phys. Rev. Lett.* **1999**, *82*, 3308–3311.
(88) Malmberg, C. G.; Maryott, A. A. *Dielectric Constant of Water from 0°C to 100°C*; National Bureau of Standards, 1956.
(89) Wyman, J. The Dielectric Constant of Mixtures of Ethyl Alcohol and Water from -5 to 40°. *J. Am. Chem. Soc.* **1931**, *53*, 3292–3301.
(90) Stillinger, F. H. Water Revisited. *Science* **1980**, *209*, 451–457.
(91) Southall, N. T.; Dill, K. A.; Haymet, A. D. J. A View of the Hydrophobic Effect. *J. Phys. Chem. B* **2002**, *106*, 521–533.
(92) Stillinger, F. H. Structure in Aqueous Solutions of Nonpolar Solutes from the Standpoint of Scaled-Particle Theory. *J. Solut. Chem.* **1973**, *2*, 141–158.

(93) Scatena, L. F.; Brown, M. G.; Richmond, G. L. Water at Hydrophobic Surfaces: Weak Hydrogen Bonding and Strong Orientation Effects. *Science* **2001**, *292*, 908–912.

(94) Chen, J.; Brooks, C. L.; Khandogin, J. Recent Advances in Implicit Solvent-Based Methods for Biomolecular Simulations. *Curr. Opin. Struct. Biol.* **2008**, *18*, 140–148.

(95) Baker, N. A. Improving Implicit Solvent Simulations: A Poisson-Centric View. *Curr. Opin. Struct. Biol.* **2005**, *15*, 137–143.

(96) Baker, N. A.; Bashford, D.; Case, D. A. Implicit Solvent Electrostatics in Biomolecular Simulation. In *New Algorithms for Macromolecular Simulation*; Leimkuhler, B.; Chipot, C.; Elber, R.; Laaksonen, A.; Mark, A.; Schlick, T.; Schütte, C.; Skeel, R.; Barth, T. J.; Griebel, M.; et al., Eds.; Lecture Notes in Computational Science and Engineering; Springer Berlin Heidelberg, 2006; Vol. 49, pp. 263–295.

(97) Bashford, D.; Case, D. A. Generalized Born Models of Macromolecular Solvation Effects. *Annu. Rev. Phys. Chem.* **2000**, *51*, 129–152.

(98) Zhang, L. Y.; Gallicchio, E.; Friesner, R. A.; Levy, R. M. Solvent Models for Protein-Ligand Binding: Comparison of Implicit Solvent Poisson and Surface Generalized Born Models with Explicit Solvent Simulations. *J. Comput. Chem.* **2001**, *22*, 591–607.

(99) Jackson, J. D. *Classical Electrodynamics*; John Wiley & Sons, Inc.: New York, NY, USA, 1962.

(100) Feig, M.; Onufriev, A.; Lee, M. S.; Im, W.; Case, D. A.; Brooks III, C. L. Performance Comparison of Generalized Born and Poisson Methods in the Calculation of Electrostatic Solvation Energies for Protein Structures. *J. Comput. Chem.* **2004**, *25*, 265–284.

(101) Onufriev, A.; Case, D. A.; Bashford, D. Effective Born Radii in the Generalized Born Approximation: The Importance of Being Perfect. *J. Comput. Chem.* **2002**, *23*, 1297–1304.

(102) Mongan, J.; Svrcek-Seiler, W. A.; Onufriev, A. Analysis of Integral Expressions for Effective Born Radii. *J. Chem. Phys.* **2007**, *127*, 185101.

(103) Lee, M. C.; Yang, R.; Duan, Y. Comparison between Generalized-Born and Poisson–Boltzmann Methods in Physics-Based Scoring Functions for Protein Structure Prediction. *J. Mol. Model.* **2005**, *12*, 101–110.

(104) Born, M. Volumen Und Hydratationswärme Der Ionen. *Z. Für Phys.* **1920**, *1*, 45–48.

(105) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. Semianalytical Treatment of Solvation for Molecular Mechanics and Dynamics. *J. Am. Chem. Soc.* **1990**, *112*, 6127–6129.

(106) Lee, M. S.; Salsbury, F. R.; Brooks, C. L. Novel Generalized Born Methods. *J. Chem. Phys.* **2002**, *116*, 10606–10614.

(107) Lee, M. S.; Feig, M.; Salsbury, F. R., Jr; Brooks, C. L., 3rd. New Analytic Approximation to the Standard Molecular Volume Definition and Its Application to Generalized Born Calculations. *J. Comput. Chem.* **2003**, *24*, 1348–1356.

(108) Grycuk, T. Deficiency of the Coulomb-Field Approximation in the Generalized Born Model: An Improved Formula for Born Radii Evaluation. *J. Chem. Phys.* **2003**, *119*, 4817–4826.

(109) Danielewicz-Ferchmin, I.; Ferchmin, A. R. Static Permittivity of Water Revisited: E in the Electric Field above 108 V m−1 and in the Temperature Range 273≤T≤373 K. *Phys. Chem. Chem. Phys.* **2004**, *6*, 1332–1339.

(110) Gong, H.; Hocky, G.; Freed, K. F. Influence of Nonlinear Electrostatics on Transfer Energies between Liquid Phases: Charge Burial Is Far Less Expensive than Born Model. *Proc. Natl. Acad. Sci.* **2008**, *105*, 11146 –11151.

(111) Bardhan, J. P. Nonlocal Continuum Electrostatic Theory Predicts Surprisingly Small Energetic Penalties for Charge Burial in Proteins. *J. Chem. Phys.* **2011**, *135*, 104113.

(112) Hildebrandt, A.; Blossey, R.; Rjasanow, S.; Kohlbacher, O.; Lenhof, H.-P. Electrostatic Potentials of Proteins in Water: A Structured Continuum Approach. *Bioinformatics* **2007**, *23*, e99–e103.

(113) Eisenberg, D.; McLachlan, A. D. Solvation Energy in Protein Folding and Binding. *Nature* **1986**, *319*, 199–203.

(114) Ooi, T.; Oobatake, M.; Némethy, G.; Scheraga, H. A. Accessible Surface Areas as a Measure of the Thermodynamic Parameters of Hydration of Peptides. *Proc. Natl. Acad. Sci.* **1987**, *84*, 3086–3090.

(115) Hermann, R. B. Theory of Hydrophobic Bonding. II. Correlation of Hydrocarbon Solubility in Water with Solvent Cavity Surface Area. *J. Phys. Chem.* **1972**, *76*, 2754–2759.

(116) Reynolds, J. A.; Gilbert, D. B.; Tanford, C. Empirical Correlation Between Hydrophobic Free Energy and Aqueous Cavity Surface Area. *Proc. Natl. Acad. Sci.* **1974**, *71*, 2925–2927.

(117) Chothia, C. Hydrophobic Bonding and Accessible Surface Area in Proteins. *Nature* **1974**, *248*, 338–339.

(118) Pierotti, R. A. A Scaled Particle Theory of Aqueous and Nonaqueous Solutions. *Chem. Rev.* **1976**, *76*, 717–726.

(119) Gilson, M. K.; Honig, B. The Inclusion of Electrostatic Hydration Energies in Molecular Mechanics Calculations. *J. Comput. Aided Mol. Des.* **1991**, *5*, 5–20.

(120) Onufriev, A.; Bashford, D.; Case, D. A. Exploring Protein Native States and Large-Scale Conformational Changes with a Modified Generalized Born Model. *Proteins Struct. Funct. Bioinforma.* **2004**, *55*, 383–394.

(121) Mongan, J.; Simmerling, C.; McCammon, J. A.; Case, D. A.; Onufriev, A. Generalized Born Model with a Simple, Robust Molecular Volume Correction. *J. Chem. Theory Comput.* **2007**, *3*, 156–169.

(122) Im, W.; Lee, M. S.; Brooks III, C. L. Generalized Born Model with a Simple Smoothing Function. *J. Comput. Chem.* **2003**, *24*, 1691–1702.

(123) Anandakrishnan, R.; Daga, M.; Onufriev, A. V. An N Log N Generalized Born Approximation. *J. Chem. Theory Comput.* **2011**, *7*, 544–559.

(124) Haberthür, U.; Caflisch, A. FACTS: Fast Analytical Continuum Treatment of Solvation. *J. Comput. Chem.* **2008**, *29*, 701–715.

(125) Zacharias, M. Continuum Solvent Modeling of Nonpolar Solvation: Improvement by Separating Surface Area Dependent Cavity and Dispersion Contributions. *J. Phys. Chem. A* **2003**, *107*, 3000–3004.

(126) Pitera, J. W.; van Gunsteren, W. F. The Importance of Solute-Solvent van Der Waals Interactions with Interior Atoms of Biopolymers. *J. Am. Chem. Soc.* **2001**, *123*, 3163–3164.

(127) Levy, R. M.; Zhang, L. Y.; Gallicchio, E.; Felts, A. K. On the Nonpolar Hydration Free Energy of Proteins: Surface Area and Continuum Solvent Models for the Solute-Solvent Interaction Energy. *J. Am. Chem. Soc.* **2003**, *125*, 9523–9530.

(128) Floris, F.; Tomasi, J. Evaluation of the Dispersion Contribution to the Solvation Energy. A Simple Computational Model in the Continuum Approximation. *J. Comput. Chem.* **1989**, *10*, 616–627.

(129) Wagoner, J. A.; Baker, N. A. Assessing Implicit Models for Nonpolar Mean Solvation Forces: The Importance of Dispersion and Volume Terms. *Proc. Natl. Acad. Sci.* **2006**, *103*, 8331 –8336.

(130) Weeks, J. D.; Chandler, D.; Andersen, H. C. Perturbation Theory of the Thermodynamic Properties of Simple Liquids. *J. Chem. Phys.* **1971**, *55*, 5422–5423.

(131) Gallicchio, E.; Levy, R. M. AGBNP: An Analytic Implicit Solvent Model Suitable for Molecular Dynamics Simulations and High-Resolution Modeling. *J. Comput. Chem.* **2004**, *25*, 479–499.

(132) Tan, C.; Tan, Y.-H.; Luo, R. Implicit Nonpolar Solvent Models. *J. Phys. Chem. B* **2007**, *111*, 12263–12274.

(133) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092.

(134) Swendsen, R. H.; Wang, J.-S. Replica Monte Carlo Simulation of Spin-Glasses. *Phys. Rev. Lett.* **1986**, *57*, 2607–2609.

(135) Geyer, C. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface, Seattle, Washington, April 21-24, 1991*; Interface Foundation of North America, 1992; p. 156.

(136) Earl, D. J.; Deem, M. W. Parallel Tempering: Theory, Applications, and New Perspectives. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3910–3916.

(137) Hansmann, U. H. E. Parallel Tempering Algorithm for Conformational Studies of Biological Molecules. *Chem. Phys. Lett.* **1997**, *281*, 140–150.

(138) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.* **1995**, *117*, 5179–5197.

(139) Bayly, C. I.; Cieplak, P.; Cornell, W.; Kollman, P. A. A Well-Behaved Electrostatic Potential Based Method Using Charge Restraints for Deriving Atomic Charges: The RESP Model. *J. Phys. Chem.* **1993**, *97*, 10269–10280.

(140) Jorgensen, W. L. Quantum and Statistical Mechanical Studies of Liquids. 10. Transferable Intermolecular Potential Functions for Water, Alcohols, and Ethers. Application to Liquid Water. *J. Am. Chem. Soc.* **1981**, *103*, 335–340.

(141) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of Simple Potential Functions for Simulating Liquid Water. *J. Chem. Phys.* **1983**, *79*, 926.

(142) Gotz, A. W.; Williamson, M. J.; Xu, D.; Poole, D.; Le Grand, S.; Walker, R. C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born. *J. Chem. Theory Comput.* **2012**, *8*, 1542–1555.

(143) Patra, M.; Karttunen, M.; Hyvönen, M. T.; Falck, E.; Lindqvist, P.; Vattulainen, I. Molecular Dynamics Simulations of Lipid Bilayers: Major Artifacts Due to Truncating Electrostatic Interactions. *Biophys. J.* **2003**, *84*, 3636–3645.

(144) Patra, M.; Karttunen, M.; Hyvönen, M. T.; Falck, E.; Vattulainen, I. Lipid Bilayers Driven to a Wrong Lane in Molecular Dynamics Simulations by Subtle Changes in Long-Range Electrostatic Interactions. *J. Phys. Chem. B* **2004**, *108*, 4485–4494.

(145) Norberg, J.; Nilsson, L. On the Truncation of Long-Range Electrostatic Interactions in DNA. *Biophys. J.* **2000**, *79*, 1537–1553.

(146) Saito, M. Molecular Dynamics Simulations of Proteins in Solution: Artifacts Caused by the Cutoff Approximation. *J. Chem. Phys.* **1994**, *101*, 4055–4061.

(147) Tasaki, K.; McDonald, S.; Brady, J. w. Observations Concerning the Treatment of Long-Range Interactions in Molecular Dynamics Simulations. *J. Comput. Chem.* **1993**, *14*, 278–284.

(148) Smith, P. E.; Pettitt, B. M. Ewald Artifacts in Liquid State Molecular Dynamics Simulations. *J. Chem. Phys.* **1996**, *105*, 4289–4293.

(149) Piana, S.; Lindorff-Larsen, K.; Dirks, R. M.; Salmon, J. K.; Dror, R. O.; Shaw, D. E. Evaluating the Effects of Cutoffs and Treatment of Long-Range Electrostatics in Protein Folding Simulations. *PLoS ONE* **2012**, *7*, e39918.

(150) Fog, A. Software optimization resources. C++ and assembly. Windows, Linux, BSD, Mac OS X http://www.agner.org/optimize/ (accessed Mar 19, 2014).

(151) OpenMP Architecture Review Board. OpenMP Application Program Interface, Version 3.1, 2011.

(152) Klenin, K. V.; Tristram, F.; Strunk, T.; Wenzel, W. Derivatives of Molecular Surface Area and Volume: Simple and Exact Analytical Formulas. *J. Comput. Chem.* **2011**, *32*, 2647–2653.

(153) Klenin, K.; Tristram, F.; Strunk, T.; Wenzel, W. Achieving Numerical Stability in Analytical Computation of the Molecular Surface and Volume. In *From Computational Biophysics to Systems Biology (CBSB11) Celebrating Harold Scheragas 90th Birthday*; IAS Series; Schriften des Forschungszentrums Jülich: Juelich, 2012; Vol. 8, pp. 71–74.

(154) Amdahl, G. M. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. In *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*; AFIPS '67 (Spring); ACM: New York, NY, USA, 1967; pp. 483–485.

(155) Onderik, J. Efficient Neighbor Search for Particle-Based Fluids. *J. Appl. Math. Stat. Inform. JAMSI* **2007**, *2*.

(156) Lee, M. S.; Olson, M. A. Evaluation of Poisson Solvation Models Using a Hybrid Explicit/Implicit Solvent Method. *J. Phys. Chem. B* **2005**, *109*, 5223–5236.

(157) Wagoner, J.; Baker, N. A. Solvation Forces on Biomolecular Structures: A Comparison of Explicit Solvent and Poisson–Boltzmann Models. *J. Comput. Chem.* **2004**, *25*, 1623–1629.

(158) Qiu, D.; Shenkin, P. S.; Hollinger, F. P.; Still, W. C. The GB/SA Continuum Model for Solvation. A Fast Analytical Method for the Calculation of Approximate Born Radii. *J. Phys. Chem. A* **1997**, *101*, 3005–3014.

(159) Hawkins, G.; Cramer, C.; Truhlar, D. Pairwise Solute Descreening of Solute Charges from a Dielectric Medium. *Chem. Phys. Lett.* **1995**, *246*, 122–129.

(160) Hawkins, G. D.; Cramer, C. J.; Truhlar, D. G. Parametrized Models of Aqueous Free Energies of Solvation Based on Pairwise Descreening of Solute Atomic Charges from a Dielectric Medium. *J Phys Chem* **1996**, *100*, 19824–19839.

(161) Onufriev, A.; Bashford, D.; Case, D. A. Modification of the Generalized Born Model Suitable for Macromolecules. *J. Phys. Chem. B* **2000**, *104*, 3712–3720.

(162) Schaefer, M.; Karplus, M. A Comprehensive Analytical Treatment of Continuum Electrostatics. *J. Phys. Chem.* **1996**, *100*, 1578–1599.

(163) Marenich, A. V.; Cramer, C. J.; Truhlar, D. G. Universal Solvation Model Based on the Generalized Born Approximation with Asymmetric Descreening. *J. Chem. Theory Comput.* **2009**, *5*, 2447–2464.

(164) Cai, W.; Xu, Z.; Baumketner, A. A New FFT-Based Algorithm to Compute Born Radii in the Generalized Born Theory of Biomolecule Solvation. *J. Comput. Phys.* **2008**, *227*, 10162–10177.

(165) Grant, J. A.; Pickup, B. T.; Sykes, M. J.; Kitchen, C. A.; Nicholls, A. The Gaussian Generalized Born Model: Application to Small Molecules. *Phys. Chem. Chem. Phys.* **2007**, *9*, 4913–4922.

(166) Ghosh, A.; Rapp, C. S.; Friesner, R. A. Generalized Born Model Based on a Surface Integral Formulation. *J. Phys. Chem. B* **1998**, *102*, 10983–10990.

(167) Gallicchio, E.; Zhang, L. Y.; Levy, R. M. The SGB/NP Hydration Free Energy Model Based on the Surface Generalized Born Solvent Reaction Field and Novel Nonpolar Hydration Free Energy Estimators. *J. Comput. Chem.* **2002**, *23*, 517–529.

(168) Gallicchio, E.; Paris, K.; Levy, R. M. The AGBNP2 Implicit Solvation Model. *J. Chem. Theory Comput.* **2009**, *5*, 2544–2564.

(169) Tjong, H.; Zhou, H.-X. GBr6: A Parameterization-Free, Accurate, Analytical Generalized Born Method. *J. Phys. Chem. B* **2007**, *111*, 3055–3061.

(170) Tjong, H.; Zhou, H.-X. GBr[sup 6]NL: A Generalized Born Method for Accurately Reproducing Solvation Energy of the Nonlinear Poisson-Boltzmann Equation. *J. Chem. Phys.* **2007**, *126*, 195102.

(171) Xu, Z.; Cheng, X.; Yang, H. Treecode-Based Generalized Born Method. *J. Chem. Phys.* **2011**, *134*, 064107.

(172) Chocholoušová, J.; Feig, M. Balancing an Accurate Representation of the Molecular Surface in Generalized Born Formalisms with Integrator Stability in Molecular Dynamics Simulations. *J. Comput. Chem.* **2006**, *27*, 719–729.

(173) Brieg, M.; Wenzel, W. PowerBorn: A Barnes–Hut Tree Implementation for Accurate and Efficient Born Radii Computation. *J. Chem. Theory Comput.* **2013**, *9*, 1489–1498.

(174) Barnes, J.; Hut, P. A Hierarchical O(N Log N) Force-Calculation Algorithm. *Nature* **1986**, *324*, 446–449.

(175) Baker, N. A.; Sept, D.; Joseph, S.; Holst, M. J.; McCammon, J. A. Electrostatics of Nanosystems: Application to Microtubules and the Ribosome. *Proc. Natl. Acad. Sci. U. S. A.* **2001**, *98*, 10037 – 10041.

(176) Deacon, A.; Gleichmann, T.; Kalb (Gilboa), A. J.; Price, H.; Raftery, J.; Bradbrook, G.; Yariv, J.; Helliwell, J. R. The Structure of Concanavalin A and Its Bound Solvent Determined with Small-Molecule Accuracy at 0.94 [Aring ]resolution. *J. Chem. Soc. Faraday Trans.* **1997**, *93*, 4305–4312.

(177) Brieg, M.; Wenzel, W. Parallelization of an Efficient Method for Calculating Born Radii. In *From Computational Biophysics to Systems Biology (CBSB11) Celebrating Harold Scheragas 90th Birthday*; IAS Series; Schriften des Forschungszentrums Jülich: Juelich, 2012; Vol. 8, pp. 33–36.

(178) Eigen http://eigen.tuxfamily.org/index.php?title=Main_Page (accessed Feb 5, 2014).

(179) Brieg, M.; Setzler, J.; Wenzel, W. Small Molecule Hydration Free Energies: Closing the Gap between Explicit and Implicit Solvent Models. *Prep.*

(180) Gilson, M. K.; Zhou, H.-X. Calculation of Protein-Ligand Binding Affinities. In *Annual Review of Biophysics and Biomolecular Structure*; 2007; Vol. 36, pp. 21–42.

(181) Jorgensen, W. L. The Many Roles of Computation in Drug Discovery. *Science* **2004**, *303*, 1813–1818.

(182) Mobley, D. L.; Dill, K. A. Binding of Small-Molecule Ligands to Proteins: "What You See" Is Not Always "What You Get."*Structure* **2009**, *17*, 489–498.

(183) Knight, J. L.; Brooks III, C. L. Surveying Implicit Solvent Models for Estimating Small Molecule Absolute Hydration Free Energies. *J. Comput. Chem.* **2011**, *32*, 2909–2923.

(184) Bordner, A. J.; Cavasotto, C. N.; Abagyan, R. A. Accurate Transferable Model for Water, N-Octanol, and N-Hexadecane Solvation Free Energies. *J. Phys. Chem. B* **2002**, *106*, 11009–11015.

(185) Kelly, C. P.; Cramer, C. J.; Truhlar, D. G. SM6: A Density Functional Theory Continuum Solvation Model for Calculating Aqueous Solvation Free Energies of Neutrals, Ions, and Solute−Water Clusters. *J. Chem. Theory Comput.* **2005**, *1*, 1133–1152.

(186) Rizzo, R. C.; Aynechi, T.; Case, D. A.; Kuntz, I. D. Estimation of Absolute Free Energies of Hydration Using Continuum Methods: Accuracy of Partial Charge Models and Optimization of Nonpolar Contributions. *J. Chem. Theory Comput.* **2006**, *2*, 128–139.

(187) Aguilar, B.; Onufriev, A. V. Efficient Computation of the Total Solvation Energy of Small Molecules via the R6 Generalized Born Model. *J. Chem. Theory Comput.* **2012**, *8*, 2404–2411.

(188) Mobley, D. L.; Dill, K. A.; Chodera, J. D. Treating Entropy and Conformational Changes in Implicit Solvent Simulations of Small Molecules. *J Phys Chem B* **2008**, *112*, 938–946.

(189) Mobley, D. L.; Bayly, C. I.; Cooper, M. D.; Shirts, M. R.; Dill, K. A. Small Molecule Hydration Free Energies in Explicit Solvent: An Extensive Test of Fixed-Charge Atomistic Simulations. *J Chem Theory Comput* **2009**, *5*, 350–358.

(190) Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I. Fast, Efficient Generation of High-Quality Atomic Charges. AM1-BCC Model: I. Method. *J. Comput. Chem.* **2000**, *21*, 132–146.

(191) Jakalian, A.; Jack, D. B.; Bayly, C. I. Fast, Efficient Generation of High-Quality Atomic Charges. AM1-BCC Model: II. Parameterization and Validation. *J. Comput. Chem.* **2002**, *23*, 1623–1641.

(192) Kondov, I. Protein Structure Prediction Using Distributed Parallel Particle Swarm Optimization. *Nat. Comput.* **2013**, *12*, 29–41.

(193) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field. *J. Comput. Chem.* **2004**, *25*, 1157–1174.

(194) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A. Automatic Atom Type and Bond Type Perception in Molecular Mechanical Calculations. *J. Mol. Graph. Model.* **2006**, *25*, 247–260.

(195) Latimer, W. M.; Pitzer, K. S.; Slansky, C. M. The Free Energy of Hydration of Gaseous Ions, and the Absolute Potential of the Normal Calomel Electrode. *J. Chem. Phys.* **1939**, *7*, 108–111.

(196) Rashin, A. A.; Honig, B. Reevaluation of the Born Model of Ion Hydration. *J. Phys. Chem.* **1985**, *89*, 5588–5593.

(197) Roux, B.; Yu, H. A.; Karplus, M. Molecular Basis for the Born Model of Ion Solvation. *J. Phys. Chem.* **1990**, *94*, 4683–4688.

(198) Hummer, G.; Pratt, L. R.; García, A. E. Free Energy of Ionic Hydration. *J. Phys. Chem.* **1996**, *100*, 1206–1215.

(199) Lynden-Bell, R. M.; Rasaiah, J. C. From Hydrophobic to Hydrophilic Behaviour: A Simulation Study of Solvation Entropy and Free Energy of Simple Solutes. *J. Chem. Phys.* **1997**, *107*, 1981–1991.

(200) Koneshan, S.; Rasaiah, J. C.; Lynden-Bell, R. M.; Lee, S. H. Solvent Structure, Dynamics, and Ion Mobility in Aqueous Solutions at 25 °C. *J. Phys. Chem. B* **1998**, *102*, 4193–4204.

(201) Ashbaugh, H. S. Convergence of Molecular and Macroscopic Continuum Descriptions of Ion Hydration. *J. Phys. Chem. B* **2000**, *104*, 7235–7238.

(202) Mobley, D. L.; Barber, A. E.; Fennell, C. J.; Dill, K. A. Charge Asymmetries in Hydration of Polar Solutes. *J. Phys. Chem. B* **2008**, *112*, 2405–2414.

(203) File:Nitro-Group-2D.png. *Wikipedia, the free encyclopedia*.

(204) Mobley, D. L.; Bayly, C. I.; Cooper, M. D.; Dill, K. A. Predictions of Hydration Free Energies from All-Atom Molecular Dynamics Simulations. *J. Phys. Chem. B* **2009**, *113*, 4533–4537.

(205) Nymeyer, H.; Zhou, H.-X. A Method to Determine Dielectric Constants in Nonhomogeneous Systems: Application to Biological Membranes. *Biophys. J.* **2008**, *94*, 1185–1193.

(206) Spassov, V. Z.; Yan, L.; Szalma, S. Introducing an Implicit Membrane in Generalized Born/Solvent Accessibility Continuum Solvent Models. *J. Phys. Chem. B* **2002**, *106*, 8726–8738.

(207) Im, W.; Feig, M.; Brooks, C. L. An Implicit Membrane Generalized Born Theory for the Study of Structure, Stability, and Interactions of Membrane Proteins. *Biophys. J.* **2003**, *85*, 2900–2918.

(208) Ulmschneider, M. B.; Ulmschneider, J. P.; Sansom, M. S. P.; Di Nola, A. A Generalized Born Implicit-Membrane Representation Compared to Experimental Insertion Free Energies. *Biophys. J.* **2007**, *92*, 2338–2349.

(209) Tanizaki, S.; Feig, M. A Generalized Born Formalism for Heterogeneous Dielectric Environments: Application to the Implicit Modeling of Biological Membranes. *J. Chem. Phys.* **2005**, *122*, 124706.

(210) Setzler, J.; Seith, C.; Brieg, M.; Wenzel, W. SLIM: An Improved Generalized Born Implicit Membrane Model. *Prep.*

(211) Bardhan, J. P. Interpreting the Coulomb-Field Approximation for Generalized-Born Electrostatics Using Boundary-Integral Equation Theory. *J. Chem. Phys.* **2008**, *129*, 144105.

(212) Jo, S.; Vargyas, M.; Vasko-Szedlar, J.; Roux, B.; Im, W. PBEQ-Solver for Online Visualization of Electrostatic Potential of Biomolecules. *Nucleic Acids Res.* **2008**, *36*, W270–W275.

(213) Im, W.; Beglov, D.; Roux, B. Continuum Solvation Model: Computation of Electrostatic Forces from Numerical Solutions to the Poisson-Boltzmann Equation. *Comput. Phys. Commun.* **1998**, *111*, 59–75.

(214) Gesell, J.; Zasloff, M.; Opella, S. J. Two-Dimensional 1H NMR Experiments Show That the 23-Residue Magainin Antibiotic Peptide Is an Alpha-Helix in Dodecylphosphocholine Micelles, Sodium Dodecylsulfate Micelles, and Trifluoroethanol/water Solution. *J. Biomol. NMR* **1997**, *9*, 127–135.

(215) De Planque, M. R. R.; Greathouse, D. V.; Koeppe, R. E.; Schäfer, H.; Marsh, D.; Killian, J. A. Influence of Lipid/Peptide Hydrophobic Mismatch on the Thickness of Diacylphosphatidylcholine Bilayers. A 2H NMR and ESR Study Using Designed Transmembrane A-Helical Peptides and Gramicidin A†. *Biochemistry (Mosc.)* **1998**, *37*, 9333–9345.

(216) Killian, J. A. Hydrophobic Mismatch between Proteins and Lipids in Membranes. *Biochim. Biophys. Acta BBA - Rev. Biomembr.* **1998**, *1376*, 401–416.

(217) Terwilliger, T. C.; Eisenberg, D. The Structure of Melittin. I. Structure Determination and Partial Refinement. *J. Biol. Chem.* **1982**, *257*, 6010–6015.

(218) Terwilliger, T. C.; Eisenberg, D. The Structure of Melittin. II. Interpretation of the Structure. *J. Biol. Chem.* **1982**, *257*, 6016–6022.

(219) MacKenzie, K. R.; Prestegard, J. H.; Engelman, D. M. A Transmembrane Helix Dimer: Structure and Implications. *Science* **1997**, *276*, 131–133.

(220) Subramaniam, S.; Henderson, R. Molecular Mechanism of Vectorial Proton Translocation by Bacteriorhodopsin. *Nature* **2000**, *406*, 653–657.

(221) Tanizaki, S.; Feig, M. Molecular Dynamics Simulations of Large Integral Membrane Proteins with an Implicit Membrane Model. *J. Phys. Chem. B* **2006**, *110*, 548–556.

(222) Dill, K. A.; Ozkan, S. B.; Shell, M. S.; Weikl, T. R. The Protein Folding Problem. *Annu. Rev. Biophys.* **2008**, *37*, 289–316.

(223) So Much More to Know …. *Science* **2005**, *309*, 78–102.

(224) Kubelka, J.; Hofrichter, J.; Eaton, W. A. The Protein Folding "Speed Limit."*Curr. Opin. Struct. Biol.* **2004**, *14*, 76–88.

(225) Duan, Y.; Kollman, P. A. Pathways to a Protein Folding Intermediate Observed in a 1-Microsecond Simulation in Aqueous Solution. *Science* **1998**, *282*, 740–744.

(226) Sugita, Y.; Okamoto, Y. Replica-Exchange Molecular Dynamics Method for Protein Folding. *Chem. Phys. Lett.* **1999**, *314*, 141–151.

(227) García, A. E.; Onuchic, J. N. Folding a Protein in a Computer: An Atomic Description of the Folding/unfolding of Protein A. *Proc. Natl. Acad. Sci.* **2003**, *100*, 13898–13903.

(228) Zhou, R. H. Trp-Cage: Folding Free Energy Landscape in Explicit Water. *Proc. Natl. Acad. Sci. U. S. A.* **2003**, *100*, 13280–13285.

(229) Pitera, J. W.; Swope, W. Understanding Folding and Design: Replica-Exchange Simulations of "Trp-Cage" Fly Miniproteins. *Proc. Natl. Acad. Sci. U. S. A.* **2003**, *100*, 7587–7592.

(230) Paschek, D.; Hempel, S.; Garcia, A. E. Computing the Stability Diagram Trp-Cage Miniprotein of the. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105*, 17754–17759.

(231) Andrec, M.; Felts, A. K.; Gallicchio, E.; Levy, R. M. Protein Folding Pathways from Replica Exchange Simulations and a Kinetic Network Model. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 6801–6806.

(232) Nguyen, P. H.; Stock, G.; Mittag, E.; Hu, C.-K.; Li, M. S. Free Energy Landscape and Folding Mechanism of a Β-Hairpin in Explicit Water: A Replica Exchange Molecular Dynamics Study. *Proteins Struct. Funct. Bioinforma.* **2005**, *61*, 795–808.

(233) Zhou, R.; Berne, B. J.; Germain, R. The Free Energy Landscape for Β Hairpin Folding in Explicit Water. *Proc. Natl. Acad. Sci.* **2001**, *98*, 14931–14936.

(234) Rao, F.; Caflisch, A. Replica Exchange Molecular Dynamics Simulations of Reversible Folding. *J. Chem. Phys.* **2003**, *119*, 4035–4042.

(235) Felts, A. K.; Harano, Y.; Gallicchio, E.; Levy, R. M. Free Energy Surfaces of Β-Hairpin and Α-Helical Peptides Generated by Replica Exchange Molecular Dynamics with the AGBNP Implicit Solvent Model. *Proteins Struct. Funct. Bioinforma.* **2004**, *56*, 310–321.

(236) Zhang, J.; Qin, M.; Wang, W. Folding Mechanism of Β-Hairpins Studied by Replica Exchange Molecular Simulations. *Proteins Struct. Funct. Bioinforma.* **2006**, *62*, 672–685.

(237) Buchete, N.-V.; Hummer, G. Peptide Folding Kinetics from Replica Exchange Molecular Dynamics. *Phys. Rev. E* **2008**, *77*, 030902.

(238) Periole, X.; Mark, A. E. Convergence and Sampling Efficiency in Replica Exchange Simulations of Peptide Folding in Explicit Solvent.*J. Chem. Phys.* **2007**, *126*, 014903.

(239) Clementi, C. Coarse-Grained Models of Protein Folding: Toy Models or Predictive Tools? *Curr. Opin. Struct. Biol.* **2008**, *18*, 10–15.

(240) Derreumaux, P. Coarse-Grained Models for Protein Folding and Aggregation. In *Biomolecular Simulations*; Monticelli, L.; Salonen, E., Eds.; Methods in Molecular Biology; Humana Press, 2013; pp. 585–600.

(241) Heilmann, N. Simulation Kleiner Proteine Mittels Einer Monte-Carlo Merhode. Diploma Thesis, Karlsruhe Institute of Technology: Karlsruhe, Germany, 2013.

(242) Wolf, M. All-Atom Modeling of Protein Folding and Aggregation. Phd Thesis, Karlsruhe Institute of Technology: Karlsruhe, Germany, 2013.

(243) Roe, D. R.; Okur, A.; Wickstrom, L.; Hornak, V.; Simmerling, C. Secondary Structure Bias in Generalized Born Solvent Models: Comparison of Conformational Ensembles and Free Energy of Solvent Polarization from Explicit and Implicit Solvation.*J. Phys. Chem. B* **2007**, *111*, 1846–1857.

(244) Nguyen, H.; Roe, D. R.; Simmerling, C. Improved Generalized Born Solvent Model Parameters for Protein Simulations.*J. Chem. Theory Comput.* **2013**, *9*, 2020–2034.

(245) Bittner, E.; Nußbaumer, A.; Janke, W. Make Life Simple: Unleash the Full Power of the Parallel Tempering Algorithm. *Phys. Rev. Lett.* **2008**, *101*, 130603.

(246) Kabsch, W. A Solution for the Best Rotation to Relate Two Sets of Vectors. *Acta Crystallogr. Sect. A* **1976**, *32*, 922–923.

(247) Kabsch, W. A Discussion of the Solution for the Best Rotation to Relate Two Sets of Vectors. *Acta Crystallogr. Sect. A* **1978**, *34*, 827–828.

(248) Pande, V. S.; Grosberg AYu; Tanaka, T.; Rokhsar, D. S. Pathways for Protein Folding: Is a New View Needed? *Curr. Opin. Struct. Biol.* **1998**, *8*, 68–79.

(249) Garde, S.; García, A. E.; Pratt, L. R.; Hummer, G. Temperature Dependence of the Solubility of Non-Polar Gases in Water. *Biophys. Chem.* **1999**, *78*, 21–32.

(250) Jorgensen, W.; Madura, J. Temperature and Size Dependence for Monte-Carlo Simulations of TIP4P Water. *Mol. Phys.* **1985**, *56*, 1381–1392.

(251) Ashbaugh, H. S.; Liu, L.; Surampudi, L. N. Optimization of Linear and Branched Alkane Interactions with Water to Simulate Hydrophobic Hydration. *J. Chem. Phys.* **2011**, *135*, 054510.

(252) Taverna, D. M.; Goldstein, R. A. Why Are Proteins Marginally Stable? *Proteins* **2002**, *46*, 105–109.

(253) MacKerell, A. D.; Bashford, D.; Bellott; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; et al. All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins†. *J. Phys. Chem. B* **1998**, *102*, 3586–3616.

(254) Piana, S.; Lindorff-Larsen, K.; Shaw, D. E. How Robust Are Protein Folding Simulations with Respect to Force Field Parameterization? *Biophys. J.* **2011**, *100*, L47–L49.

(255) Orsi, M.; Haubertin, D. Y.; Sanderson, W. E.; Essex, J. W. A Quantitative Coarse-Grain Model for Lipid Bilayers. *J. Phys. Chem. B* **2008**, *112*, 802–815.

(256) McLaughlin, S. The Electrostatic Properties of Membranes. *Annu. Rev. Biophys. Biophys. Chem.* **1989**, *18*, 113–136.

(257) Srinivasan, J.; Trevathan, M. W.; Beroza, P.; Case, D. A. Application of a Pairwise Generalized Born Model to Proteins and Nucleic Acids: Inclusion of Salt Effects. *Theor. Chem. Acc.* **1999**, *101*, 426–434.

(258) IEEE Standard for Floating-Point Arithmetic. *IEEE Std 754-2008* **2008**, 1–70.

(259) Flynn, M. Some Computer Organizations and Their Effectiveness. *IEEE Trans. Comput.* **1972**, *C-21*, 948–960.

(260) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. CHARMM-GUI: A Web-Based Graphical User Interface for CHARMM. *J. Comput. Chem.* **2008**, *29*, 1859–1865.

# C Publications

### *Journals, Proceedings and Book Chapters*

1. Anand, P.; Strunk, T.; Brieg, M.; Meliciani, I.; Wolf, M.; Klenin, K.; Wenzel, W. Performance of An All-Atom Free Energy Approach For Protein Structure Prediction. *Biophysical Journal* **2011**, *100*, 48a.
2. Kondov, I.; Maul, R.; Klenin, K.; Brieg, M.; Poschlad, A.; Bagrets, A.; Meded, V.; Bozic, S. Multiscale Materials and Biomolecular Simulations at Simulation Lab NanoMikro. In; Kondov, I.; Poghosyan, G.; Kirner, O.; Schneider, O.; Schmitz, F., Eds.; KIT Scientific Publishing, Karlsruhe: Karlsruhe, Germany, **2011**; pp. 5–15.
3. Strunk, T.; Anand, P.; Brieg, M.; Wolf, M.; Klenin, K. V.; Meliciani, I.; Tristram, F.; Kondov, I.; Wenzel, W. Benchmarking the POEM@HOME Network for Protein Structure Prediction. In *Proceedings of the 3rd International Workshop on Science Gateways for Life Sciences*; CEUR-WS.org: London, UK, **2011**; Vol. 819.
4. Brieg, M.; Wenzel, W. Parallelization of an Efficient Method for Calculating Born Radii. In *From Computational Biophysics to Systems Biology (CBSB11) Celebrating Harold Scheragas 90th Birthday*; IAS Series; Schriften des Forschungszentrums Jülich: Jülich, 2012; Vol. 8, pp. 33–36.
5. Strunk, T.; Wolf, M.; Brieg, M.; Klenin, K.; Biewer, A.; Tristram, F.; Ernst, M.; Kleine, P. J.; Heilmann, N.; Kondov, I.; Wenzel, W. SIMONA 1.0: An Efficient and Versatile Framework for Stochastic Simulations of Molecular and Nanoscale Systems. *J Comput Chem* **2012**, *33*, 2602–2613.
6. Brieg, M.; Setzler, J.; Wenzel, W. A Reparametrized Implicit Solvent Model for Accurate Computation of Hydration Free Energies. *Biophysical Journal* **2013**, *104*, 507a.
7. Brieg, M.; Wenzel, W. PowerBorn: A Barnes–Hut Tree Implementation for Accurate and Efficient Born Radii Computation. *J. Chem. Theory Comput.* **2013**, *9*, 1489–1498.
8. Heilmann, N.; Setzler, J.; Brieg, M.; Strunk, T.; Wolf, M.; Seith, C.; Wenzel, W. Thermodynamic Characterization of Protein Folding Equilibriums at the All Atom Level. *Biophysical Journal* **2013**, *104*, 369a–370a.
9. Brieg, M.; Setzler, J.; Wenzel, W. Assessment of Nonpolar Terms in Implicit Solvent Models to Estimate Small Molecule Hydration Free Energies. *Biophysical Journal* **2014**, *106*, 408a.
10. Heilmann, N. M.; Wolf, M.; Strunk, T.; Setzler, J.; Brieg, M.; Wenzel, W. Thermodynamic Characterization of Protein Folding Using Monte Carlo Methods. *Biophysical Journal* **2014**, *106*, 260a.
11. Setzler, J.; Seith, C.; Brieg, M.; Wenzel, W. Modeling Membrane Proteins with Slim, a New Implicit Membrane Model. *Biophysical Journal* **2014**, *106*, 89a.
12. Brieg, M.; Setzler, J.; Wenzel, W. Small Molecule Hydration Free Energies: Closing the Gap between Implicit and Explicit Solvent Models. (in preparation)
13. Setzler, J.; Seith, C.; Brieg, M.; Wenzel, W. SLIM: An Improved Generalized Born Implicit Membrane Model. (in preparation)

### *Talks*

14. Brieg, M.; Wenzel W. PowerBornRadii: A fast accurate method for calculating Born Radii. *Workshop on Computer Simulation and Theory of Macromolecules 2011*, Hühfeld, Deutschland **2011**

### *Posters*

15. Brieg, M.; Kondov, I.; Wenzel, W. Proteinmodellierung mit biophysikalischen Kraftfeldern auf modernen Hochleistungsrechnerarchitekturen. *Forschungstag Lebenswissenschaften 2011*, Heidelberg, Deutschland **2011**
16. Brieg, M.; Kondov, I.; Wenzel, W. Protein Modeling on High Performance Computing Architectures Using Biophysical Force Fields. *CFN Summer School 2011*, Bad Herrenalb, Deutschland **2011**
17. Brieg, M.; Kondov, I.; Wenzel, W. Parallelization of an Efficient Method for Calculating Born Radii. *Fast Methods for Long-Range Interactions in Complex Systems, Poster Presentations, Summer School*, Jülich, Deutschland **2011**
18. Brieg, M.; Yavorskyy, B.; Meded V.; Poschlad, A.; Klenin, K.; Kondov I. Research in the SimLab NanoMikro. *SCC Hausmesse 2012*, Karlsurhe, Deutschland **2012**
19. Brieg, M.; Setzler, J.; Heilmann, N.; Seith, C.; Wenzel, W. Studying protein folding with atomistic Monte Carlo simulations in implicit solvent. *Annual Meeting of the German Biophysical Society 2012*, Göttingen, Deutschland **2012**
20. Brieg, M.; Setzler J.; Wenzel, W. A Reparametrized Implicit Solvent Model for Accurate Computation of Hydration Free Energies. *Joint Meeting of the British and German Biophysical Society 2013*, Hünfeld, Deutschland **2013**
21. Setzler, J.; Brieg, M.; Seith, C.; Wenzel W. Modeling Peptide Conformations and Insertion in an Implicit Membrane Model. *Joint Meeting of the British and German Biophysical Society 2013*, Hünfeld, Deutschland **2013**
22. Brieg, M.; Setzler J.; Wenzel, W. A Reparametrized Implicit Solvent Model for Accurate Computation of Hydration Free Energies. *Workshop on Computer Simulation and Theory of Macromolecules 2013*, Hünfeld, Deutschland **2013**
23. Setzler J**.**, Brieg M.; Seith C.; Wenzel W. Modeling Membrane Proteins with a new Implcit Membrane Model. *Workshop on Computer Simulation and Theory of Macromolecules 2013*, Hünfeld, Deutschland **2013**
24. Heilmann, N.; Setzler, J.; Wolf, M.; Strunk, T.; Brieg, M.; Wenzel, W. Thermodynamic Characterization of Protein Folding Using Monte Carlo Methods. *Workshop on Computer Simulation and Theory of Macromolecules 2013*, Hünfeld, Deutschland **2013**
25. Brieg, M.; Setzler J.; Wenzel, W. Assessment of Nonpolar Terms in Implicit Solvent Models to Estimate Small Molecule Hydration Free Energies. *Workshop on Computer Simulation and Theory of Macromolecules 2014*, Hünfeld, Deutschland **2014**

# *D List of Additionally Employed Software*

Figures:

- GIMP
- Inkscape
- XmGrace
- Python’s Matplotlib
- PyMol
- Microsoft Word
- 2D Sketcher of ChemDoodle

Word processing:

- Microsoft Word 2010
- Zotero
- Grammarly

# E Acknowledgements

First, I would like to thank Prof. Dr. Wolfgang Wenzel for giving me the opportunity to do my doctoral graduation in the research field of computational biophysics after having spent so much time on theoretical particle and high-energy physics. In addition, I would like to thank him for the guidance he provided during my graduation, but also for giving me the time to follow my own research ideas.

Second, I would like to thank Prof. Dr. Gerd Schön for being the second reviewer of my thesis.

Many thanks go out to Ivan Kondov for all the support and advice he provided over the last years, the interesting long and short discussions we had, and his always-helpful suggestions and answers to my questions.

Special thanks go to Julia for the good cooperation, for bearing with me even through some very harsh arguments, and for supporting me through the hard time of writing my thesis. Thanks!

Furthermore, I would like to express my gratitude to Holger Marten at the SCC for his support he provided during my graduation.

I would also like to thank Julia and Ivan for proofreading my thesis and helping to make it readable and understandable.

Thanks go also out to my parents and grandparents for always providing a nice stay away from the stress of a doctoral graduation.

Furthermore, I would like to thank the following people:

- Timo for creating the well-structured and written code base of SIMONA, which I really appreciate.
- Moritz for dealing with most of the Python code that enables running SIMONA.
- Frank and Konstantin, especially for the discussions about the power diagram and PowerSASA code.
- Julia and Carolin for bearing with me as long as it took to finish the implementation of the membrane model.
- All my voluntary and non-voluntary bug hunters that informed me about them while not being too upset.
- Thomas and Eugen for sharing the office with me and the many funny discussions we had.

Last but not least, I would like to thank all my friends for making my doctoral graduation an enjoyable time.

I would also like to express my gratitude to all members of the AG Wenzel at the INT, the Junior Research Group Multiscale Biomolecular Simulation, the SimLabs at the SCC, especially the SimLab NanoMikro, as well as Scientific Computing and Simulation group of the SCC for the pleasant stay during the last few years.

Finally, I acknowledge funding from the SimLab NanoMikro, the Baden-Württemberg Stiftung and the Landesgraduiertenförderung Baden-Württemberg. I would further like to acknowledge the computational resources provided by the Steinbuch Centre of Computing, the Institute of Nanotechnology and the bwHPC project. I acknowledge PRACE for awarding me access to resource HERMIT based in Germany at HLRS Stuttgart and JADE based in France at CINES.