DEVELOPMENT OF A MULTISCALE PROTOCOL FOR THE STUDY OF ENERGETICS OF PROTEIN DYNAMICS

NILS J. D. DRECHSEL

Department of Experimental and Health Sciences
Research Programme on Biomedical Informatics
Computational Biochemistry and Biophysics Laboratory
Universitat Pompeu Fabra

Ph.D. Thesis
TESI DOCTORAL UPF / 2013

Supervisor: Jordi Villà Freixa

Co-Supervisor: Ken A. Dill

Barcelona, 2013

This thesis is dedicated to my mother, entirely.

ABSTRACT

Multiscale Molecular Dynamics is a popular trend in the field of computational chemistry and physics. Coarse-grained force-fields have been around for years, and used independently, but used cooperatively with all-atom force-fields combines their advantages and cancels their disadvantages. This seems to be the case, however, only when they are both compatible. In this thesis, a Multiscale Molecular Dynamics Protocol is introduced, based on earlier work by Benjamin Messer, Z. Fan, Arieh Warshel, and in other parts by Christopher Fennel and Ken Dill. The protocol consists of the following tool-set:

- A parametrization machinery that created a new coarse-grained force-field named AmberCG.
- A multiscale thermodynamic cycle utilized within a free energy perturbation context to cooperatively use the best of coarse-grained and all-atom force-fields.
- A collective variable that performs a linearization of the phase space to improve separation of product and reactant states.
- A new algorithm to calculate functional quantities on spheres bounded by complicated solvent accessible surface areas - which as a special case calculates the amount of solvent accessible surface area.
- A novel algorithm based on simple one dimensional Depth-Buffers, to identify atoms which actively form the boundary of the solvent accessible surface areas.

Executing the protocol involves the following steps:

1. Construction of a coarse-grained force-field, based on an all-atom force-field. This involves setting up coarse-grained potentials and optimization of their parameters against selected reference structures and conformations.
2. Parametrization of a solvation model which is compatible to the force-field.

3. Usage of the coarse-grained force-field to sample the conformational space of a reaction.

4. Correction of the coarse-grained results with an all-atom force-field.

5. Analysis of the results using appropriate collective coordinates.

6. Reiteration until accuracies are met.

Alternatively, instead of using the methods in the protocol, they can be utilized stand-alone. They simplify calculations, thus providing speed-ups, while at the same time aiming to maintain or improve accuracy. Of course, there is no free lunch, and often the methods will include inaccuracies that exceed an acceptable threshold. However, the multiscale protocol is meant to be seen as an iterative technique, in which deficiency can be detected, and the protocol adjusted to restore balance.

RESUMEN

Las simulaciones de dinámica molecular multiescala (Multiscale Molecular Dynamics) son una tendencia al alza en el sector de la Química y la Física computacionales. Los coarse-grained force-fields o campos de fuerza de grano grueso han existido desde hace años, utilizados de forma independiente, y también en cooperación con all-atom force-fields o campos de fuerza de todos los átomos dónde se combinan sus ventajas y cancelan sus desventajas. En este último caso sólo es cierto cuando los dos force-fields son compatibles. En esta tesis, introduzco un protocolo de Multiscale Molecular Dynamics basado en parte a trabajos anteriores de Benjamin Messer, Z. Fan, Arieh Warshel, y también en los de Christopher Fennell y Ken Dill. El protocolo consiste el siguiente conjunto de herramientas:

1. Un método de parametrización con cuál creé un nuevo coarse-grained force-field llamado AmberCG.

2. Un ciclo termodinámico multiescala utilizado en un contexto de perturbación de energía libre para usar cooperativamente el mejor de los coarse-grained force-fields y el de los all-atom force-fields.

3. Una variable colectiva que realiza una liberalización del espacio de fases para mejorar la separación de los estados de productos y reactivo.

4. Un nuevo algoritmo para calcular las cantidades funcionales en esferas limitadas por complicadas superficies accesibles al solvente - que como un caso especial calcula la cantidad de superficie accesible a solvente.

5. Un nuevo algoritmo basado en un buffer de profundidad, para identificar los átomos que forman activamente el límite de las superficies accesibles al solvente.

La ejecución del protocolo implica los siguientes pasos:

1. Construcción de un coarse-grained force-field, basado en un all-atom force-field. Esto implica la creación de potenciales coarse-

grained y la optimización de sus parámetros contra las estructuras de referencia seleccionados y sus conformaciones.

2. Parametrización de un modelo de solvatación compatible con el force-field.

3. Uso del coarse-grained force-field para muestrear el espacio conformacional de una reacción.

4. La corrección de los resultados coarse-grained con un all-atom force-field.

5. Análisis de los resultados utilizando coordenadas colectivas adecuadas.

6. Repetición hasta alcanzar las precisiones deseadas.

Alternativamente, los métodos del protocolo pueden ser utilizados de forma independiente. Esto simplifica los cálculos y procura mantener, si no mejorar, la precisión. Sin embargo, todo tiene un coste y con frecuencia, los métodos incluirán inexactitudes que superarán el umbral aceptable. Aun y así, el protocolo multiescala es una técnica iterativa, en la que la deficiencia puede ser detectada, y el protocolo ajustado para restablecer el equilibrio.

PUBLICATIONS

Nils J. D. Drechsel, César L. ÁÀvila, Raúl Alcántara, and Jordi Villà-Freixa.
"Multiscale molecular dynamics of protein aggregation."
Current Protein and Peptide Science 12, no. 3 (2011): 221-234.

Nils J.D. Drechsel, Christopher J. Fennell, Ken A. Dill, and Jordi Villà-Freixa
"TRIFORCE: Tessellated Semi-Analytical Solvent Accessible Surface Areas And Their Derivatives"
Submitted to Journal of Computational Chemistry

Nils J.D. Drechsel, Christopher J. Fennell, Ken A. Dill, and Jordi Villà-Freixa
"A Multi-Layered Depth-Buffer for the Detection of Atoms Contributing to the Boundaries of Exposed Surface Areas"
Submitted to Journal of Chemical Theory and Computation

Other publications:

Mulero MC, Ferres-Marco D, Islam A, Margalef P, Pecoraro M, Toll A, Drechsel NJ, Charneco C, Davis S, Bellora N, Gallardo F, López-Arribillaga E, Asensio-Juan E, Rodilla V, González J, Iglesias M, Shih V, Mar Albà M, Di Croce L, Hoffmann A, Miyamoto S, Villà-Freixa J, López-Bigas N, Keyes WM, Domínguez M, Bigas A, Espinosa L
"Chromatin-Bound IκBα Regulates a Subset of Polycomb Target Genes in Differentiation and Cancer."
Cancer Cell. 2013 Jul 9. doi:pii: S1535-6108(13)00279-1. 10.1016/j.ccr.2013.06.003

ACKNOWLEDGMENTS

Giving enough credit to everyone, who was in some way involved in the completion of this thesis, is not an easy task, and surely I will not succeed. The Ph.D. was a journey, both scientifically and culturally, in which I had many companions. First of all I want to say thanks to my supervisor and mentor Jordi Villà Freixa. Jordi is an incredible person. He is not just extremely knowledgeable in so many totally different areas (really Jordi, how do you do that?), but he is also full of interesting ideas, and always an inspiration. I guess he often had it very difficult with me, because I never seemed to be doing what he was telling me to do, but he always knew when to give me space, and when to keep me close.

My second mentor, Christopher Fennell, was sharing my enthusiasm for the small details. His guidance through the last stages of my Ph.D. was indispensable. Thank you Chris, I think we have made, and still are, a really good team. I'd also like to specially thank Ken Dill. Ken is also a really amazing person. He and his wife Jolanda were doing regular bbq and pool parties at their place in Port Jefferson, and they created a fantastic environment. He took me into his lab unconditionally, gave me his personal bike for my daily journey to the Laufer Center, and provided me with financial support in a time when the financial crisis hit hard on Europe. Thank you Ken!

Another special thanks goes to César Ávila. He came to Barcelona for a short stay during his Postdoc and later as a Professor, and I think especially during his first stay we were a great team, both scientifically and as friends. César, like Jordi, is very knowledgeable in many things and somehow he managed to get accustomed to ADUN, our molecular simulator in record time. I won a bet once and made him wear a German football-shirt that he had to wear during the world-cup; he gave me a letter from Paul, the octopus. Those were good times!

When I joined Jordi's laboratory in Barcelona, I was welcomed very warmly by the existing members, or those that joined shortly after: Kashif, Toni, Ignasi (who corrected my Spanish abstract; many thanks!), Sinan, Raúl and Norma. In the following months and years, we have been one hell of a crew. Over the years of course, people come and

people leave, but the spirit of the lab stayed. Julien, Nate, Noelia and Stefan it was a very nice time with all of you.

I also want to say thanks over the ocean to the Stony Brook people. Especially Alberto, Arijit, Justin, Adam and Amber made me very welcome, showed me some weird American traditions (like the Doughnuttiest day), and were fun to have around.

Back to Europe, I cannot thank my family enough, they all supported me a lot. My mother always believed in me, helped me in every way possible, and I knew that she would have loved to see this day coming. I want to say thanks to my father, who I unfortunately have neglected a bit too much during the past months - I'll make up for it! My brothers Dirk and Christian - you guys are my soul. I love you and cannot nearly express enough what you mean to me! Thanks also to Paul, my uncle, who was always interested in my studies.

There has been one very special person in my life, and she probably supported me like no one else. Thank you Özgen, without you a lot would have been different. It was surely not always easy to be my girlfriend, especially during the time when I was writing the thesis. Özgen is not just an incredible person, she is not just an excellent scientist, not just an inspiration to a lot of people, but she also means the world to me.

There are more people in Barcelona that made this thesis happen in one or the other way. Emrecim, Myk, Jana, Leszek, Radek, Andrey, Zina, Mili, Michi, Ana, Besray, Thasso, Billur, Jelena, Luciano, Blanca, Nati, and many more - thanks to everyone of you.

Last but not least, thanks also to Philipp, Inge, Rami and Paddy. Rami and Paddy have been friends for long; Philipp has always been a very good friend since we met each other in Karsruhe. Coming from two very different areas, we somehow managed to both explore the field of molecular dynamics. Sometimes it's weird how things turn out.

PREAMBLE

Multiscale Molecular Dynamics is a promising direction in computational biology. In a world in which hardware manufacturers cannot satisfy fast enough the computational demand of ever growing molecular simulations, and in which there is a widening gap between scientific desire and reality, it stands to reason to utilize existing hardware more efficiently to try to close this gap. Many things in our daily lives are multiscale to begin with. We think in multiscalar terms when we plan a route from one end of the country to the other, or when we design complex devices or software. Our body works multiscalar by combining low level entities to higher order structures, which themselves belong to even higher order structures.

This thesis is a step into this direction. I aim here to give researchers a piece of their time back. Time that they had to invest to push simulations beyond existing limits - so that they are able to push even further. Multiscalability in the context of molecular dynamics involves changes on many levels: The forces with which the dynamics are progressed, the representations of both the solute and solvent, and the cooperativity between multiple multiscalar resolutions. We leave the "more physical world" and enter the "less physical world", without going too far of course, because our reasoning is still guided by physics.

The structure of this dissertation reflects these properties. I start with an introduction to the history of molecular dynamics, which is, I think, not appreciated enough. There are a few key people who have not just founded this field, but who have constantly contributed major ideas and techniques and brought forward science at an amazing rate. Next, I move forward to statistical mechanics, its importance to multiscale methodologies, and the reason why it is the key ingredient and basis (or should be) in every computational biologist's work. I end the coarse introduction with a general chapter about solvation, and hopefully provide enough proof that modeling the solvent is a cumbersome task.

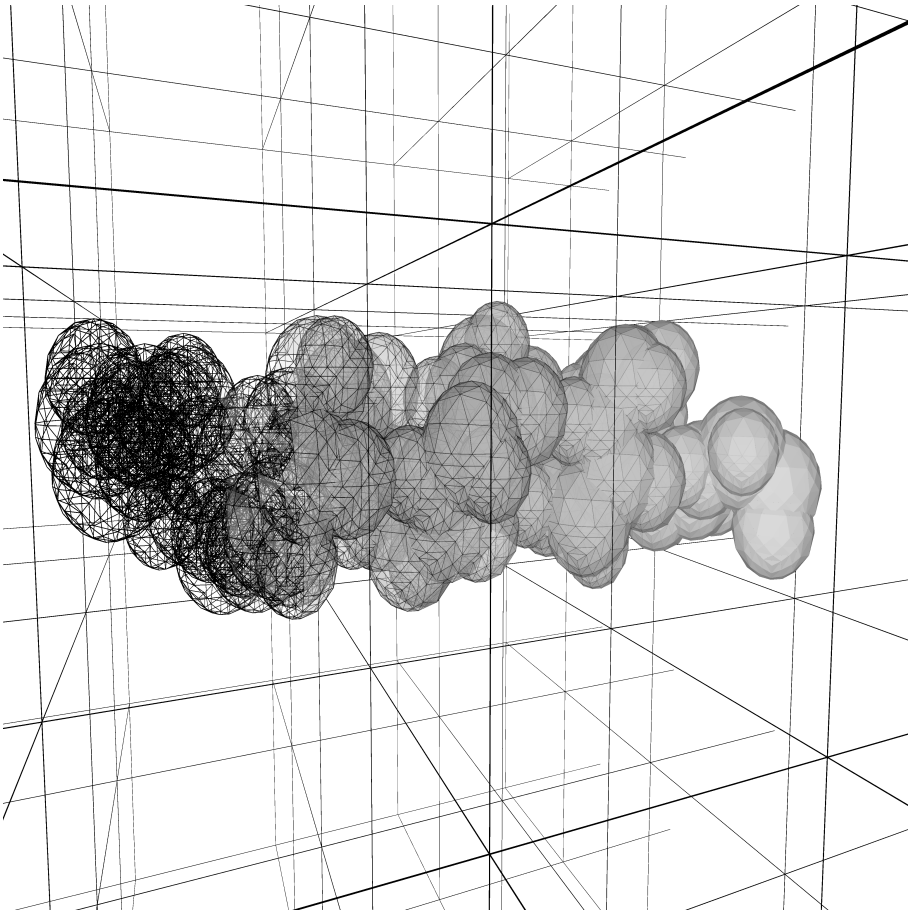
Now we move down one level of resolution, and look at some methods, which have been utilized, or which could be excellent replacement for used methods, in more detail. This is usually accompanied by some equations, on which I explain certain differences, advantages

or disadvantages. The thesis contains five main results, three of which have been published or submitted to peer-reviewed journals. It starts with an article about multiscale methodologies in the context of protein aggregation, in which a thermodynamic cycle, based on previous work by Arieh Warshel et al., is presented, which is the basis for our use of multiscale methodology. Later on, a collective variable based on contact maps is proposed. It lifts some problems that I encountered while working on free energy surface representations for the chapter on AmberCG, a coarse-grained force-field based on the Amber force-fields, which is the subject of the consecutive chapter.
After these first three results, I introduce two new algorithms that play a role for non-polar solvation. Back when I started at the laboratory, one of my first surprising tasks was to create a small addition to our molecular simulator, ADUN. The addition consisted of a novel non-polar solvation method, Semi-Explicit Assembly (SEA), envisioned by Christopher Fennell and Ken Dill, that had to be made compatible with ADUN. I remember the day when I connected the two softwares, and discovered that with the finite difference derivatives that we got from SEA we wouldn't be able to run efficient molecular dynamics simulations. It was then that I got the new task of quickly implementing analytic derivatives for the method. Three and a half years later, we are finally at the point in which we have a framework to actually approach this. The result is a powerful new algorithm that is able to integrate arbitrary functional values on a sphere, bounded by a complex series of spherical arcs. This algorithm, together with an optimization is given in the last two chapter of the results section, which is ended by an outlook towards where we can go from there.

CONTENTS

LIST OF FIGURES

LIST OF TABLES



Artistic representation of the backbone of an alpha helix in a simulation grid, undergoing a transformation from a wireframe to a solid representation

1

INTRODUCTION

1.1 A SHORT HISTORY OF MOLECULAR DYNAMICS

The arguably first published molecular dynamics simulation was performed by Bernie J. Alder in collaboration with Thomas E. Wainwright in 1957[1], in a time where computers used magnetic cores as memory storage, magnetic tapes and punched cards as input devices and magnetic drums for program code and bulk storage. Their simulation set out to study phase transitions of a liquid and used up to 500 hard sphere particles in a periodic boundary box. More simulations on liquids, solids and gases were performed in the following years[2], notably by Aneesur Rahman[3] in 1964, who calculated the motions of Lennard-Jones spheres for a box of 864 argon particles to study two-body correlations. Although the simulation was not much larger in terms of the amount of particles, the temporal range of the simulations was extended by orders of magnitude. This was possible by using more sophisticated machinery, i.e. computers based on transistors and internal registers.

During that time, simulations relied on internal coordinates, while as it is well known, modern molecular dynamics is dominated by simulators based on Cartesian coordinates. It all changed in 1966, at the Weizmann institute, when modern molecular dynamics was born. In those years, Arieh Warshel and Shneior Lifson were both working on the consistent force-field[4,5], a collection of potentials that were derived from equations collected by Bixon and Lifson[6] for cycloalkanes, which were based on studies by Bartell and Kohl[7] on C-C and C-C-C bond lengths and angles, Hendrickson[8], Wiberg[9] on angle force constants, Hendrickson[8], Wiberg[9] and Pitzer[10] on torsions, and Hendrickson[8] on non-bonded potentials. The development was intensified when Michael Levitt joined the team to investigate the minimas of arbitrary molecules[11]. Arieh Warshel realized that the force-field could be extended by incorporating quantum mechanical treatments[12,13], which

would become the basis for modern hybrid quantum mechanics/-molecular mechanics and the empirical valence bond method (EVB)[14].

Years later during the 70s, Frank H. Stillinger and Rahman[15–17] simulated liquid water at multiple temperatures and provided insight into a balanced simulation technique. It was during this time, that Martin Karplus and Andy McCammon performed their ground breaking studies on the bovine pancreatic trypsin inhibitor[18;19] in 1977, in which, for the first time in history, a protein was simulated by all-atom molecular dynamics. A similar study on the same protein was performed earlier by Arieh Warshel and Michael Levitt with the first coarse-grained molecular dynamics simulation[19;20] in 1975. At the same time, Arieh Warshel managed to produce the first simulation of a biological process, the photoisomerization of the rhodopsin complex to prelumirhodopsin intermediate[21].

Following Warshel's research on Cartesian potentials, Bruce Gelin used the developed code and modified it into a software that would later be the foundation for CHARMM, in order to carry out studies on the bovine pancreatic trypsin inhibitor and hemoglobin[19;22]. In a similar field, David A. Case, who later became overseer of the development of AMBER, a powerful software suite for molecular dynamics initiated by Peter A. Kollman, was using the methodology for studies on myoglobin[23]. Continuing on Gelin's work, Andy McCammon completed the software into a real molecular dynamics simulator.

During these years, researchers had to face computational limits due to the scarcity of supercomputers. On the software side, in the following years, a technique called umbrella sampling, known from Monte-Carlo simulations[24], was adapted for molecular dynamics[25]. It improved the abilities to obtain proper sampling of processes under study, thus reducing computational demand, and became very popular over the years, followed by its extension in 1992, the weighted histogram analysis method[26]. Improvements on the one hand were however negated on the other, when water molecules were introduced[27] into the previously only in gas phase executed simulations. Warshel was one of the first to perform free energy perturbation experiments in explicit solvent, utilizing umbrella sampling[28]. Combined with the surface constrained all atom solvent[29] they were able to calculate free-energy profiles for benzene-like molecules.

Most of the initial studies performed in Karplus' laboratory were performed on computers capable of executing between 20 000 and 200 000 floating point instructions per second. Considering the fact

that a system, as small as the bovine pancreatic trypsin inhibitor with united atom model and without water contains around 500 atoms, thus more than 500 000 interactions (disregarding cutoffs), the simulations were painfully slow.

Klaus Schulten, Professor at the Technical University in Munich, was aware of these problems. Lisa Pollack wrote a nice summary[30] of the events unfolding in the year 1987* . Schulten arguably was the first one that saw the necessity of including computer scientists in his team, to create efficient simulation suits, and so after recruiting enough manpower, he set out to create a new molecular simulations platform from scratch, that would later be run under the name NAMD. In the same time period, Kit F. Lau and Ken A. Dill developed a lattice model to study folding behavior in simplified systems[31].

Like Schulten, but several years later in the mid 90s, Herman J. C. Berendsen, David van der Spoel and Rudi van Drunen presented GROMACS[32], a molecular dynamics simulator that should become one of the fastest in its field. Berendsen already influenced molecular dynamics in the mid 80s with the development of the Berendsen thermostat[33], a popular method to rescale velocities for constant temperature simulations using a proportional scheme. Berendsen's method was not the first thermostate to find appreciation. A method by Hoover[34;35], that included an additional term to the forces to keep the kinetic energy constant, was modified by Nosé with an extention that introduced additional degrees of freedom into the system. The method should later be known as the Nosé-Hoover thermostat[36]. 10 years before, Steven

* Back in 1987, Klaus Schulten, Professor at the Technical University in Munich, was interested in simulating an important macro-molecule that played part in the process of photosynthesis, but was set back by the fact that, including water, the simulation would require to calculate the dynamics of over 100 000 atoms. Due to a fortunate encounter with Helmut Grubmüller, a student at the university, he became interested in the idea to create his own "home-made" supercomputer. He recruited Grubmüller, who at the time was trying to find funds for creating a faster computer. Together with Helmut Heller, who was already a member of Schulten's laboratory, they set out to build one. Schulten himself took a lot of risk when he waged all his grant money, around 60 000 DM, on the hardware of the computer. The computer was built using a network of transputers, a special processor that was meant to run in parallel with other transputers. This was a cheap way to increase computational power, but needed special software, construed for parallel processing, to work. They created a parallel code that was named EGO. In the mean-time, Schulten accepted a new position at the university of Illinois. Once the computer was ready, to avoid delays with shipping, he transported it himself in his backpack through customs at Chicago airport, despite the problematic bylaws regarding the transportation of supercomputers during the Cold War.

A. Adelman and Jeff D. Doll introduced the Langevin thermostat[37], that added a frictional force to push the temperature to a set temperature. It was also the birth of usable implicit solvation methods, like Generalized Born[38], which is an approximation to the Poisson-Boltzmann method to calculate solvent electrostatics by using point-charges and estimation of the buriedness of individual atoms in the macro-molecule.

At the end of the 90s, T. Simons, Charles Kooperberg, Enoch Huang and David Baker presented the core of Rosetta[39], a program that until nowadays is still the de facto leader in protein fold prediction. It uses a simulated annealing protocol of fragments from a pdb database sweep to create native like structures that are scored using Bayesian scoring functions. In the same year, Christopher Jarzynski developed his infamous Jarzynski-Equality[40], an equation to extract equilibrium free energies, from non-equilibrium measurements. Kevin W. Plaxco, Kim T. Simons and Baker emphasized the correlation between contact order and folding rates[41], Yong Duan and Peter A. Kollman published the first simulation extending 1 µs[42], and Yuji Sugita and Yuko Okamoto presented a novel sampling technique called Replica-Exchange[43], that should prove incredibly useful and become indispensable in modern molecular dynamics. It was now possible to start multiple parallel simulations of the same macro-molecule in different temperatures, and to exchange temperatures throughout simulation, in a way in which detailed balance was still ensured.

With the 2000s came a time of a massive technological burst. It was the birth of powerful supercomputers and frameworks like the Earth Simulator, Blue Gene, Cray, and many more that provided the computational power to increase the sizes and temporal ranges of simulations to study artificial folds[44], proton pumping through an anti-porter[45], computational enzyme design[46], and millisecond protein folders[47]. Outside the supercomputing centers, inside the laboratories, computer clusters utilizing GPUs became supported by molecular dynamics simulation suits or libraries[48–50]. On the software side, to speed things up, improved implicit solvation models were derived[51;52], coarse-grain methodologies improved[53–55], and new sampling schemes conceived[56;57]. In the last few years, molecular dynamics was taken beyond scientific laboratories, into the homes of interested people. Folding@Home[58], PS3-Grid[59], GPU-Grid[60], Rosetta@Home[61] or games like Fold-it[62] are examples.

1.2 STATISTICAL MECHANICS

Statistical mechanics provides the link between the microscopic scale of molecular dynamics simulations to macroscopic quantities, which are of real interest. Such quantities could constitute of free energies or rate constants, all of which can be expressed with probabilities. In every molecular dynamics simulation, the movement of the atoms is governed by the derivatives of the potential energy. However, equilibrium and steady states are governed by free energy, which is a measure of the probability of a macro-state, i.e. a set of micro-states (conformations) of equal energy. Free energy ΔG is an aggregation of enthalpic parts ΔH, which constitute of kinetics, potentials energies and displacements costs, and an entropic part ΔS, which is a direct measure of the amount of conformations of a certain molecular ensemble.

$$\Delta G = \Delta H - T\Delta S \tag{1}$$

In that context, entropy can be seen as the width of a valley in free energy space, which constitute a molecular ensemble, while enthalpy would be related to its depth, dependent on the temperature T.

A molecular system which is advanced in time with methods that obey detailed balance will at some point reach an equilibrium distribution and will stay there[63]. Detailed balance is a condition which ensures that the transition of a system is time-reversible in equilibrium, i.e. that theoretically the system could return to the starting state in the same path through which it advances. This has impact on the rates of a system: At equilibrium, forward and backward rates have to be equal. So for a two-state process with states A and B:

$$A \underset{k_1}{\overset{k_0}{\rightleftharpoons}} B \tag{2}$$

at equilibrium, detailed balance requires

$$k_0[A] = k_1[B] \tag{3}$$

If a system is at equilibrium, it is possible to extract thermodynamic quantities, i.e. quantities that define the system as a whole. Methods that do not obey detailed balance will not converge a system to the Boltzmann distribution, however they might still be able to sample the Boltzmann distribution, if they obey the weaker balance condition[64].

Balance condition does not impose direct reversibility, but rather ensures that a process is reversible through circular transitions[65].

The difference in free energy between reactants and products can principally be expressed as the ration of partition functions of both potentials.

$$\exp\left(-\Delta G_{A\to B}\beta\right) = \frac{Q_B}{Q_A} \tag{4}$$

Where Q_A and Q_B are the partition functions of potentials A and B, and β the inverse Boltzmann constant times temperature. Partition functions are therefore powerful entities, but they are computationally demanding to calculate. The probability of a system being in a certain microstate is related to its Boltzmann factor, normalized by the partition function.

$$p_i = \frac{\exp\left(-E_i\beta\right)}{\sum_j \exp\left(-E_j\beta\right)} \tag{5}$$

1.2.1 *The Sampling Problem*

The conformational space of macromolecules is large, and grows rapidly with the number of atoms that are involved. A full sampling of any free energy surface is only possible for small molecules, and additionally impeded by the presence of small minima, caused by frustrated pathways[66]. Frustration occurs when different inter-atomic potentials compete with each other, causing the overall surface to assume a rough shape, which slows down exploration. This results in an incomplete sampling for simulations that are bound by technical / economical limits. When incomplete sampling plays a role, the simulation cannot achieve a Boltzmann distribution, which would yield inaccurate results in the analysis of the simulations, when the former is required[67].

In the discipline of protein folding, which is the biological process of an polymer chain forming a well defined secondary and tertiary structure, frustration is a major point of discussion. Levinthal formulated his famous paradox in the late 60's[68], comparing the combinatorial impossibility of performing a random search in phase space (this would require more than 10^{10} years for a 100 amino-acid large protein, if each amino-acid would only exhibit two conformations and they could be searched in 1 ps[66]) with the speed proteins actually fold in reality.

Multiscale molecular dynamics is an attempt to lift the inadequacy to generate sufficient data on a biological process. It is allowing a system to move faster through phase-space, yet narrowing its exploration with physical constraints.

1.2.2 *Free Energy Integration Schemes*

The above notions about equilibrium states become important when the subjects of interest are free energy differences between different states in a biological process. This can involve for example protein folding, ligand binding, protein-protein interactions, conformational changes. Various methodologies[69;70] exist which either operate at equilibrium, or at non-equilibrium by utilization of Jarzynski's equation[40] or its derivatives[71;72].

In all free energy difference schemes, a system is evolved from a starting distribution with potential energy function $U_0(\boldsymbol{x})$ to a final distribution with potential energy function $U_1(\boldsymbol{x})$ which represent ensembles of reactants and products, e.g. unfolded and folded states. The evolution is performed by following a path usually parametrized by the variable λ, thus the path starts with potential U_0 which corresponds to $\lambda = 0$ and ends in potential U_1 which corresponds to $\lambda = 1$. Intermediate values of λ correspond to mixed potentials in the form of :

$$U^{\lambda}(\boldsymbol{x}) = (1-\lambda)U_0(\boldsymbol{x}) + (\lambda)U_1(\boldsymbol{x}) \tag{6}$$

Often, forward and reverse integrations can be combined advantageously, and if detailed balance is upheld, then both directions can be simultaneously collected from a single integration[73].

1.3 MULTISCALE MOLECULAR DYNAMICS

Multiscale molecular dynamics is a general term describing dynamics on different resolutions, e.g. mesoscale and atomistic levels[74], or coarse-grained and explicit representations[55]. The coupling between those resolutions can take multiple forms, and varies for post-processments[75], parallel[76] or sequential schemes[54]. The type of multiscale methodology that we are concerned with specifically is of technical matter and does not involve resolutions above the atomistic scale, i.e. information flow to meso or macro-scales of higher order cellular structures is not regarded. We employ coarse-grained techniques,

coupled to an atomistic system, to accelerate and simplify molecular dynamics simulations.

In all coarse-grained multiscale molecular dynamics schemes, a part of the atoms is replaced by simplified structures which are able, on average, to exhibit the energetics and dynamics of the substituted section.

Coarse-grained force-fields embody molecules and their interactions with simplified representations and potentials, mainly to achieve a speed-up in computations and therefore simulations. Multiple degrees of freedom are integrated into a few[77], causing to a decrease in the number of interactions, vibrational modes and frustration. The field of coarse-grained force-fields is vast. In the past, force-fields have been designed for specific purposes, biased towards a reference structure in the case of Gō models[78], where a biasing potential was added that would steer the simulation towards the reference, or created with elastic networks[79]. Biasing towards a structure can cancel out effects created by the coarse-graining of side-chains, in which chemically specific interactions are lost[80].

One of the first surprisingly successful models was created by Michael Levitt[81], during a period when computer time was rare and expensive. Coarse-graining seemed to be a cheap solution to extend computational limits, and is used upon today to go beyond the time-scales of all-atom explicit simulations, whenever detail and discreetness is not of utmost importance, and more average thermodynamic properties acceptable.

Coarse-grained force-fields can be broadly decomposed into the number of beads, used for their description, and the type of parametrization that was used for their development. For proteins, the number of beads can range between 1, usually located around the alpha carbon, up to 4-6, with increasing level of detail, i.e. representation of beta carbons and different parts of side-chains. It is also possible to leave the main-chain representation explicit[55], or use simplified but explicit water[82].

1.4 COLLECTIVE VARIABLES

The concept of free energy, introduced in the section about statistical mechanics (section 1.2), requires the generation of ensembles which are proxies for macro-states. Ensembles are usually gathered by projecting conformations from a very high-dimensional conformational

space to very low dimensional reaction-coordinate space, assuming that the real conformational space is actually a low dimensional manifold. Ususally, the partition function of state i can be estimated by calculating how often that state is visited[83].

$$Q_i = \sum_j n_{ij} \tag{7}$$

Where Q_i is the partition function of state i and n_{ij} is an entry in the transition matrix. Since we know that the ratio between partition functions relates to the free energy difference between two states (see equation 4), it is possible to extract these quantities from the surfaces.

The reaction coordinates can be understood as giving qualitative and quantitative information about the state of a reaction. Reactants and products would both occupy space in this reaction-coordinate space, which can provide visual qualitative feedback, as well as enabling the calculation of free energy differences, thus yielding quantitative data. The reaction-coordinates differ largely between the type of reaction which is investigated, and can compromise distances between atoms or domains, native contacts or contact order, structural discriminators like the radius of gyration, comparative metrics like the root mean square deviation of universal similarity metric, or totally non-descriptive quantities derived from principle component analysis[84–86]. Principally, every function that either compares a reference to a template, or yields information about the state of a single conformation without comparison can be employed. As such, we can borrow heavily from adjacent fields. In the end, the quality of a reaction coordinate differs with the underlying data and the quantities that should be extractable from it. For example, it can be measured by borrowing from transition path theory. A transition path is a pathway which leads from the reactants state to the products state or vice-versa without recrossing[87]. $p(TP|r(\boldsymbol{x}))$ is the probability distribution of being in a transition path, given coordinates $\boldsymbol{x}$ projected on a reaction coordinate $r(\boldsymbol{x})$. If the distribution has a sharp and high peak, we can conclude that the coordinate is well chosen, because it translates into the collapse of the transition states into a single value of the reaction coordinate (see Best et al.[87] for excellent illustration).

In the special field of protein folding, the arguably best reaction coordinate would be P_{fold}, the probability of folding before unfolding[88]. P_{fold} distinguishes clearly between two states. The folded state ($P_{fold} = 1$) and the unfolded state ($P_{fold} = 0$). For every conformation

in between, it would yield the probability, that the protein either tends to move towards the folded, or unfolded state. At $p = \frac{1}{2}$ the reaction would be in a clearly defined transition state, where both paths are equally probable. Despite its supreme characteristics, P_{fold} is cumbersome to calculate, and unusable for ab initio calculations, since for this type of simulations, the folded state is unknown.

Using metrics for reaction coordinates poses a similar problem. A metric needs two coordinates to generate a distance value. What should be used as the reference coordinate? We see therefore, the choice of reaction coordinate very much depends on the problem under investigation. For most of our folding studies, we tried to reproduce known folded structures. The root mean square deviation metric[89], or with the radius of gyration[90] were easily accessible reaction coordinates with physically meaningful properties.

1.5 SOLVATION

1.5.1 *Explicit Solvation*

Solvation is the process of putting a solute into a solvent - in case of protein studies usually water, because it is the native environment for proteins. This process affects both the solute and the solvent in proximity to the solute fundamentally on multiple levels: (1) A cavity in the solvent is created. If the cavity is large enough, then the hydrogen network is broken and needs to be recreated[91]. For large solutes, the solvent will dewet its surface[92] around hydrophobic particles[93], initiating a potential hydrophobic collapse, if the solute is a protein. (2) Water is a much larger dielectric than protein; the ratio can be as high es 40:1 for the protein interior due to mainly uncharged inflexible amino-acids, and as high as 8:1 for the exterior due to charged and very flexible amino-acids[94]. As such, solute-solute electrostatic interactions are screened by the reorganization of solvent due to the new electric field. (3) Non-polar and polar first solvation shell effects occur in the solvation layer in direct contact with the solute. Non-polar solvation includes cavity formation, and van der Waals interactions between the solute and solvent.

The unique properties of water, i.e. the creation of a hydrogen bond network[95], presence of a dipole moment, and its polarization ability, renders it difficult to model. Water forms significant structure around solutes, to up to 14 Å, but ordering effects are present even beyond

this point and can, if not rebalanced, causing a significant heat-up of a simulation system[96]. It is not surprising that there is considerable spread in water representations, and the search for a perfect model still contiues. These models usually differ by the number of point charges, which are distributed over the molecular plane, the individual partial charges, van der Waals parameters, and polarization abilities.

An early water model was ST2, by Stillinger and Rahman[97]. It is based on 5 partial charges, which take into account the hydrogens and the lone ion pairs in tetrahedrical positions around the oxygen[98]. Simpler models include the transferable intermolecular potential functions TIPS2, TIPS3 and TIP3P[98;99] which place 3 partial charges on the oxygen and hydrogen atoms. This reduces pairwise distance evaluations from 17 to 9, considering that van der Waals energies are only calculated between oxygen pairs. Another example for 3 point charge models is SPC, the simple point charge model. All these models differ in the equilibrium bond angles, point charges and oxygen-hydrogen distances. Tests on radial distribution function g_{oo}, g_{oh}, g_{hh}, which describe pairwise correlations between oxygen-oxygen, oxygen-hydrogen and hydrogen-hydrogen interactions have shown that some water models describe experimental results better than others[98]. Furthermore, when compared to more complicated point charge water models like TIP4P and TIP5P, then the simpler models provide not enough structure in oxygen-oxygen correlations[100]. However, when other quantities are taken into account, e.g. lifetime, average number, and average energy of hydrogen bonds, then SPC is competitive to TIP3P or TIP4P, which are suggested to be preferential water models for simulations[101].

1.5.2 *Boxed Water*

Molecular dynamics simulations need to have a finite amount of simulated particles, the less the better. The simulation area of solvated molecules therefore cannot extend arbitrarily, but must be confined. The usual strategy is to utilize a box that is adjacent to periodic images of itself[102], which removes artificial surface effects[103]. This creates the illusion of an infinite amount of water molecules around a solute, with the disadvantage that correlations between spherical entities are no longer radial, thus water molecules are not longer correctly oriented[104].

In a different approach, called surface constraint all-atom solvation, a spherical solvation environment, centered around the solute[105], is established. The method tries to represent infinite solvent through a limited solvation sphere, which is split into 3 segments. In the first segment, the solute including first solvation shell is located. In the second segment, surrounding water, and in the third bulk water, which is modeled implicitely. The water in the first two segment is treated explicitely, i.e. all interactions between and inside the segments are represented by pairwise expressions. Interactions between the first segment and the bulk water can be approximated using a Born or Kirkwood expression, depending on the complexity of the charge distributions. For the interactions between segments two and three, i.e. where the "surface" water molecules interact with infinite bulk water, a constraint potential is created that punishes "escaping" water molecules. The potential is derived by setting into relation the actual position of a water molecule, by the probability distribution of their expected position.

1.5.3 *Thermodynamic Cycles*

A general therodynamic cycle for solvation is shown in figure 1. The cycle is compartmentalized into 6 parts[106], in which the molecule is discharged, solvated and recharged. Using such a cycle to determine the solvation free energy allows for the compartmentalization into additive terms. The cycle can be arbitrarily extended, for example to calculate protein-ligand binding free energies. A very convenient method is based on Protein Dipoles Langevin Dipoles, used with their scaled version in combination with Linear Response Approximation (PDLD/S-LRA)[107]. The linear response approximation is valid, when the free energy functions of the reactant and product states have the same curvature. It accelerates the free energy calculations significantly, because intermediate steps in the mixed potential of reactant and product are neglected[108].

The idea behind PDLD is to model the solvent microscopically as dipoles on a grid, in which the density of the grid is chosen to match the actual solvent density[109]. These dipoles can be polarized according to the Langevin expression, which relates the average orientation of a dipole to the magnitude of the surrounding field[110].

PDLD/S is the semi-macroscopic version of PDLD, and involves splitting up the individual solvation free energy contributions into simpler terms. The terms correspond to transferring the environment

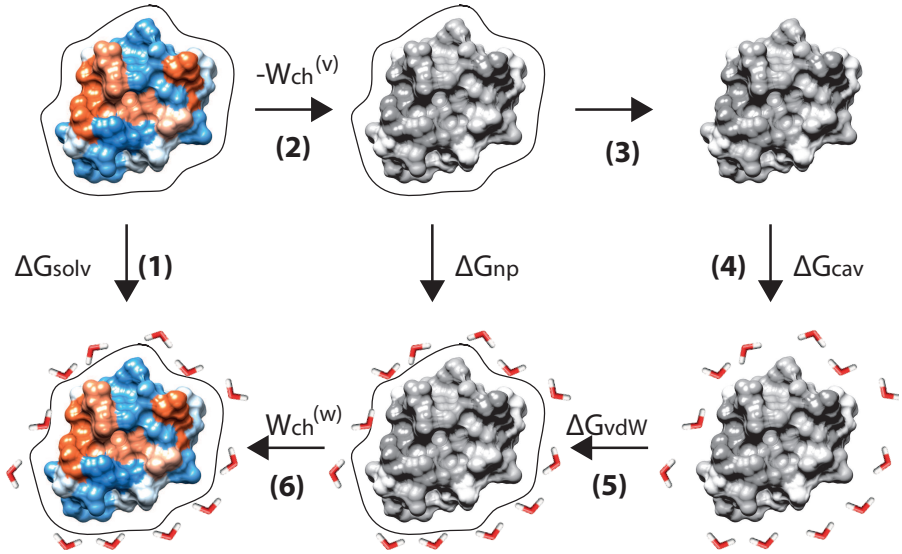


Figure 1: Thermodynamic cycle for the evaluation of solvation free energies of a given molecule: (1) The physical pathway. (2) The protein is decharged in gas-phase, i.e. all its charges are set to zero. (3) Van der Waals interactions between the solute and solvent are removed. (4) It is put into water. (5) Van der Waals interactions are turned on, here, the costs of creating a cavity, breaking of hydrogen bonds and first solvation shell effects play a role. (6) The protein is recharged.

from water dielectric into protein dielectric, discharging the ligand in the protein dielectric, and transferring the environment back into water dielectric. Now, similar to the thermodynamic cycle from figure 1, van der Waals interactions are turned off, and the ligand inserted into the protein. Once at that point, the previous steps are redone in inverted order. The extra step of changing the dielectric simplifies transferring the charges into the real protein environment[111].

1.5.4 *Implicit Solvation*

In the domain of molecular dynamics simulations, it stands to reason to simply fill a box with simulated solvent molecules and perform standard calculations. All of the solvation effects would then be treated explicitly by the corresponding force-field. Another way is to treat the effects implicitly by assuming that water is a homogenous dielectric continuum. When a thermodynamic cycle is used to split solvation

energetics into multiple parts (see figure 1), a distinction can be made between polar and non-polar effects. Segments (4) (creation of a cavity) and (5) (dispersion interactions) correspond to the non-polar part of solvation and Segments 2 (discharging) and 6 (recharging) to the polar part.

In implicit solvation, detail about the actual orientation and locations of water molecules is lost, and replaced by an averaged potential of mean force. The calculations return free energies, as such we speak about non-polar and polar solvation free energies. Using implicit solvation has generally multiple advantages over explicit solvation[51;112–119]:

(1) The amount of interactions are reduced. Adding one water molecule to a system of n atoms, adds $O(n)$ interactions to the system, so the cost of interactions is $O(n^2)$. Since proteins reside in three dimensional space, and proteins need to be solvated generously dependent on their diameter, the increase in water molecules itself is of the order $O(n^3)$. Even if the amount of interactions is limited by enforcing a spatial cutoff, the amount of solvent-solvent and solvent-solute interactions will outweigh all other interactions.

(2) Interactions add frustration to the energy landscape. Reducing interactions decreases frustration, and smoothens the landscape, hence simulations can proceed faster.

(3) Instantaneous relaxation of the solvent. When a protein is put into water, or if a conformational change, etc. disturbs the solvent equilibrium, it needs time to relax and return to equilibrium conditions. In implicit solvation methods, instantaneous response of the solvent is assumed, removing the necessity to return to equilibrium.

Disadvantages consist of the intrinsic inaccuracies of the methods, the loss of discreteness (which plays a role once single water molecules are involved in physical or biological processes), and difficult analytical functions for gradients which are expensive to evaluate.

In terms of polar solvation, which treats electrostatic effects, it exists a clear hierarchy, starting from all-atom, to dipolar fluid, dipole-lattice to continuum models[120]. We have already seen an example of dipolar models, the PDLD method. Continuum methods involve the Poisson-Boltzmann methods, in which the protein is modeled with point charges, located at the center of each atom[112–114]. In principle, Poisson-Boltzmann approaches treat the solvent and solute as a dielectric continuum, and calculate the effect on the protein dipoles explicitely, thereby reproducing PDLD/S results, and vice-versa[121].

More macroscopically, Generalized Born models[51;118;119] are derived from the Born equation for the transfer of a single ion from gas phase into solvent. They rely on Born radii, which change during a simulation and thus have to be calculated frequently. The radii resemble the buriedness of an atom inside the protein and need to be calculated with respect to all other atoms, usually by integration of the excluded volume[119].

1.5.4.1 *Non-Polar Solvation*

From the thermodynamic cycle in figure 1 it can be concluded that the non-polar solvation free energy is the sum of the costs of creating a cavity and attractive van der Waals energies.

$$\Delta G_{np} = \Delta G_{cav} + \Delta G_{vdW} \tag{8}$$

Based on previous results[122], in which it was found that the non-polar solvation free energy for alkanes largely correlates with their solvent accessible surface area, the non-polar solvation free energy is usually modeled with a γA model, in which the surface area is multiplied by an empirical surface tension coefficient[106]. Principally, with such an approach, some polar first solvation shell effects could be approximated as well - any effect that correlates with the exposed surface area[123].

1.5.5 *γA approaches*

γA approaches are modeled proportional to the solvent accessible surface area and are usually defined by the following equation:

$$\Delta G_{np} = \sum_i \left(\gamma_i A_i\right) + b \tag{9}$$

Often, a single tension coefficient γ is used instead of one per atom type. Values for the coefficients can differ significantly between $5\,\text{cal mol}^{-1}\text{Å}^{-2}$ and $138\,\text{cal mol}^{-1}\text{Å}^{-2}$ which is more than an order of magnitude[106]. coefficients can be calculated by a fitting to alkane transfer free energies[124]. The A usually refers to the solvent accessible surface area of a molecule, which has been defined by Lee and Richards in the following way[125]: *"Accessible surface area, A of an atom is the area on the surface of a sphere of radius R, on each point of which the center of a solvent molecule can be placed in contact with this atom without penetrating*

any other atoms of the molecule. The radius R is given by the sum of van der Waal's radius of the atom and the chosen radius of the solvent molecule.". For water, the radius R of the solvent is usually set to 1.4.

1.5.5.1 *Semi-Explicit Assembly*

Semi-Explicit Assembly is a novel method to calculate non-polar solvation free energies, by calculation of a more accurate solute-solvent boundary[126]. A big drawback in the γA approach is its ignorance of solute geometry, i.e. once the solute differs from a linear form (as it is the case for amino-acids), the approximation doesn't hold anymore[127]. Furthermore, in γA, two effects are expressed in the same simple equation - the cavity formation cost and the attractive dispersion energies, where the latter is constantly underestimated[126]. The crucial part is the amount of error that can occur. Systematic tests by van Gunsteren et. al have shown[128] that errors are in the range of 11 kcal mol^{-1} to 67 kcal mol^{-1} for protein like models. The error weighs even more heavily if we think about the fact that the free energy difference between native and denatured basins can be as low as 10 kcal mol^{-1}, independent of the size of the protein[129]. This is due to the favorable van der Waals interactions from interior atoms, which do not have a surface accessible to the solvent and are therefore neglected from the calculation of non-polar contributions.

The method therefore tries to estabish a solvent-solute boundary which is deformed by the relative attractions and repulsions of atoms adjacent to the first solvation shell and their immediate naighbors. Every atom is modeled as a source for a Lennard Jones potential, originiating from its center and emanating throughout space. This Lennard Jones field has minima, in which solvent atoms feel the most attraction, and are likely to reside. The set of minima lies in a two dimensional manifold, surrounding the protein, which ultimately determines the non-polar free energy potential. It is subsequently probed, and the acquired quantities compared to precomputed values that relate to non-polar solvation free energies for single atoms.

1.5.5.2 *Solvent Accessible Surface Area*

Overall, we find a strong correlation between non-polar first solvation shell effects and the number of solvent molecules around the solute, which itself correlates well with the solvent accessible surface area[123]. In one approach the surface area is directly used to calculate non-polar

solvation free energies, in the other it is used as a starting point. In all cases, it is important to accurately calculate this quantity, especially since it can change a lot during simulations[130]. In all methodologies, a molecule is modeled as the aggregation of spheres, and each sphere corresponds to an atom. Every sphere is associated with an extended van der Waals radius that includes the radius of the solvent molecule.

There are numerous ways to calculate the accessible area, ranging between exact and approximate, differing in the speed of calculations of areas and gradients. An important numerical method was introduced by Frank Eisenhaber et al. in the 90's[131], in which a double cubic lattice is used. The first lattice compromises the whole protein and can be used to establish neighbor-lists[132], which are necessarily used in molecular dynamics simulators to rapidly determine which atoms are close to which other atoms. Another lattice is used with extends only to the dimensions of a single sphere and uses the fact that, if dots were drawn on the surface of the sphere, only those dots could be buried which are situated in the overlap of neighboring spheres. The method is fast, but it lacks analytical derivatives which limits its usability in molecular dynamics simulations.

1.5.5.3 *Identification of Exposed Boundaries*

Most analytical solvent Accessible surfaces are methods are based on the Gauss-Bonnet Theorem. Utilizing the theorem, a Gauss-Bonnet Path has to be established. This relies on identification of the arcs that form the boundaries of the solvent accessible surface areas. Locating these arcs is computationally expensive due to the large amount of possible intersection-points, of which most of them are actually occluded by intersecting neighbors and therefore buried. A limited set of algorithms exists today to address this problem. A subset of the atoms of a molecule are totally buried, as such they have no surface area and all their intersections with other neighbors on the surface of the sphere are therefore buried as well. These atoms can be safely removed from the process of identifying the boundary of the exposed area.

Jörg Weiser, author of the linear combinations of pairwise overlaps algorithm, was very active in the field, and developed two independent methods for the removal of buried atoms. In the first[133], certain geometrical facts are exploited to determine if an atom is buried or not. The decision is based on the intersections of 3 and 4 neighbors and the occlusion of certain points on the interface.

The second method[134] uses statistical considerations to form an opinion about the buriedness of an atom. Along four tetrahedral rays k, which are shot from the center of sphere S_i into space, the neighbor density ρ_{ik} is estimated using a Gaussian kernel.

$$\rho_{ik} = \sum_{j}^{m_{ik}} \exp\left(-\alpha_i \frac{d_{ij}^2}{(r_{H_2O} + r_j)^2}\right) \tag{10}$$

In which m_{ik} is the number of neighbors of S_i and α_i a constant which depends on the atom type. d is the distance between S_i and S_j and r respective radii.

For each atom type, thresholds ρ_i^* were determined and saved in a table. If $\rho_i > \rho_i^*$ then the atom is termed buried and removed from the list of valid atoms.

Fraczkiewicz et al. proposed a very elegant algorithm to detect the exposed boundary. Their method is based on half spaces, i.e. separating hyperplanes trough the interfaces of the intersection between a sphere S_i and a neighbor sphere S_j. They exploit the fact that all intersection-points are located on intersections between these hyperplanes. First, the hyperplanes are created and orthogonal vectors from the center of sphere S_i to the hyperplane calculated. Now, the vectors are geometrically inverted, i.e. if they had a modulus of d, they now have $1/d$. The convex hull of the inverted vectors is formed. The hull is now a geometrical construct with vertices that are compromised of a subset of the inverted vertices (roughly speaking the vertices that extended the most). The hull contains no cavity, and has the advantage that a lot of the inverted points were removed, since they are internal to it. New vectors towards the faces of the elements of the hull are calculated. These vectors already point towards intersection-points.

The method is extremely elegant, but has multiple disadvantages which render it unusable for molecular dynamics. The geometric inversion assumes that no half space will ever intersect with the center of S_i or that the half space is inverted. This however is a regular case for molecular dynamics simulations or in particular simulations that rely on hydrogens. Hydrogens bonded to oxygen atoms mostly reside in the oxygen's van der Waal's radius. Their intersection always produces an inverted half-space that cannot be handled by the method. It is also assumed that there are enough independent vertices to form a three dimensional convex hull, which might not be the case if there are not many neighbors around S_i, or if the neighbors are not in a general position.

2

OBJECTIVES

The objectives of this thesis can be summarized as follow:

1. Establisment of a protocol that allows the execution of multiscale molecular dynamics simulations

2. Parametrization of a coarse-grained force-field to use within the context of the multiscale protocol

3. Definition of appropriate reaction coordinates for the study of protein dynamics biased towards protein folding

4. Development of an algorithm to perform on-the-fly functional value integrations and their derivatives on exposed boundaries to pave the road towards a semi-analytical Semi-Explicit Assembly method

3

METHODS

In this chapter, methods are presented that are used in the multiscale protocol, or which can be used alternatively. Within the protocol, we perform a free energy calculation of the difference between a conformation in the coarse-grained potential and the all-atom potential, using an integration schema.

3.1 FREE ENERGY INTEGRATION SCHEMES

Free Energy Integration is a useful tool to derive free enery differences for processes under investigation. In a biological process, reactants evolve over time into products, which causes a change in free energy of the system. This energy difference is a very important quantity and can principally be extracted by performing molecular dynamics simulations, in which the system is slowly transformed from the reactants to the products. Processes can for example involve mutations, in which the reactants correspond to the unmutated protein, and the products to the mutated. Both entities are described by different potentials (one excluding, the other including the mutation). In our multiscale protocol, the reactants describe a coarse-grained force-field and the products an all-atom force-field (and vice-versa). The scenarios are very different, but the underlying technique is compatible.

3.1.1 *Thermodynamic Integration*

In thermodynamic integration[135], the system is slowly advanced towards U_1, by incrementing λ continuously. At each λ, a molecular dynamics simulation is performed in the mixed potential $U^\lambda(\boldsymbol{x})$ until

equilibrium is regained. The average potential energy derivative with respect to λ is integrated, which directly yields ΔG.

$$\Delta G = \int_{\lambda=0}^{1} \left\langle \frac{\partial U^{\lambda}(\boldsymbol{x})}{\partial \lambda} \right\rangle_{\lambda} d\lambda \tag{11}$$

3.1.2 *Free Energy Perturbation*

Free energy perturbation[136] on the other hand accumulates averages of small Boltzmann weighted potential energy differences between successive λ values at equilibrium.

$$\Delta G = -k_B T \sum_{i=0}^{n-1} \log \left\langle \exp(-\beta (U^{\lambda_{i+1}}(\boldsymbol{x_i}) - U^{\lambda_i}(\boldsymbol{x_i}) \right\rangle_{\lambda_i} \tag{12}$$

The technique can be used in forward, backward, and "Bennet-style" forward-backward mode[70]. It is reported[69] that free energy perturbation is sensitive to the size of the λ-steps, i.e. enough overlap between successive potentials must exist in order to yield good estimates.

3.1.3 *Jarzynski Averaging*

Methods in which the overlap between successive potentials only matter in practice, but not in theory, were made available by exploitation of Jarzynski equation, in which equilibrium is no longer a necessity[40]. If the above techniques were performed in non-equilibrium, an excess in ΔG would be arise, because fast switching results in friction that incorrectly adds to the free energy. Jarzynski averaging spawns multiple evaluations of the forward (and/or backward) work:

$$W_f(\mathrm{Path}_n) = \sum_{i=0}^{n-1} U^{\lambda_{i+1}}(\boldsymbol{x_i}) - U^{\lambda_i}(\boldsymbol{x_i}) \tag{13}$$

Which are then converted to the free energy difference.

$$\Delta G = -k_B T \log \langle \exp(-\beta W_f) \rangle \tag{14}$$

It should be noted that arguably, non-equilibrium approaches are not necessarily superior to equilibrium approaches[69]. While in theory

the switching speed between different λ values should not matter, in reality it does.

3.1.4 *Crooks' relation*

Crooks' relation is a generalization of the Jarzynski equation[71]. It states that it is possible to extract the free energy difference between state A and state B from the ratio of probabilities that a work W, that was previously spent to force A into B, is released into the system[137].

$$\frac{P_F(+\beta W)}{P_R(-\beta W)} = \exp\left(\beta(W - \Delta G)\right) \tag{15}$$

Where P_F and P_R are the probabilitiy distributions with respect to the forward and backward transformation respectively.

3.2 PARAMETRIZATION TECHNIQUES

As previously mentioned, the integration is performed between a coarse-grained and an all-atom potential. Multiple methods exist to parametrize such potential, using all-atom data as a reference.

3.2.1 *Iterative Boltzmann Inversion*

Boltzmann inversion is a method to parametrize interactions based on statistical data, i.e. frequencies, which are converted into energy potentials[138]. The general idea is that if structural data on species, i.e. residues, atom-types, etc.. is available en masse, then it is possible to estimate the potential of an interaction $u(\boldsymbol{x})$ by counting how often the structures are expressing a certain value of $\boldsymbol{x}$. These collected frequencies are an intuitive measure of which values $\boldsymbol{x}$ are equilibrium values for $u(\boldsymbol{x})$ and which values should express high energies. The structures may be derived from the protein-database, or from trajectory data of reference simulations[139]. The interactions have to be compiled analytically, and their reactants laid out. The potential function $u(\boldsymbol{x})$ can now be iteratively estimated by considering the following process: An initial potential of mean force u_{PMF} is created from a radial distribution function rdf_{ref}. This function, loosely speaking, gives the probability to find a two-particle system in a certain state $\boldsymbol{x}$.

$$u_{PMF} = -k_B T \log rdf_{ref}(\boldsymbol{x}) \tag{16}$$

This initial potential is now successively refined by considering additional radial distribution functions rdf_i

$$u_{i+1} = u_i + k_B T \log \frac{rdf_i(\boldsymbol{x})}{rdf_{ref}(\boldsymbol{x})} \tag{17}$$

3.2.2 *Subtraction Method*

If the radial distribution functions are generated in dilute solutions, problems due to under-sampling might arise. The Subtraction method [140] was established to address this issue [141]. It works in three steps. First, the potential of mean force is computed for two solute-pairs in a solvent box (u^{AA}_{PMF}). Afterward, the potential is recalculated without solute-solute interactions to assess the effect of the solvent onto the potential ($u_{PMF,excl}$). Finally, both potentials are subtracted to yield the plain potential.

$$u^{CG} = u^{AA}_{PMF} - u_{PMF,excl} \tag{18}$$

3.2.3 *Force Matching*

Force matching aims to reproduce forces gained from explicit all-atom trajectory data on the coarse-grained level [142]. For every coarse-grained bead in the representation, all-atom forces are collected and matched against parametrized coarse-grained forces. The parametrization $g_0 \dots g_n$ can be obtained optimizing the following objective function [143]:

$$\chi^2 = \frac{1}{3LN} \sum_{l=1}^{L} \sum_{i=1}^{N} \left| F^{ref}_{il} - F^{p}_{il}(g_0, \dots, g_n) \right|^2 \tag{19}$$

In this case, F^{ref}_{il} is the reference, all-atom force acting on the ith atom of the lth atomic configuration. It does therefore not rely on radial distribution functions like the introduced iterative Boltzmann inversion. The actual force functions can be derived e.g. by choosing $g_0 \dots g_n$ to be spline parameters. Equation 19 then represents an optimization of splines to the forces. Potential functions can then be obtained by integration over the splines [141].

Force-matching produces force-fields that exhibit greater transferability than those parametrized by iterative Boltzmann inversion [54]. Transferability plays a role when the force-field is used in different

temperatures than it was parametrized with, which can undermine reliability in e.g. replica-exchange simulations. This is due to the fact that the potential obtained is a potential of mean force which is a free energy in phase space[54]; thus it will be sensitive to temperature and thermodynamic variations.

3.2.4 *Multiscale Coarse-Graining Method*

An advancement of standard force matching has been introduced with multiscale coarse-graining method (MS-CG)[143]. A problem of standard force matching procedures is that the fitting of the force parameters becomes rapidly intractable as the number of parameters grows[54]. The idea of MS-GC is to develop a system of overdetermined linear equations, that is subsequently solved, and from which force parameters are extracted. Generally, this can be achieved if the the force-field depends linearly on the fitting parameters. If this is the case, the derivative of the force will be constant.

$$\frac{\partial \boldsymbol{F}_{il}^{p}}{\partial \boldsymbol{g}} = \text{const} \tag{20}$$

The system can then be solved by setting

$$\left(\frac{\partial X^2}{\partial g_j}\right)_{j=1\ldots N} = 0 \tag{21}$$

which can be denoted with matrices:

$$\left\| \frac{\partial \boldsymbol{F}_{il}^{p}}{\partial g_j} \right\|^{T} \boldsymbol{F}_{il}^{p} = \left\| \frac{\partial \boldsymbol{F}_{il}^{p}}{\partial g_j} \right\| \boldsymbol{F}_{il}^{ref}, i = 1, \ldots, N \; l = 1, \ldots, L \tag{22}$$

which can be subsequently simplified to an overdetermined system of linear equations:

$$\boldsymbol{F}_{il}^{p}(g_0, \ldots, g_N) = \boldsymbol{F}_{il}^{ref} \tag{23}$$

3.3 COLLECTIVE VARIABLES

Within the protocol, we calculate free-energy differences for states using free energy surfaces. The surfaces are low dimensional projections of an optimally low dimensional manifold residing in a high dimensional space. Because the shape of the manifold is generally not

known, is is usually difficult to find collective variables which perfectly project conformations into the low dimenional space. We present here a collection of general collective coordinates, that (except for principle component analysis) do not require post-processing the data.

3.3.1 *Root Mean Square Deviation*

The Root Mean Square Deviation (RMSD) is a very old and widely used metric[89]. It is calculated by directly comparing a selection of the atoms of a molecule with those of a reference molecule in a possibly different conformation, in the case of proteins usually the alpha carbons or heavy atoms. The metric computes the squared spatial deviation between the same atoms in the molecule and the reference. Any such deviation depends on the spatial displacement and orientation between the structures. In the context of RMSD, the deviation is meant to be minimal, as such the two structures need to be aligned first by translation and rotation.

$$\mathrm{RMSD}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} ||\boldsymbol{x}_i - \boldsymbol{y}_i||^2} \tag{24}$$

RMSD has some problematic disadvantages. Due to the quadratic nature of the calculation, if part of the structures perfectly match, but just a small part does not, this small part would dominate the deviations and result in bad distances. Furthermore, translation and rotation of the molecule to maximally align with the reference is an optimization task which ultimately might end in a local minimum. Several methods have been proposed in the past to compensate. MaxSub[144] and TM-score[145] are two examples in which only parts are compared that are already well aligned. SABIC[146], or lcpfold, (introduced in chapter 4.2) are methods based on internal coordinates, in which the necessesity for rotation and translation is removed.

3.3.2 *Radius of Gyration*

Radius of Gyration[90] is an intrinsic measurement of how compact the conformation of a protein is. A small value would indicate compactness, and a large value (dependent on the size of the protein) openness. The measure is performed by calculating the protein's center, which can be e.g. the center of mass, or some other form of atomistic average.

The distance of each atom to this center is computed, squared, added and subsequently its root taken out.

$$R_{gyr} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(x_k - x_{center})} \tag{25}$$

In which N is the number of atoms.

3.3.3 *Native Contacts*

Internal contacts of protein residues provide rich information about tertiary connectivities, folds, and secondary structure assemblies. Usually, residues are described as being in contact if their euclidean distance is between 2 and 9 Å[147]. The distance can be measured from the geometric centers or centers of mass, or between the closest atoms belonging to both residues. Native contacts describe residues that are in contact in the native state. These types of contacts are highly correlated with P_{fold} (r>0.90)[148] and therefore of much interest in describing folding processes, because - with knowledge of the conformations in the native state - they are very simple to calculate in contrast to P_{fold}.

Is the native state not known, we can still formulate an intrinsic similarity metric using internal contacts. For this purpose, a contact map is created, which, for each residue pair of a protein, holds binary information whether the residues are in contact or not[147]. Subsequently, maps of different conformations are compared using any form of comparison, e.g. the number of exact matches.

3.3.4 *Universal Similarity Metric*

The universal similarity metric (USM) was first introduced by Li et al in 2001[149]. It is a distance metric based on Kolmogorov complexity[150], which is a measure of the amount of information in given data. To be precise, it is a measure for the shortest set of instructions I for a Universal Turing Machine T, that can output a certain string S. The Universal Turing Machine was introduced by Alan Turing in 1936 and is arguably the minimal model of a computer that can process arbitrarily complex data. Every modern computer, except for quantum computers, is not more powerful than a Turing Machine - just faster. The USM is able to approximate every other similarity metric, including those that are yet to be invented[151]. The type of Turing Machine

that is internally used does not change the similarity distance more than an additive constant. The metric is defined as:

$$d(o_1, o_2) = \frac{\max\left\{K(o_1|o_2^*), K(o_2|o_1^*)\right\}}{\max\{K(o_1), K(o_2)\}} \tag{26}$$

, where o_1 and o_2 are the objects that are compared, and o_1^* and o_2^* the shortest set of instructions for o_1 and o_2. K gives the Kolgomorov distance

$$K(o_1|o_2) = \min\{|I|\} \tag{27}$$

with $T(P, o_2) = o_1$ and I being sets of instructions as already mentioned.

The above metric is only upper semi-computable[151], which means that it can only be approximated by overestimation. Surprisingly, this is very easy to achieve. First, contact maps of the reference and template proteins are created. Second, they are written as a binary string, and this string compressed using a standard tool like ZIP. $K(o)$ is then the size in bytes of that string.

3.3.5 *Principal Component Analysis*

P_{fold} is said to be the ideal reaction coordinate for protein folding. It is the reaction coordinate along which the system evolves the slowest[88]. This in fact relates to the first eigenvector of a principal component analysis (PCA) performed on trajectory data. This implies that from a pre-established trajectory we can extract reaction coordinates that in the optimal case relate to the perfect reaction coordinate, and in all other cases to reaction coordinates that still discretize the data optimally. Such result is possible, because simulation data is of very high dimensionality, but the trajectories themselves lie on a much lower dimensional manifold, which can be expressed just with very few eigenvectors[86]. In the case that the low dimensional manifold is nonlinear in nature, where PCA would result in inaccurate results because of its linear nature, non-linear PCA is available[152], as well as PCA on internal coordinates[153].

3.4 SOLVATION

The all-atom and coarse-grained force-fields must be compatible for the free energy integrations to converge. This requires a similar handling of the solvent as well. Since it would be impractical for the

coarse-grained force-field to utilize explicit solvent (even though such force-fields exists[53]), the coarse-grained force-field is assumed to contain implicit solvent corrections, and the all-atom force-field an implicit counterpart. For polar solvation, there are numerous possible methods which can be utilized:

3.4.1 *Polar Solvation*

Polar solvation models can be cataloged into a number of different categories[123]. Easily, we can make a distinction between polarizable and non-polarizable methods. For polarizable methods, the reaction field is critical. It is created by the dipole moment of the solute, which polarizes the solvent, and induces a dipole moment, which in turn polarizes the solute[154]. Methods utilizing the reaction field have been developed[155;156]. These methods assume a cutoff, after which the force due to the reaction field approaches zero[96].

Many polar solvation methods have been modeled according to the Poisson equation[112–114].

$$\nabla\varepsilon(\boldsymbol{r})\nabla\phi(\boldsymbol{r}) + 4\pi\rho(\boldsymbol{r}) = 0 \tag{28}$$

In which ∇ is the Laplace operator, that creates a vector of squared partial derivatives, $\boldsymbol{r}$ is the vector of atomic coordinates, $\varepsilon(r)$ the dielectric constant, $\phi(\boldsymbol{r})$ the potential and $\rho(\boldsymbol{r})$ the charge density. The goal is to evaluate equation 28 to calculate the charge density, given its distribution.

Calculation of this quantity is very cumbersome and time intensive. The Ewald method splits up calculations of the electrostatic energies into two components. A rapidly varying term for short distances and a slowly varying term for long distances. The rapidly varying term is evaluated directly, and the slowly varying term in fourier space[115;157], which accelerates convergence. Particle Mesh Ewald uses the Ewald summation, but operates on a mesh. For each node on the mesh the Poisson equation is evaluated.

Another derivative of the Poisson equation is the Generalized Born method, which is based on the Born equation[116;117] for the transfer of a single ion from gas phase into solvent.

$$\Delta G = -\frac{q^2}{2r}\left(1 - \frac{1}{\epsilon}\right) \tag{29}$$

Where q is the charge, r the ionic radius and ϵ the dielectric constant of the solvent. The Born equation can be extended into the Generalized

Born equation for the approximation of the charge density of multiple distributed point charges[117].

$$G_{pol} = -166\left(1 - \frac{1}{\varepsilon}\right)\sum_{i,j}\frac{q_i q_j}{f_{GB}} \tag{30}$$

This generalized formula is the basis of a wide branch of solvation methods that aim to approximate the Poisson equation more and more accurately[51;118;119]. Their main difference is the calculation of the Born Radii in the smoothing function f_{GB}.

$$f_{GB} = \sqrt{r_{ij}^2 + R_i R_j \exp\left(\frac{-r_{ij}^2}{4R_i R_j}\right)} \tag{31}$$

In which r_{ij} are pairwise distances and R_i and R_j the aforementioned Born Radii. Broadly speaking, these radii correspond to the buriedness of the atom in the solute[51]. The Poisson Equation can be solved for perfect Born Radii, which inserted into the Generalized Born equation will yield polar solvation free energies of equal accuracy to the Poisson Equation. Since this would be more expensive than just solving the Poisson-Equation itself, approximations to the perfect radii must be seeked[158]. Approximations to the radii usually utilize integration of excluded volume[119], to account for the buriedness of an atom. To complicate things further, Born Radii methods rely on some form of intrinsic radius, which is constant. There is significant spread amongst the radii, which can change the outcome of the calculations[159–161].

3.4.2 *Non-polar Solvation*

In terms of non-polar solvation, γA approaches are mostly utilized. They require calculation of the solvent accessible surface area, for which a continuum of methods is available. We will present here two widely used method based on pairwise overlaps, later we move on to the basis of analytical methods, the Gauss-Bonnet Theorem, and ultimately we conclude with Semi-Explicit Assembly.

3.4.2.1 *Pairwise overlaps*

Fast methods with derivatives are usually of statistical and probabilistic origin. Shoshana Wodak et al. derived a simple method based on

single overlaps[162]. For a single overlap, the buried area is a simple analytical expression, in contrast to multiple overlaps:

$$A_{ij} = \pi r_i (r_i + r_j - d_{ij}) \left(1 + \frac{r_j - r_i}{d_{ij}}\right) \tag{32}$$

In which r_i and r_j are the radii of sphere S_i and S_j and d_{ij} their distance. They treat the (normalized) buried area as a probability that a point on the surface is inside the buried area, thus, with the inverse, they calculate the probability that a point is outside all buried areas. The method principally does relatively well for the calculation of the whole molecular area due to a cancellation of individual atomic errors. If atomic areas matter, the method is unfortuatnely insufficient.

A somewhat similar approach was undertaken by Jörg Weiser et al. which also concentrated their calculations on single overlaps between a sphere and its neighbors, as well as neighbor spheres with other neighbor spheres[163]. Using the same equation 32, they calculate the buried area A_i of sphere S_i as:

$$A_i = P_1 S_1 + P_2 \sum_{j \in N(i)} A_i j + P_3 \sum_{\substack{j,k \in N(i) \\ k \in N(j) \\ k \neq j}} A_{jk} + P_4 \sum_{j \in N(i)} A_{ij} \left(\sum_{\substack{j,k \in N(i) \\ k \in N(j) \\ k \neq j}} A_{jk} \right) \tag{33}$$

Where $N()$ are neighbor-lists, and P_1 tp P_4 parameters for each atom type, obtained by multiple linear regression. The method is doing very well; calculated areas correlate to true areas with r~0.9 and derivatives r~0.6. However, problems arise when molecules are treated that exhibit untrained geometries. Also, hydrogens are not supported.

Surface areas do not necessarily be approximated. They can be calculated exactly thanks to the Gauss-Bonnet theorem, which will be explicitly treated in the next section. A fair amount of methods have been developed based on it[164–169]. The drawback of these methods however is that they need an exact description of the boundary of the solvent accessible surface area, and with it the precise intersection-points on this boundary, where two neighbor spheres intersect on the surface of a third sphere. Finding these points is cumbersome and computationally expensive and will be discussed after presenting the Gauss-Bonnet theorem.

3.4.3 *The Gauss-Bonnet Theorem*

We follow here the derivation of Manfredo Do Carmo[170] in a simplified form, focusing on unit spheres. First, we consider a triangle on a unit sphere. Unless in two dimensional space, the sum of the interior angles ϕ_i of a triangle T will exceed π. The surplus is called excess. Gauss showed in an early paper[171] that the excess is equal to the integral of the Gaussian curvature K over the surface T.

$$\sum_{i=1}^{3} \phi_i - \pi = \iint_T K d\sigma \tag{34}$$

Where σ is an infinitesimal surface patch of T. The Gaussian Curvature is the product of the principal curvatures, which are a measure of the minimal and maximal bends of a regular surface at each point[172]. In spheres, this curvature is 1 and the above equation reduces to the known formula for the calculation of the area A of a spherical triangle.

$$\sum_{i=1}^{3} \phi_i - \pi = A \tag{35}$$

This formula, as well as all others that will be presented require a regular surface, which for the Local Gauss Bonnet Theorem also needs to be simple. A regular surface is compact, i.e. has a triangulation with a finite number of triangles[173], and it has a boundary which consists of closed piecewise regular curves which do not intersect[170]. A simple surface does not intersect with itself on any point.

The solvent accessible surface area is such a regular surface. Its boundary consists of spherical arcs which connect intersection-points. The intersection-points are points on the surface where three spheres meet: The sphere which surface is to be calculated and two neighbor spheres. We consider now a complete tessellation of the surface area with triangles, where each triangle has two sides that are also boundary arcs. This is always possible for regular surfaces[170]. The Local Gauss Bonnet Theorem now states:

$$\sum_{i=0}^{k} \int_{s_i}^{s_{i+1}} k_g(s) ds + \iint_R K d\sigma + \sum_{i=0}^{k} \theta_i = 2\pi \tag{36}$$

In which s is a parametrization for an arc connecting intersection-points p_i and p_i, i.e. one "exterior side" of a triangle of the tessellation,

k_g the geodesic curvature of the arcs and θ the external angles of the intersection-points. $\iint_R K d\sigma$ is again the area - now for the full surface, instead of just a single triangle. The Gaussian Curvature k_g is a measure of the arc length of each boundary segment[164].

The local Gauss bonnet theorem can be extended to allow for holes in the surface R. The new theorem is called Global Gauss Bonnet Theorem and sets into relation the geometry of surfaces and their topology. It uses the Euler-Poincaré Characteristic χ, which is a determinant of the topology of a tessellation, dependent on the number of distinct faces F, edges E and vertices V.

$$F - E + V = \chi \tag{37}$$

The Global Gauss Bonnet can now be stated as:

$$\sum_{i=0}^{n} \int_{C_i} k_g(s) ds + \iint_R K d\sigma + \sum_{i=0}^{k} \theta_i = 2\pi\chi(R) \tag{38}$$

Where C_i is a curve of the respective arc. This can be rewritten[166] to express the area in terms of the tangential angles at each intersection-point Ω_i, the arc lengths ϕ_i, opening angles of the intersection circles Θ_i, and the radius of the sphere r.

$$A = r_i^2 \left(2\pi(2 - \chi) + \sum_{i=0}^{n} (\Omega_i + \Phi_i \cos\Theta_i) \right) \tag{39}$$

3.4.4 *Semi-Explicit Assembly*

The method starts from a standard solvent accessible surface area with a solvent radius of 1.4. From the center of each atom that has a surface to the solvent, rays are shot in sufficient resolution. Along those rays, the Lennard Jones field, emanating from the atom and its neighbors is probed. Van der Waals interactions between two atoms S_i and S_j are usually approximated by the Lennard Jones potential:

$$LJ(d_{ij}, \varepsilon_{ij}, \sigma_{ij}) = 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{d_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{d_{ij}} \right)^{6} \right] \tag{40}$$

In which d_{ij} is the distance between the atoms, ε_{ij} the geometric mean of the respective well-depths and σ_{ij} the arithmetic mean of the equilibrium distances. The method locates the minima of the Lennard

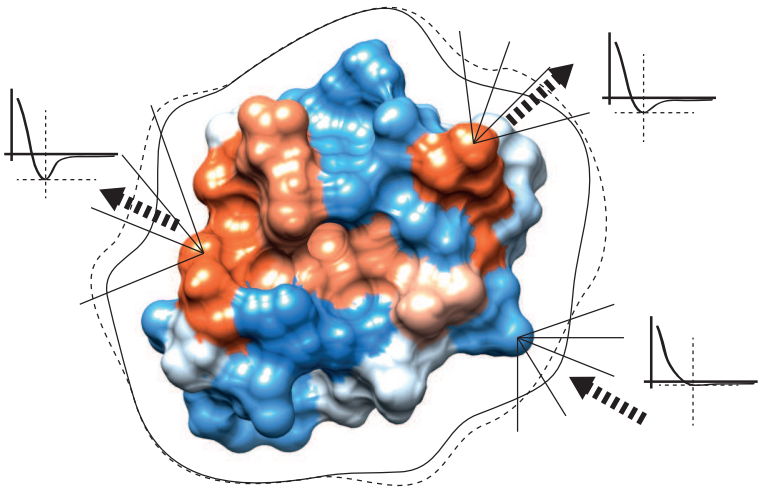


Figure 2: Representation of a molecule for which the Semi-Explicit non-polar solvation free energy is calculated. From the center of each atom, rays are drawn towards the solvent accesible surface area. Along the rays, the manifold, representing the minimum of the Lennard-Jones field, is probed. The solvent accessible surface area is depicted with a filled and the manifold with a dashed line. Dependent on the strength of the Lennard Jones field, the averaged effective Lennard Jones parameter will result in a strong potential, which pushes the manifold away from the exposed area, or in a weak potential, which will pull the manifold closer to the molecule. The former is expected in crevices or in areas where lots of neighbors back up the potential, the latter in atomistic outliers, in which the neighbor density is low.

Jones field; a point where water molecules experience the most non-polar attraction, and therefore are most likely to reside - disregarding other forces. For all rays emanating from one atom, a new Lennard Jones potential is created by averaging all contributions. These new effective parameter σ_{ef} and ε_{ef} are used to look up solvation free energy values ΔG_i from a precomputed table. The table was created by solvation Lennard Jones sphere (i.e. with differing σ and ϵ parameters) from gas-phase into water and calculation of their respective solvation free energies. This yields the final semi-explicit assembly formula:

$$\Delta G_{np} = pV + \sum_i f_i \Delta G_i \tag{41}$$

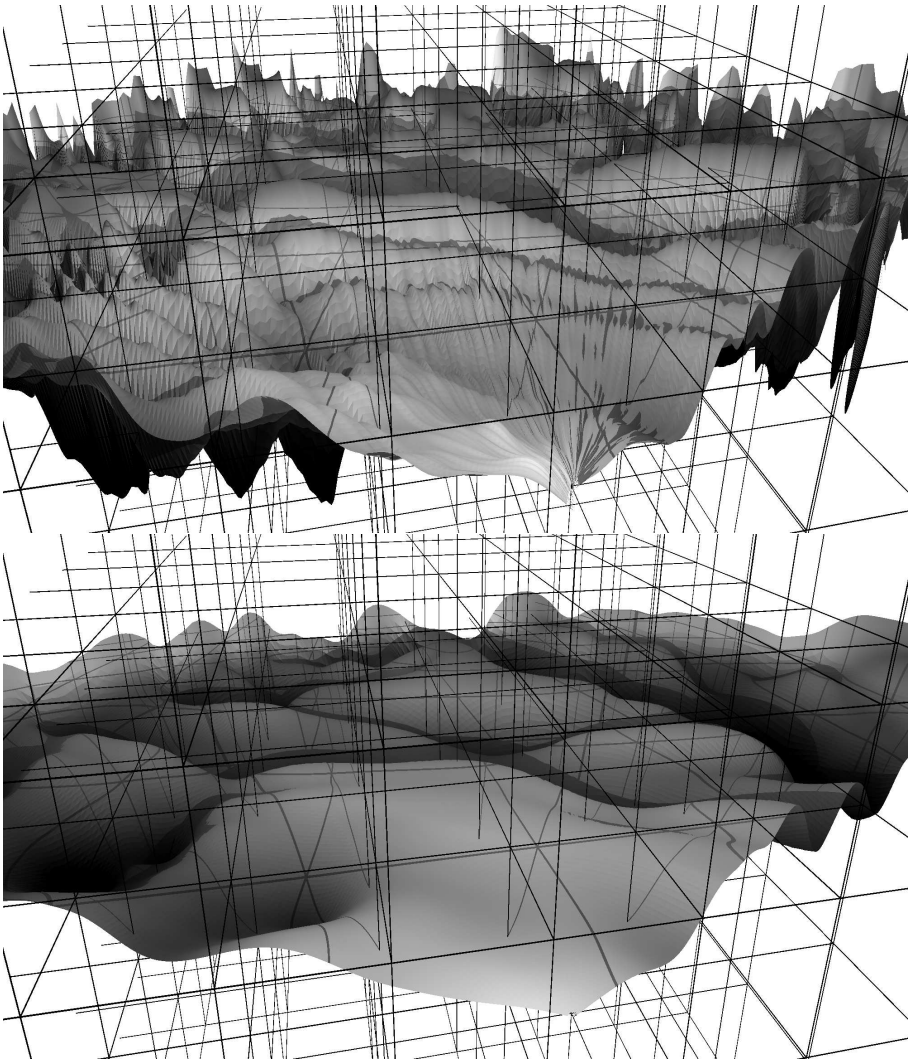
Where f_i is the fraction of the solvent accessible surface area compared to total surface area, and pV is the cavity creation cost.

4

RESULTS

The results are divided into 2 parts. The first part deals with the multiscale protocol itself, a new collective variable based on contact maps, and a parametrization of a new coarse-grained force-field. The second part is about a new surface area integration algorithm, and an optimization, both designed to be utilized within the Semi-Explicit Assembly methodology, but can also be used stand alone.

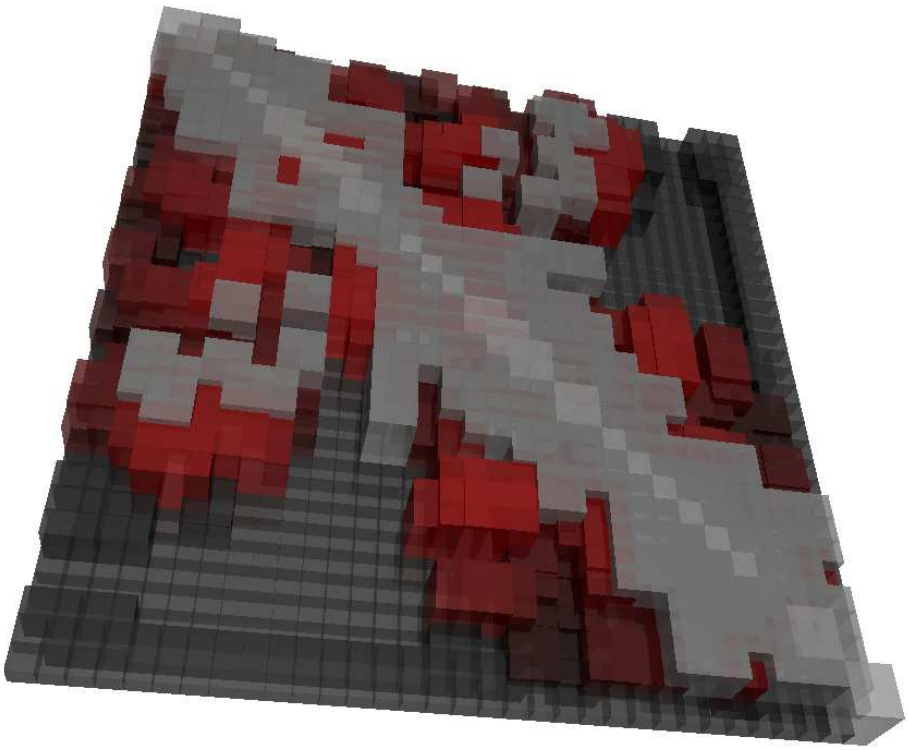
In the first part, we start by presenting the multiscale protocol, to give the reader in-depth knowledge about its theory and implementation. We then move towards an example in which we test the protocol on two coarse-grained force-fields, and present the development of one of these force-fields. In between, the collective variable is introduced.



Artistic representation of different, yet synchronized multiscalar free energy surfaces

4.1 MULTISCALE MOLECULAR DYNAMICS OF PROTEIN AGGREGATION

Nils J. D. Drechsel, César L. ÁÀvila, Raúl Alcántara, and Jordi Villà-Freixa.
"Multiscale molecular dynamics of protein aggregation."
Current Protein and Peptide Science 12, no. 3 (2011): 221-234.



Artistic representation of a contact map difference with matching contacts (grey) in which some mismatches (black and displaced downwards) were found to be correctable (red)

4.2 LOCAL CONTACT P_{fold} SIMILARITY

Nils J. D. Drechsel[1] Jordi Villà–Freixa[2]

[1] Universitat Pompeu Fabra, Computational Biochemistry and Biophysics Laboratory, Research Unit on Biomedical Informatics, IMIM Hospital del Mar-Universitat and Universitat Pompeu Fabra, C/Doctor Aiguader, 88, 08003 Barcelona, Catalunya, Spain

[2] Escola Politècnica Superior, Universitat de Vic, C/ de la Laura, 13, 08500 Vic, Catalunya, Spain

The method presented herein yields a new collective variable, which can be used as an order parameter or reaction coordinate, depending on the quantity of interest. It extends the idea of contact-map comparison between a reference and a template, by incorporating hydrophobic-zipper theory, calculating the propensity of a mismatch in the contact-maps to correct itself to a native contact through only a local conformational search. A Markov Random Field is created from the difference of contact maps and used to propulse marginal probabilities which express the belief that certain torsions between the involved amino-acid-pairs can fold from the template into the reference. The torsional probabilities were collected prior to this through a series of Monte Carlo Markov Chain samplings of the torsional potential of various main-chain and side-chain torsions, for different force-fields and temperatures and saved into a readily accessible database.

4.2.1 *Introduction*

The search for better collective variables, i.e. reaction coordinates and order parameters, is a still ongoing quest in many disciplines, particularly protein folding[174]. Generally speaking, the quality of a collective variable highly depends on the quantity of interest. They can be used for post-processing trajectory data, or for guiding simulations during sampling.

In the domain of collective variables, many methods are based on isocommittor surfaces[88]. The isocommittor is a hyperplane through phase space in which the commitment functions ϕ_A and ϕ_B reach equality. Those functions express the probability that a trajectory, started from this point, reaches state A before state B and vice-versa. This links directly to transition path sampling techniques[175], in which a Markovian random walk is started from state A, while the Metropolis condition, i.e. the test if a move should be accepted or rejected, is biased towards state B. Once transition states and transition paths have been identified, good reaction coordinates can be defined along these interconnecting pathways.

Perfect reaction coordinates can be extracted via post-processing of trajectory data, in terms of eigenvector decompositions by principle component analysis[84–86]. The first two eigenvectors of such decomposition are orthogonal by definition and represent the vectors with which the surface can be explained the most, i.e. the least detail is lost and resolution is largest. This however is only true if the low dimensional manifold which is extracted by the decomposition can be described linearly. If not, the analysis will result in poor coordinates and nonlinear principle component analysis should be used instead[152].

Often stated, the best coordinate for protein folding is p_{fold}, the probability of folding before unfolding. With such a coordinate, the important states like the product state $p_{fold} = 0$, reactant state $p_{fold} = 1$ and transition state $p_{fold} = 0.5$ are easy to identify. p_{fold} correlates very well to the fraction of native contacts (r>0.90)[148]. Seeking to approximate p_{fold}, it is only reasonable to start with this fraction, calculated through a comparison of contact maps[147].

In this article, we extend the idea of a simple contact map and calculate approximately the propensity of two conformations to fold into the other, using only local conformational searches, supported by native contacts that have already been formed. We follow the rational of the hydrophobic zipper[176]. This hypothesis is based on the assump-

tion that proteins assume their native tertiary structure, by first establishing tentative local contacts, then growing these local structures through zipping, and later building tertiary structure and contacts through coalescence[177]. The idea is driven by the assumption that when a native contact a is formed, a neighboring native contact b is much more likely to form as well, with a very local conformation search, in contrast to an exhaustive global search if cooperativity by a was not given. In other terms, the probability of assuming native contact $p(b)$ is higher given a than without.

$$p(b|a) > p(b) \tag{42}$$

The now presented method starts from a comparison of contact maps, and approximates the propensity of a mismatch in the maps to correct itself into a native contact, if other nearby native contacts have already been formed. The method needs two reference structures, optimally the native and fully extended conformations for comparison.

The goal of this collective variable is the linearization of the folding-space, and separation of both product and reactants, so that it can be utilized as a reaction coordinate. Both is achieved by performing the same analysis on the native and fully extended conformations and combining both information into a single coordinate.

4.2.2 *Method*

The method creates two contact maps, one from a reference native structure and the other from a template structure exhibiting an arbitrary conformation. We seek to calculate the propensity, or the ability of the template to fold into the native structure by only employing local conformational searches. That way, we cluster various conformations which would easily adopt the reference structure in a molecular dynamics simulations simply through minimization to local potential energy minima.

In a first step, contact maps CMAP_0 and CMAP_1 for structures S_α and S_β are created respectively. Any method can be used to do so, e.g. a crisp contact map[147] with a contact threshold between 6 Å and 12 Å. This means, two residues are considered to be in contact if their beta carbons (or alpha carbons in the case of glycine) are not further apart than the threshold. The contact maps are "subtracted" to identify contacts which differ between S_α and S_β. A difference map DMAP is the result of this operation. It contains values 0, 1 and -1 indicating

a mismatch, a contact match, and a no-contact match respectively. We are now interested in the ability of the mismatches to fold to their correct contacts.

In a second step, a Random Markov Field (RMF) is created from the difference map [178]. Random Markov Fields are related to Bayesian Networks. The latter asserts causation and therefore exhibits a tree-like shape. The former does not, and specifically allow loops. As such, belief propagation between the nods of the network or field differ between the two models.

First, we define labels $L = \{\text{contact}, \text{no-contact}\}$ to express the ability of a residue pair to be able to create a native contact or not. Second, we define the following unary and binary energy functions:

$$U(\text{node}, \text{label}) = \begin{cases} -\log\left(p_{\text{ltfold}}(\text{node})\right) & \text{label} = 0 \\ -\log\left(1 - p_{\text{ltfold}}(\text{node})\right) & \text{label} = 1 \end{cases} \tag{43}$$

$$B(\text{label}_0, \text{label}_1) = \begin{cases} 1 & \text{label}_0 = \text{label}_1 \\ 0 & \text{else} \end{cases} \tag{44}$$

The unary function $U(\text{node}, \text{label})$ defines the energy needed to change a mismatch into a native contact. It uses the probability p_{ltfold} which expresses the ability of a residue to adapt native torsions. This probability will be defined later on. The binary energy defines the cost for two adjacent contacts to be different from one another. In the used form it will encourage the formation of a smooth field.

Propensities for mismatches to convert to native forms are calculated iteratively by sending messages between nodes. During initialization of the field, messages with content p_{ltfold} are sent from matching nodes to mismatching nodes. In subsequent iterations, only messages between mismatches are exchanged. The messages between mismatches are calculated according to the following formula:

$$m_{ij,st}(l) = \sum_{p \in L} \left[e^{-B(L(p),L(l))} e^{-U(L(p))} \prod_{k=N(ij)\setminus st} m_{k,ij}(L(p)) \right] \tag{45}$$

In which $N(ij)\setminus st$ is the set of adjacent nods without node st, i.e. we are not sending information originating from st back to it. Such a message contains the belief of node ij that node st should have either label defined by L. The algorithm successively visits every node and sends

messages to nearby nodes, until the believes converge. It has to be noted that for belief propagation on loopy networks, the system is not guaranteed to converge and may oscillate between states[179]. However, in our tests with this method we never experienced such behavior.

The total score ρ can now be written as:

$$\rho = \frac{1}{N} \sum_{ij} m_{ij,ij} \tag{46}$$

Which represents a message from node ij to itself, using N as the total number of nodes.

An important part of the calculation of the belief is the probability p_{ltfold}, which represents the ability of an arbitrary template torsion $T_{\beta i}$ to assume the same angular value as a reference torsion $T_{\alpha i}$ using only a local conformational search, disregarding any entropical effects or other conformational barriers. Given two structures S_α and S_β, torsional data θ is collected from both the reference S_α and the template structure S_β.

For each torsion T_i we calculate $\theta_{\alpha\, i}$ and $\theta_{\beta\, i}$. We will denote the difference between the two quantities with θ_i. The quantity in which we are interested can now be expressed as:

$$p(f \mid \boldsymbol{\theta}) \tag{47}$$

With f being the event of folding, $\theta = (\theta_0 \dots \theta_N)$, and N the number of torsions that have been collected from S_α and S_β. That is, we are seeking the probability that the whole template will fold into the reference, by only employing local conformational searches. For a single torsion we get:

$$p(f \mid \theta_i) \tag{48}$$

which is the probability that with a local conformational search, the torsion T_i of S_β will fold into S_α when they vary with torsional difference θ_i. This probability density will be calculated directly by performing a series of Markov Chain Monte Carlo samplings. The whole range of degrees is properly discretized.

For an arbitrary torsion T_i, starting from an initial position at θ_{start}, the system performs the metropolis algorithm until it either reaches θ_i or $\theta_i - \pi$, indicating the successful folding or unfolding event respectively. The frequency of the former and the latter is recorded, normalized and used directly as $p(f \mid \theta_i)$. In each step of the sampling, we

draw from a uniform distribution to decide whether to move into the folding direction or the unfolding direction. Both local folding events are supposed to be equally likely, the probability of going into the folding direction is 0.5. The move is accepted if the following criterion is met:

$$u \leqslant min(r, 1) \tag{49}$$

with u being drawn from a uniform distribution and

$$r = \frac{p(\theta^* \mid y)}{p(\theta^{-1} \mid y)} \tag{50}$$

with θ^* corresponding to the next parameter in the chosen direction and $p(\theta^{-1} \mid y)$ to the last parameter in the chain. The probability $p(\theta \mid y)$ is derived from an angular and torsional potential $U(\theta)$ following a Boltzmann distribution.

$$p(\theta \mid y) = Z^{-1} exp\left[-\beta U(\theta)\right] \tag{51}$$

with

$$Z = \int exp\left[-\beta U(\theta)\right] \, d\theta \tag{52}$$

The method has similarities with dihedral-RMSD. The same source data is used to derive a measurement, but dihedral-RMSD punishes large torsional differences, while this method does not.

4.2.3 *Results and Discussion*

In bad collective coordinates, the transition state does not collapse into the coordinate of maximum free energy[180]. In figure 3, which follows an example from reference[180], we see on the left side a collective variable q. All states, A, B and the transition state TS are well separated. The coordinate of maximal free energy q^* collapses into the coordinate for the transition state q_{TS}. On the right side, we see that this is not the case anymore. Without information from coordinate q_1, the states are not properly separable. The lcpfold technique asymmetrically shifts coordinates closer to the native state - with a stronger effect on coordinates which are already close to this state and a weaker

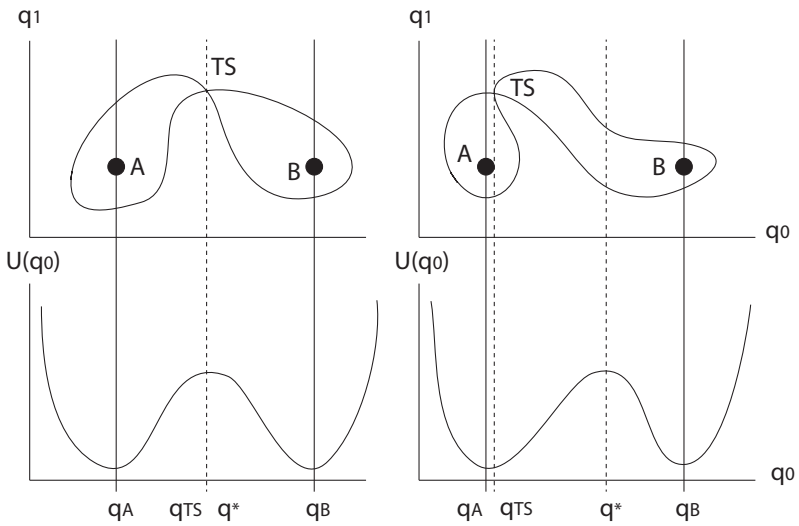


Figure 3: On the left, collective variables that also are good reaction coordinates, i.e. states A, B and TS are well separated and q_{TS} coincides with q^*. On the right, collective variables that represent bad reaction coordinates. States A and TS are not separable.

effect on coordinates that are afar. We will consider now a second reference structure in the complete extended conformation. If we let this structure undergo the same analysis, we will see the same asymmetric effect. The rational of this extended analysis is to probe the stability of each conformation, and its "ability" to lose its already formed contacts. The shift can be further emphasized by exponentiation as shown in equation 53, in which we also inverted the score, such that a perfect match has score 0 and the worst match a score of 1.

$$\rho^* = (1 - \rho)^{\alpha} \tag{53}$$

In which alpha is a positive value greater than 1. Figure 4 gives a visual comparison of the two shifting processes for an example created by trajectories from the chicken villin headpiece. There, we display the ratio between ρ^* and contact-map score. ρ^* will always be smaller than the contact-map score. The greater the difference between the two, the more the shift will approach zero. The less the difference, the more it will approach 1.

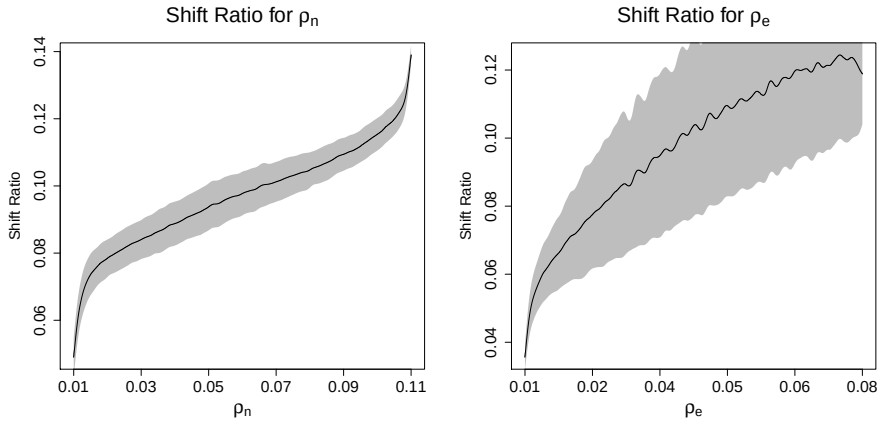


Figure 4: On the left, an example of the shift ratios for the native state, and on the right for the extended state

The new collective variable, lcpfold is using both analysis for comparison to the native ρ_n^* and to the extended ρ_e^* conformations. Optimally, ρ_n^* and ρ_e^* are negatively correlated. A low score in one should imply a high score in the other. We project both values into a one dimensional score using their ratios, making sure the whole range $(0 \dots 1)$ is properly utilized:

$$\text{lcpfold} = \begin{cases} \frac{1}{2}\frac{\rho_n^*}{\rho_e^*} & \rho_n^* \leqslant \rho_e^* \\ 1 - \frac{1}{2}\frac{\rho_e^*}{\rho_n^*} & \text{else} \end{cases} \tag{54}$$

The projection is visualized in figure 5. Straight lines emerging from the center of the coordinate system represent points on the energy surface that are collapsed into the same coordinate. Through the asymetrical shift, the transition state will be closer to the line representing a ratio of 0.5, while basins A and B will move away from it. The goal of this collective variable is to improve linearization of the folding pathway and separation of both product and reactant states. We are specifically not interested in improving $P(TP|r)$, the posterior transition path distribution, which has been proposed earlier[87] as an optimization goal, and for which expensive transition path samplings would have to be performed. A good reaction coordinate, or order parameter, is able to distinguish well between the product and reactant state[175]. Rephrased, we can say that a good reaction coordinate minimizes the uncertainty of a coordinate r being in neither the reactant

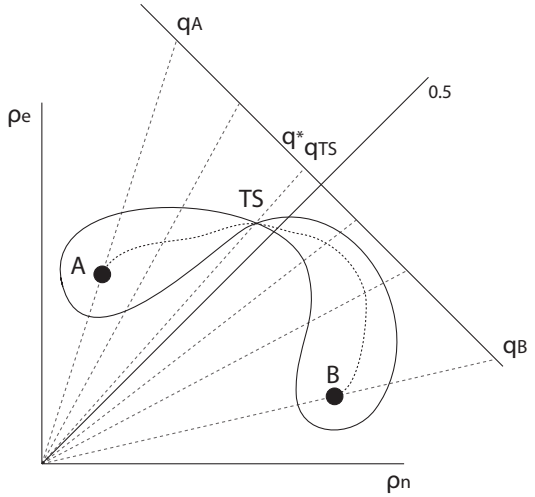


Figure 5: Projection of conformations into lcpfold coordinates, utilizing the shift and negative correlation of the sub-coordinates ρ_n and ρ_e

nor the product state. This follows from the so called commitment probabilities[181], where $\phi_B(x)$ is the probability to move into state B before state A and ϕ_A the probability to move into state A before B, which is $1 - \phi_B(x)$. So to speak, at $\phi_A = \phi_B = 0.5$ uncertainty is largest. When $\phi_B = 0$ or $\phi_B = 1$, uncertainty is lowest. Figuratively, two distributions that are infinitely far apart are easily separable, it is always clear if the system is in the reactants or the products state. On the other hand, if the distributions overlap to a great part, it is impossible to distinguish the states properly.

To visualize the method, in a first step we take the histogram of folding simulations of the 35 residue NLE-NLE variant of the Villin Headpiece. In this variant, two lysines are replaced by norleucines. Trajectories were obtained from reference[182]. There, 9 folding attempts, each with hundreds of continuous trajectories, have been performed from different starting structures. We collected samples from two folding attempts and projected them onto a set of reaction coordinates (the method proposed herein, RMSD, fraction of native contacts) to obtain the equilibrium distributions.

The distributions in all reaction coordinates showed two peaks associated with high probabilities, which we assigned labels disordered denatured, and native, corresponding to the respective states. Both basins are approximately normally distributed, separated by a low-probability barrier which consists of a mixture of the two distributions. We will first extract the distributions $P(r|RS)$ and $P(r|PS)$ for being in the reactant and product state respectively. The partitioning is performed with an Expectation-Maximization algorithm[183].

Once the distributions have been extracted, they are compared using the Kullback-Leibler divergence $KL(f\|g)$. The divergence between two distributions f and g is a measure of the information loss when g is used to explain f instead of f to explain f. In the case of two distributions that correspond to products and reactants, it is a measure on how separable they are.

$$KL(f\|g) = \int f(r) \log \left[\frac{f(r)}{g(r)}\right] dr \tag{55}$$

Because $P(r|RS)$ and $P(r|PS)$ are normal distributions $N(\mu_1, \nu_1)$ and $N(\mu_2, \nu_2)$, the above integral can be formulated in closed form[184]:

$$KL(f\|g) = \frac{1}{2}\left(\log\left[\frac{\sqrt{\nu_2}}{\sqrt{\nu_1}}\right] + \frac{\nu_1}{\nu_2} + \frac{(\mu_1-\mu_2)^2}{\nu_2} - 1\right) \tag{56}$$

We use the symmetrical version of the divergence:

$$KL^s(f\|g) = KL(f\|g) + KL(g\|f) \tag{57}$$

The divergence for RMSD is 39.64 for contact maps 41.88 and lcpfold 226.52, indicating that lcpfold is doing a better job in separating the two states then RMSD and contact maps, whereas contact maps has a slight advantage over RMSD.

We can use lcpfold as a reaction coordinate to visualize the folding of villin from a high lcpfold value to a low value, using radius of gyration as a second coordinate as is shown in figure7. The linearization of the folding space is observable as a canyon connecting both states, wheras the main reaction axis is utilized very well between its limits of $(0, 1)$. A disadvantage of the improved separation is the now more blurry interface between the two basins, which might correspond to a wider $p(TP|r)$ distribution. This effect however can be controlled via parameter α in equation 53, which should correlate with the spread of $p(TP|r)$.

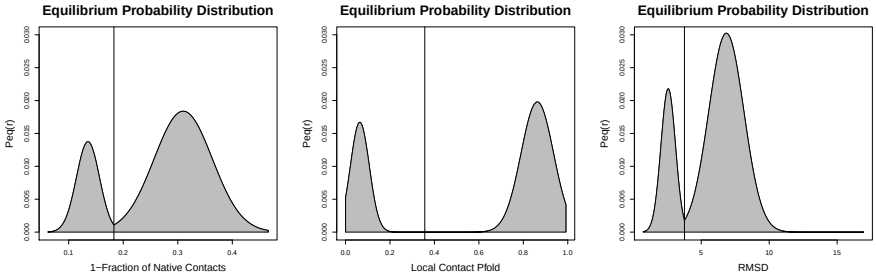


Figure 6: The equilibrium distributions for coordinates contact maps, lcfold and RMSD

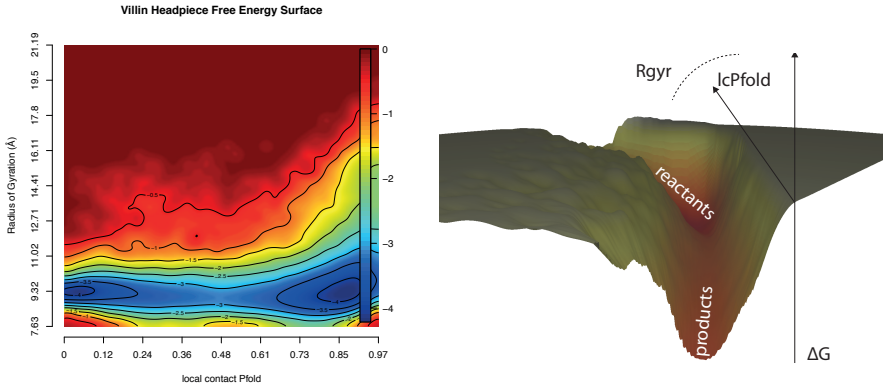
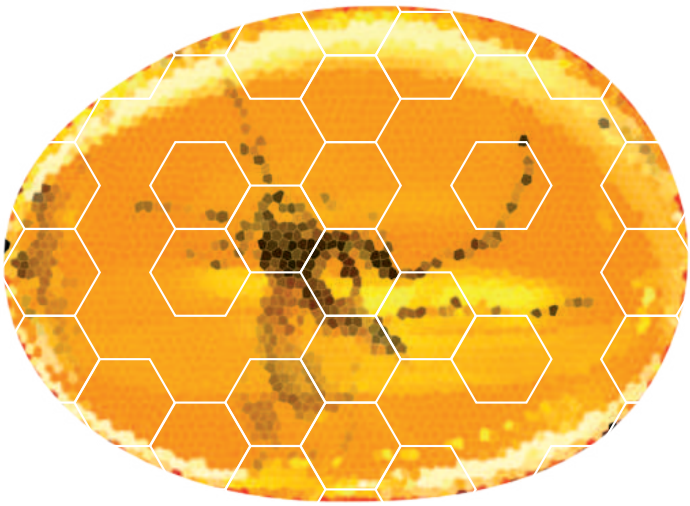


Figure 7: Folding surface for Villin using lcpfold coordinates on the left, and a funnel landscape with a visible linear canyon on the right.

4.2.4 *Conclusion*

We have presented here a novel collective variable which extends the idea of contact map similarity. Through a series of logical steps, the propensity for a conformation S_β to establish contacts as in a reference conformation S_α is evaluated. This is performed by computation of contact maps of S_β and S_α, comparison of the contact maps and calculating a score which relates to the probability that a certain contact mismatch can be corrected with a local conformational search, helped by already established correct contacts. We utilize here a database of precalculated probabilities, which for an arbitrary torsional difference gives the probability that one torsion will assume the other. The process is simplified by disregarding any tertiary or entropical effects, i.e.

we do not investigate if it is actually possible to assume this torsion in the neighborhood of clashing atoms. The probabilities are force-field and temperature dependent and will give different results if those quantities are changed. Furthermore, the method is asymmetrical, i.e. it might be more likely for a structure S_β to fold into S_α than for S_α to fold into S_β. We have compared the method quantitatively to two ubiquitous methods, namely contact maps and RMSD, and found it to be competitive.



A coarse-grained amber stone including fossilized mosquito. "Amber.png" ©2013 aha-soft, used under a Creative Commons license

4.3 AMBERCG

Nils J. D. Drechsel[1] César L. Ávila[2] Jordi Villà–Freixa[1]

[1]Universitat Pompeu Fabra, Computational Biochemistry and Biophysics Laboratory, Research Unit on Biomedical Informatics, IMIM Hospital del Mar-Universitat and Universitat Pompeu Fabra, C/Doctor Aiguader, 88, 08003 Barcelona, Catalunya, Spain

[2]Universidad Nacional de Tucumán, Departamento Bioquímica de la Nutrición, Instituto Superior de Investigaciones Biológicas (CONICET-UNT), Chacabuco 461 (4000), Tucumán, Argentina, Universidad Nacional de Tucumán

4.3.1 *Introduction*

Although it is a concept known since long, the advent of enhanced computational architectures and, more importantly, the introduction of integrative algorithms and (semi-)automated work-flows have led to a generalisation in the use of multiscale simulations in biomedicine and other disciplines. Initiatives like the Virtual Physiological Human (VPH) in Europe[74], the Biomedical Information Science and Technology Initiative (BISTI) in the USA[185] or the Systems Biology Institute (SBI) in Japan[186], among others, have multiscale simulations as top priorities to provide multilevel models of biomedical interest.

As an example of this new multiscale view and, initially, because of the need to reduce their computational demand, molecular simulations of protein energetics and conformational dynamics have served as a vigorous driving force for the design of new algorithms. Thus, to mention two extreme cases, QM/MM approaches were developed as protocols to link quantum and molecular mechanics descriptions of different regions of the system[187–189]; while other multiscale approaches have been developed to deal with complex conformational[190;191] and energetics problems[192] in proteins.

However, multiscale simulations arguably provide a correct vision of complex multilayer problems, as they allow for a distinctive focus on the aspects that are more relevant in each case. Thus, multiscalability means the flow of information, or the connectivity between two or multiple (abstract) biological layers that coexist in the assumed models, and represent the same context in different levels of detail. In this study, we concentrate on the molecular level and representations of atomistic behavior with different degrees of coarseness.

Protein function is intimately related to its structure, which is determined by the individual interactions among amino acid residues. The interplay between hydrophobic and hydrophilic interactions (including charge-charge and hydrogen bond interactions) determines the overall shape of a protein, i.e. its native state. Such a state, then, corresponds to a global minimum in the free energy surface (FES) of a single protein in solution[193;194]. Studying the conformational space of proteins and peptides can yield hints on their propensity to achieve a given secondary, super-secondary or tertiary structure and, thus, their dynamical behaviour and their energetics. The determination of the FES requires an accurate description of the potential energy surface (PES), along with a statistically robust sampling of the phase space:

a giant task because of the size and the complexity of the biomolecular system. The problem can be summarized in three main items for multiscale molecular simulations in general and for protein folding in particular: the definition of both proper coarse grain and explicit potentials, the connection between the potential energy surfaces at the different levels of detail, and the use of a sampling technique that ensures correct exploration of the relevant regions of both the coarse grain and all-atom PES. A thorough review of the different methods that exist to solve each of these problems can be found in reference [195]. In this article we also propose a protocol based on Warshel's multiscale method that is now tested here. This particular method relies on the assumption that main chain interactions are critical for proper secondary structure modelling, and hence it is possible to build a coarse grain potential that explores the relevant regions in the all-atom potential. With this assumption, the multiscale method is based on standard free energy perturbation techniques to project the reconstructed all-atom FES based on the coarse grain potential as a reference state. The test is carried out on two different problems. First, an analysis of the applicability of the multiscale method is performed by analyzing the behavior of two short peptides: an α–helix with the sequence $(Ala)_{15}$ and a β–hairpin with the sequence $(Val)_5ProGly(Val)_5$. As a consequence of this test, the coarse grain potential presented is shown to have problems in determining the conformational space of α helices and β sheets. The native basins are never visited and the most populated states are far from physically sound structures.

In this article we also present the modularity of the high performance productivity molecular simulation program Adun [196], which will help in the endeavor to design complex protocols for multiscale modeling of protein function. In fact, Adun provides the possibility of sharing molecular simulation templates through a distributed database system [197], bringing the possibility of comparing simulations using different set ups, force fields and algorithms between different laboratories: one of the main objectives of multiscale simulation initiatives in biomedicine [198]. In the first chapter we analyze the original coarse-grained force-field introduced by Messer et al. In the succeeding chapter, this force-field is parametrized as AmberCG, a coarse-grained force-field which is compatible to the Amber series of force-fields. The performance of AmberCG is then tested against the same basic models we used for tests with the original force-field.

4.3.2 EnzymixCG

The coarse grain force-field used here is adopted from Warshel and coworkers[192]. Although it has no official name, we will refer to it as EnzymixCG throughout the text. This force-field replaces all of the protein's side-chains by single spheres which comprise the side-chain's interactions on a higher level. Water is completely removed from the environment and replaced by correction terms to the force-field function, which model the solvent as a dielectric continuum. Due to its simplicity and reduction of interacting parts, calculation of energies and forces is vastly accelerated. Another contributing factor to the acceleration is the removal of high frequency vibrational modes through the deletion of almost all hydrogens and side-chain parts. As a result the force-field enables the simulated proteins to move through phase space in a smoother and faster way, thus avoiding unwanted stalls in small minima on a rough energy surface. The force field is based on the following representation of the potential energy surface, following the notation in[192]:

$$\begin{aligned} U_{sp}(\boldsymbol{R}) &= U_{mm0} + U_{ss0} + U_{ms0} + U_{solv0} \\ &= U_{mm} + U_{ssef} + U_{ssQQ} + U_{sself} + U_{msef} + U_{msQq} + \\ &\quad + U_{mmHB} + U_{mmphi-psi} + U_{mmqq} \end{aligned} \tag{58}$$

where "m" and "s" refer, respectively, to "main" and "side" chain contributions, where the side chain is described by a single sphere centered in the centroid of the all-atom representation of the amino acid. **R** represents the coarse grain coordinates. The detailed expression for the different terms in Eq. 58 can be found in the original publication[192], but we want to emphasize here two expressions.

First, the effective electrostatic contributions are given by

$$U_{msQq} + U_{mmqq} = \sum_i \sum_j 332 \left(\frac{Q_i q_k}{\varepsilon'_{eff} r_{ik}} \right) + \sum_k \sum_{k' \neq k} 166 \left(\frac{q_k q_{k'}}{\varepsilon''_{eff} r_{kk'}} \right) \tag{59}$$

where Q_i and q_k refer to charges in ionized residues and residual atomic charges on the main chain, while $\varepsilon''_{eff} = 10$ and $\varepsilon''_{eff} = 4$. Second, as the isotropic dielectric represented by these two terms is not able to

properly reproduce the hydrogen bond energy, an additional term is introduced[192]:

$$U_{mmHB} = \begin{cases} -9 & r \leqslant 2.0 \\ -9 \exp\left(-15(r-2.3)^2\right) & r > 2.0 \end{cases} \tag{60}$$

These two terms, along with the $U_{mm}phi-psi$ term are mainly responsible for the determination of secondary structure propensities.

4.3.3 *Multiscale Free Energy Perturbation*

In order to analyze the ability of the coarse grain model to reproduce the overall shape of the potential energy surface, an experiment was carried out on two model peptides exhibiting extreme secondary structural behaviors. On the one hand, the peptide $(Ala)_{15}$ was explored as an example of pure α–helix structure, while $(Val)_5ProGly(Val)_5$ was used to explore the ability of our approach to determine the FES for a typical β–hairpin. The figure 8 contains the native structures of the two model peptides on the right side, and an extended conformation on the left side, from which the simulations were started. The results of the simulations are shown in figure 24. Top row shows results for $(Ala)_{15}$, bottom row for $(Val)_5ProGly(Val)_5$. On the left side are conformations for EnzymixCG and on the right for AmberCG.

The results of the REMD runs (see methods section) are summarized in 9. This shows a representation of the RMSD across a large series of structures versus the total potential energy obtained for them. The funnel-like shape, representing the expected behavior for an ideal PES with a unique deep minimum, is not entirely recovered by the original coarse grain force field. In particular, the plot for the β–hairpin is clearly bimodal, a symptom of more than one optimal structure being explored. In addition, the minimum RMSD of the explored structures is far from the original system, which suggests problems in the ability of the coarse grain potential to find the native topology. Interestingly, the hydrogen bonding pattern (figure 10) appears to follow the expected funnel shape (figure 10) while the main chain torsions are contrary to the expected behavior (figure 11).

In order to analyze the characteristics of the different regions explored by the simulation, a projection of the free energy surface for the coarse grain potential onto two global coordinates, RMSD and R_g, is shown in 12. In the case of the α–helix, the figure shows the presence of a deep minimum in the lowest surface region, a representative

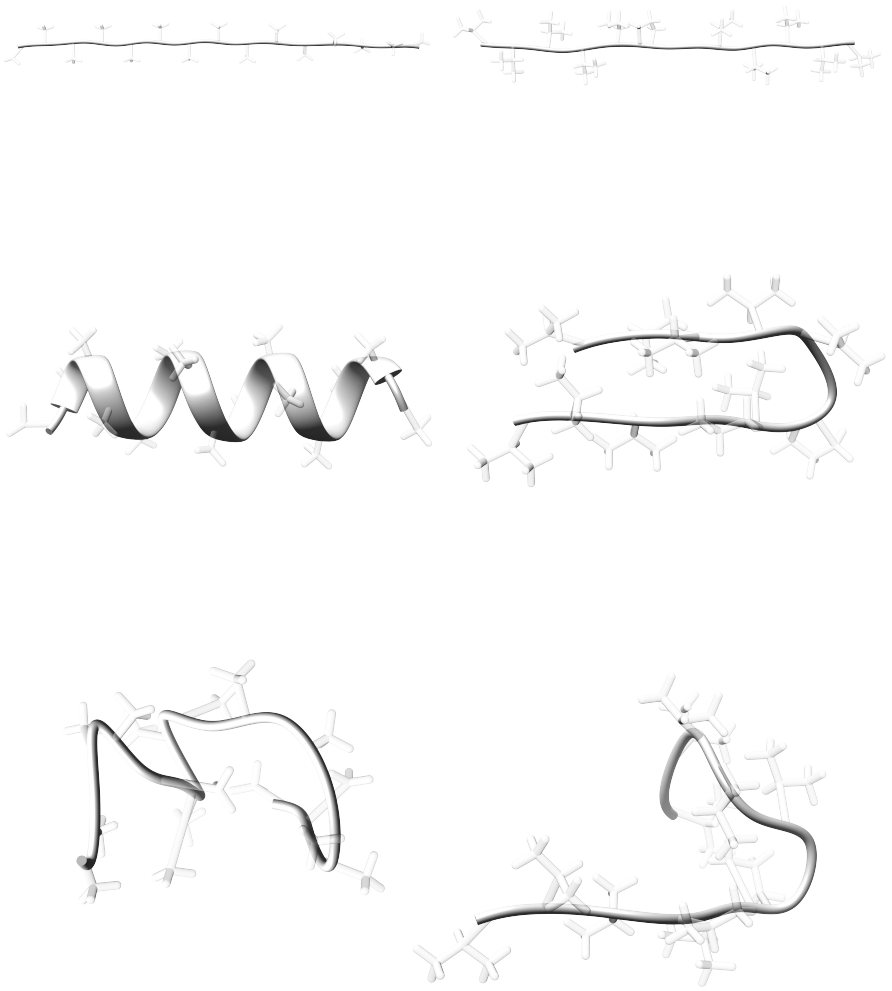


Figure 8: Folding simulation of a $(Ala)_{15}$ (left column) and $(Val)_5ProGly(Val)_5$ (right column). The top row shows fully extended conformations, middle row folded reference conformations, and bottom row conformation taken from the free energy minimum after the analysis of the trajectories

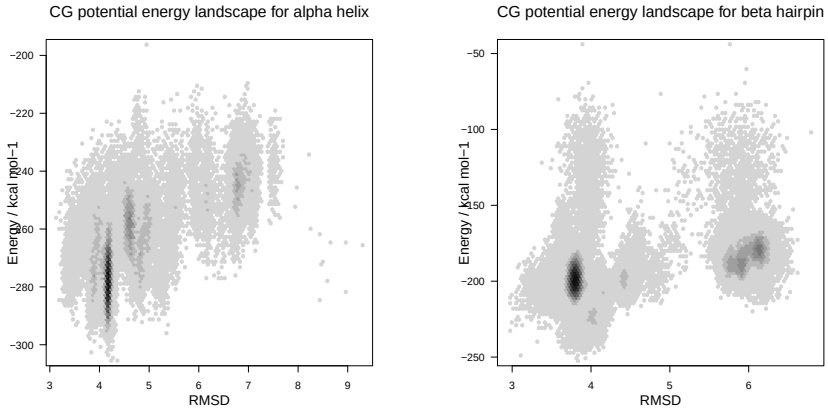


Figure 9: Correlation between RMSD and Energy for the complete sampling space of the model peptides. Hexagon binning for snapshots of the 300K trajectory for $(Ala)_{15}$ (left) and $(Val)_5ProGly(Val)_5$ (right)

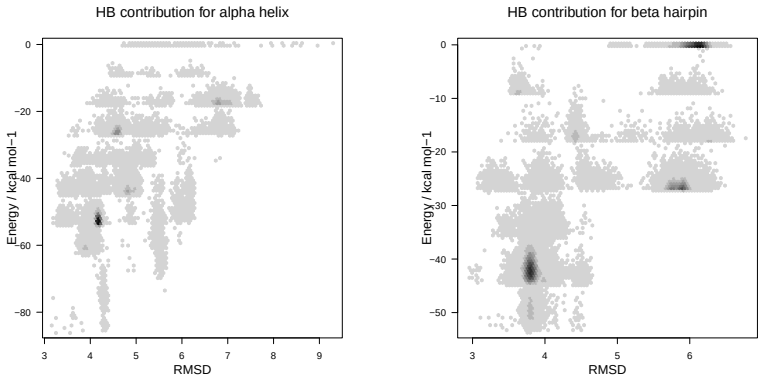


Figure 10: Correlation between RMSD and Hydrogen Bond energy contribution for the model peptides. Hexagon binning for snapshots of the 300K trajectory for $(Ala)_{15}$ (left) and $(Val)_5ProGly(Val)_5$ (right).

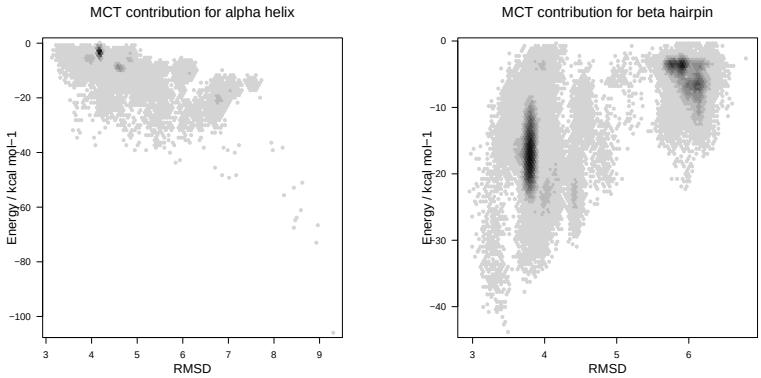


Figure 11: Correlation between RMSD and main chain torsion energy contribution for the model peptides. Hexagon binning for snapshots of the 300K trajectory for $(Ala)_{15}$ (left) and $(Val)_5ProGly(Val)_5$ (right).

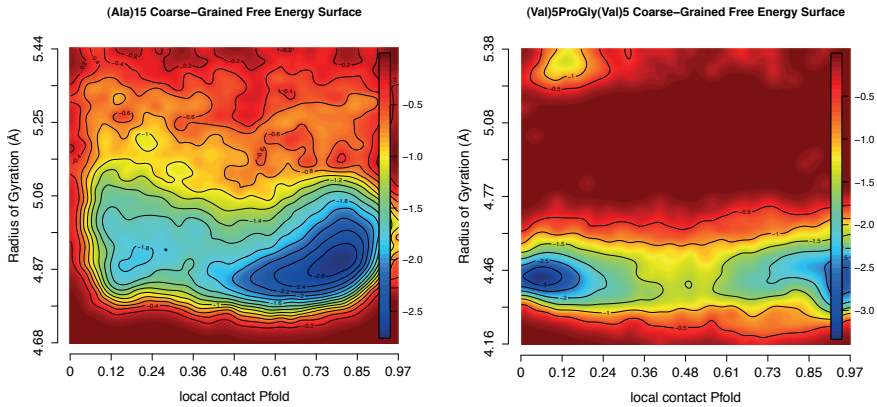


Figure 12: Coarse-grained peptide folding FES. Data for $(Ala)_{15}$ (left) and $(Val)_5ProGly(Val)_5$ (right) are depicted. The coordinates are calculated against a model α–helix and β–hairpin respectively.

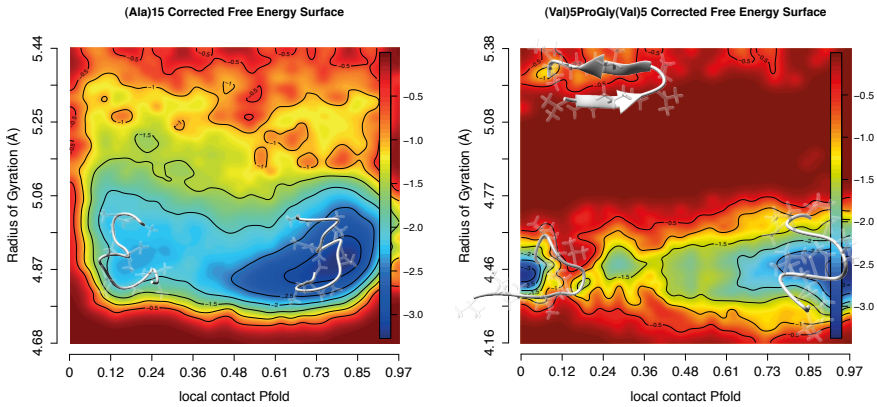


Figure 13: Peptide folding FES. Data for $(Ala)_{15}$ (left) and $(Val)_5ProGly(Val)_5$ (right) are depicted. The coordinates are calculated against a model α–helix and β–hairpin respectively.

structure of which is presented in 8. With respect to the β–hairpin, the structure is clearly distorted from the native arrangement. In both cases, the CG folding FES shows a single dominant non-native minimum present in addition to the native basin[199]. In agreement with the previous figures, it seems clear that the coarse grain model is problematic in handling β secondary structure interactions. The origin of this problem is not the HB pattern, but, as suggested above, the $\mathtt{Uphi-psi}$ term. Despite the limitation of the coarse grain potential, especially for β–hairpin structures, it is possible to demonstrate the feasibility in building a complete pipeline for multiscale simulations in a portable software like Adun. Therefore, we analyze the ability of the method to reconstruct the free energy surface for the all-atom system. Figure 13 shows the results for the corrected FE, obtained as follows. First a series of 32×32 representative structures were obtained from coarse-grained simulations of $(Ala)_{15}$ and $(Val)_5ProGly(Val)_5$. For each of these a free energy perturbation (FEP) protocol was carried out in which one slowly transforms the structure from a coarse grain representation to an all-atom representation (see the Methods section for details). As the FEP involves the appearance of new atoms in the system, it is obvious that care must be taken not to prevent the explosion of the simulation right after the first (fully coarse grain) FEP window. The final reconstructed all-atom FES is shown in figure 13. In terms of the final structures, the expected α–helix and β–hairpin structures

are compared with the central structures in the relevant minimum region in 8. The last row in 8 shows a representative structure for the minimum FE regions in 13. In 13 it can be seen that the FEP protocol does not significantly introduce changes into the shape of the CG FES when moving to the all-atom representation. This result is significant, as it shows that the global pipeline of the method is entirely sound, as it brings about the possibility of improving individual modules of the protocol, in particular the quality of the coarse grain potential used as reference for the FEP.

4.3.4 *Parametrization of a new coarse-grained force-field*

Next, a new coarse-grained force-field, called AmberCG, is adopted from the original version from Messer et al. The same functional form is used, but the parameters are changed to conform to the AmberXX series. The objective was to reproduce free energies surfaces, derived from a joint cooperation of Amber96[200] and Generalized Born OBC[51]. Mixtures of Amber force-fields and Generalized Born Methods do not always work well together. In a study by Shell et al.[201], where a number of different Amber force-fields was tested against various Generalized Born variants, Amber96 with GB OBC seemed to produce the best trade off in terms of stability and fold correctness.

For Amber96 we use the general functional form:

$$V(r)_{\text{AmberAA}} = \sum_{\text{bonded}} + \sum_{\text{non-bonded}} + \sum_{\text{solvation}} \tag{61}$$

The bonded and non-bonded terms are expressed through their bond, angular, torsional, van der Waals and electrostatic parts:

$$\sum_{\text{bonded}} = \sum_{\text{bonds}} k_b(b-b_0)^2 + \sum_{\text{angles}} k_\theta(\theta-\theta_0)^2 + \sum_{\text{dihedrals}} (V_n/2)(1+\cos[n\phi-\delta]) \tag{62}$$

$$\sum_{\text{non-bonded}} = \sum_{\text{LJ}} (A_{ij}/r_{ij}^{12}) - (B_{ij}/r_{ij}^{6}) + \sum_{elec} (q_i q_j/r_{ij}) \tag{63}$$

The functional form of the solvation term can be found in Onufriev et al.[51].

In our new AmberCG model, the main-chain main-chain interactions are handled explicitly, and their representation is all-atom. Parameters for these interactions are taken from the Amber96 parameter library and are not subject to reparametrization / optimization. Reparametrization is performed for all interactions involving coarse-grained side-chains, i.e. van der Waals between side-chains, van der Waals between side-chains and main-chain, hydrogen-bonds, side-chain main-chain bonds, side-chain angles and torsions.

Reparametrization of coarse-grained potentials is a task that can be accomplished in many ways (Force-Matching[142], Boltzmann-Inversion[139], or Subtraction Method[140]) dependent on the quantities of interest. In the most rigorous optimization approach, theoretically, one would use an optimizer which would compare free energy surfaces, and move into the direction of the surface that is the least different from one obtained by all-atom simulations. This approach is utterly infeasible considering the computational demand. We use here a technique where we compare against potential energy surfaces, which in comparison to free energy surfaces lack an entropic part. However, the coarse-grained free energy surfaces do not have to be absolutely correct - they will be corrected by the multiscale approach anyways. We just need to make sure the force-field will be capable of sampling appropriate parts of phase-space.

For this purpose, the small proteins Tryptophan-Cage and Tryptophan-Zipper were simulated using the Amber96 force-field in cooperation with Generalized Born OCB until their phase space was adequately sampled. 10 000 structures, each, were randomly selected as reference conformations. The potential energy + solvation free energy of these conformations was calculated and subsequently used by the optimizer to calculate the Pearson correlation coefficient between all-atom conformations and coarse-grained counterparts.

As an optimizer, we employed a genetic algorithm, which generated coarse-grained force-field parametrization. These parametrizations were tested by calculating coarse-grained energies for all 20 000 structures and computation of the aforementioned Pearson correlation coefficient. The coefficients were directly used as a fitness measurement to guide the genetic algorithm. After a series of initial test to estimate best settings for the algorithm, a final production run was performed on the LaPalma supercomputer for 325 iterations (see figure 14).

The energy correlations for the final, best individual of the production run are shown in figure 15.

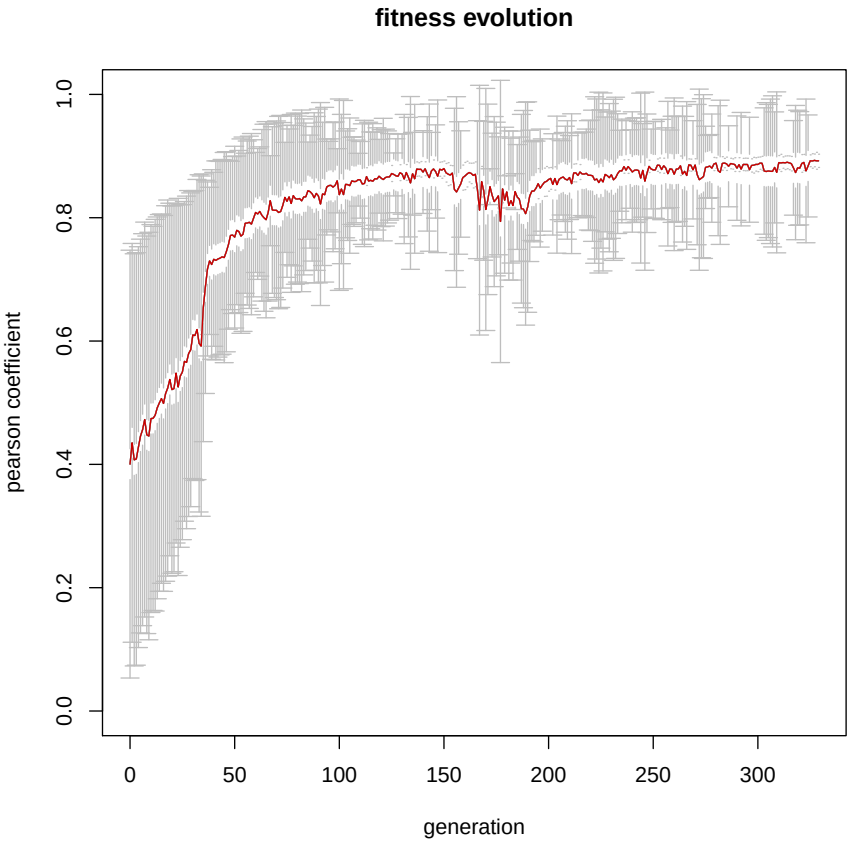


Figure 14: Evolution of the correlation between all-atom and coarse-grained potential energy surfaces

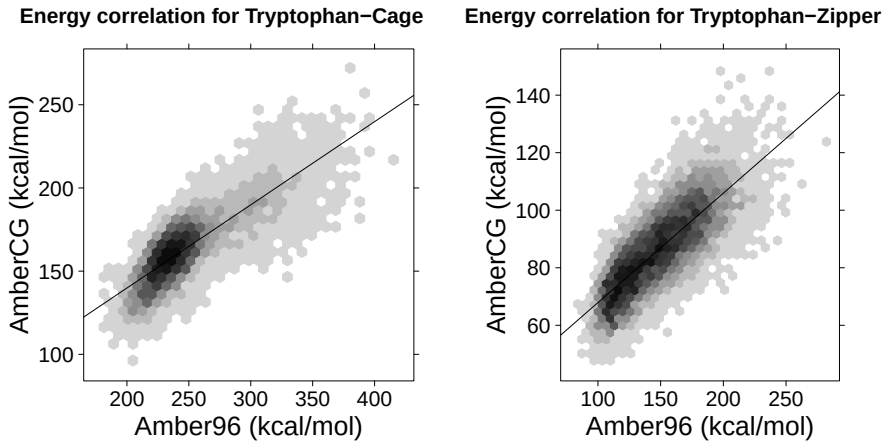


Figure 15: Correlations between all-atom and coarse-grained energies for Tryptophan-Cage (left) and Tryptophan-Zipper (right)

4.3.5 *Evaluation*

Here, we analyze the coarse-grained force-field parameters after extraction from the genetic algorithm. A statistical population analysis of production run is shown in figure 16. It shows the variability of the parameters throughout the optimization. We can clearly see that the genetic algorithm was very prudent to change hydrogen-bond values, indicating that this is a very sensible parameter to be adjusted. Likewise, some of the torsions have low variability. Torsions have been the main optimization targets for improvements in all-atom force-fields over the last years. These interactions compromise more than just angular constraints, but include often electrostatic and other non-bonded components as well. It is therefore not surprising that these interactions have to be treated very carefully - especially since we are optimizing against structures of mostly helical and beta-sheet nature; and they are those secondary structures elements that depend the most on the proper handling of torsions. A series of tests have been performed to check agreement of the reproducibility of all-atom energies by AMBERCG. 10 ns molecular dynamics trajectories of 22 structures have been obtained through the MODEL database[202]. These structures exhibit a vast variety of different folds and were originally described by Rueda et al.[203]. The objective is to take a subset of the trajectory for all structures, and compare their all-atom and coarse-grained potential + solvation free energies. MODEL-trajectories are compressed by an es-

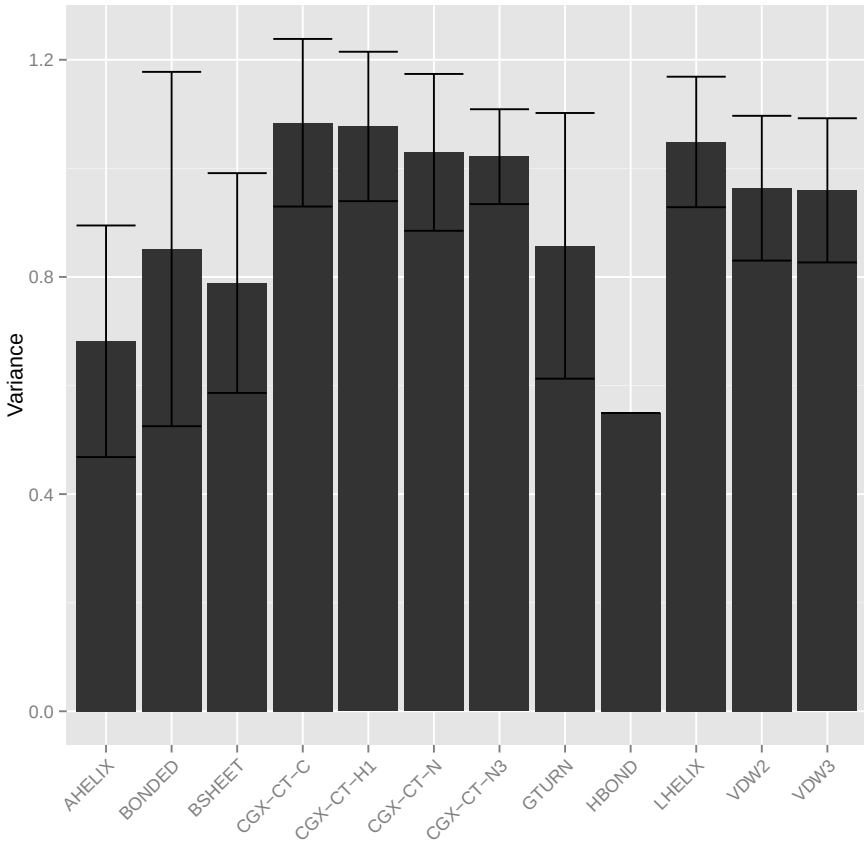


Figure 16: Variation in optimized parameters. A small variance indicates that the optimizer held the parameter relatively constant, implying that it is very sensitive to change.

sential dynamics type algorithm[204] and therefore need to be extracted. The compression is not loss-less and as such the structures cannot be used as is. We use the following protocol to generate structures to compare against:

1. Download of structures from Amber99SB trajectories
2. Decompression using the tool described in reference[204]
3. Addition of Hydrogens using the tool reduce[205]
4. Minimization using Rosetta (this will shift the structures slightly out of the Amber99SB potential, but will fix problems with side-chains), no backbone minimization
5. Minimization using ADUN[206] and Amber96 potential, allowing for minor backbone minimization, which shifts structures back into a compatible potential
6. Removal of the structures, which remain unphysical even after minimization, using a one-dimensional outlier detector
7. Convertion of structures into coarse-grained representations

Afterward, Pearson correlation coefficients were calculated for the entire set of molecules. Table 1 shows the individual correlations and the average correlation coefficient, figures 17 to 23 show individual correlations per structure.

Lastly, the folding free energy surfaces of the two model peptides, $(Ala)_{15}$ and $(Val)_5ProGly(Val)_5$ were repeated with the new AmberCG force-field. Simulations of 0.8 µs each were performed, starting from an extended conformation as depicted in 8. As expected, the results have dramatically improved. $(Ala)_{15}$ is folded almost perfectly. The area of lowest free energy corresponds very well to the native basin. The native basin in $(Val)_5ProGly(Val)_5$ is almost reached, sampling however stopped at a not completely elongated hairpin. Even though qualitatively $(Val)_5ProGly(Val)_5$ looks more visually appealing in the EnzymixCG structure (compare figures 8 and 24, the AmberCG structure scores better in both RMSD and lcpfold. The reason is that although it is not fully elongated, the amino-acid-amino-acid matching between the two strands of the hairpin match those of the native structure, while they are shifted in the EnzymixCG structure.

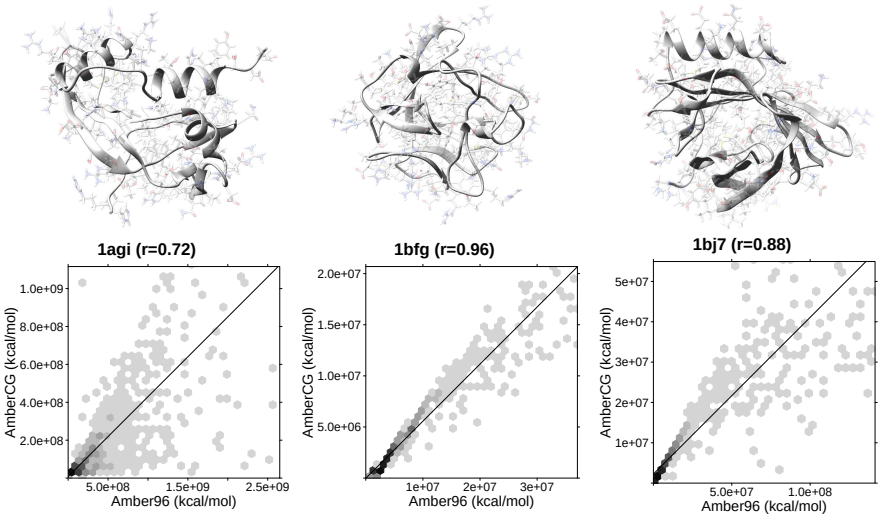


Figure 17: Correlations for structures 1agi, 1bfg, 1bj7

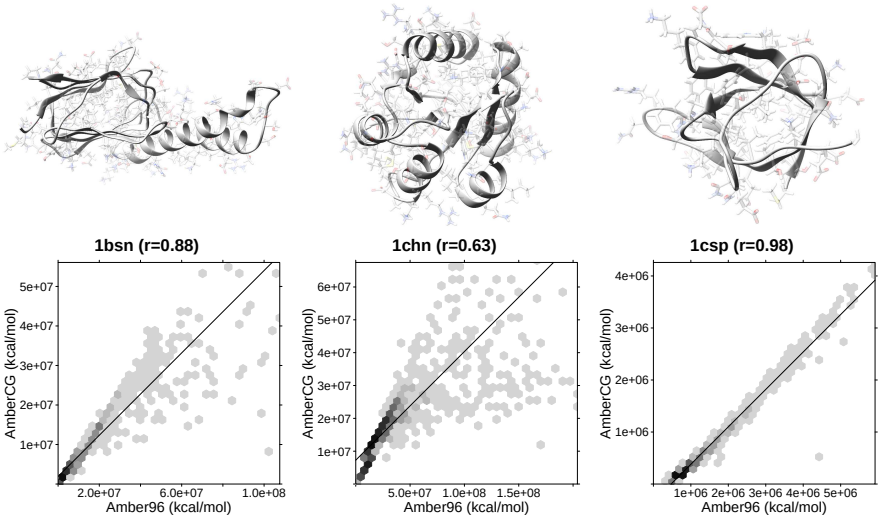


Figure 18: Correlations for structures 1bsn, 1chn and 1csp

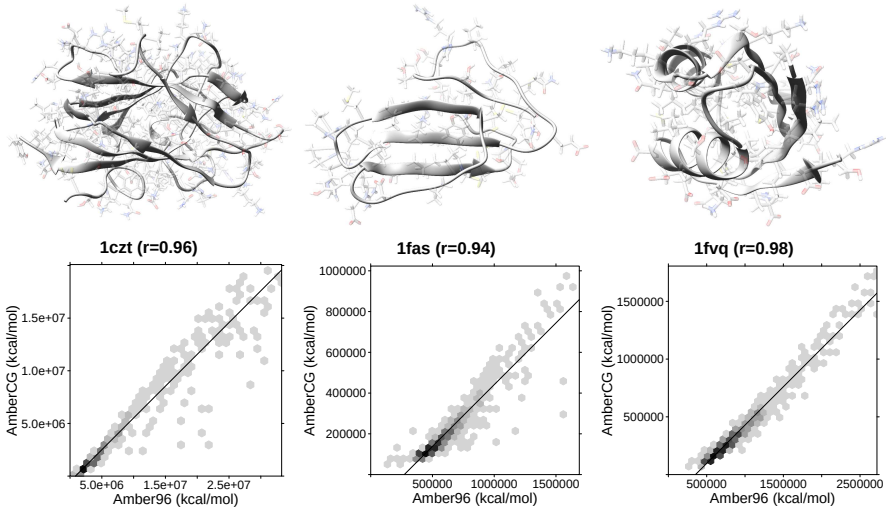


Figure 19: Correlations for structures 1czt, 1fas and 1fvq

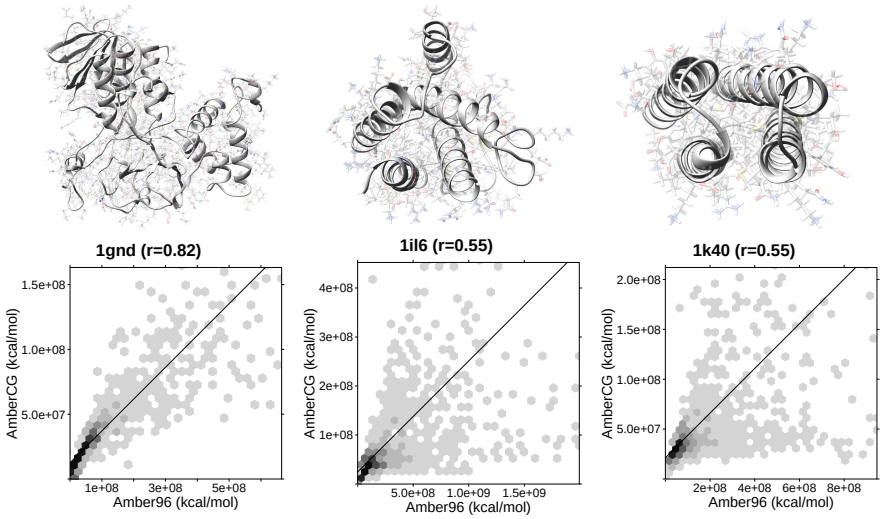


Figure 20: Correlations for structures 1gnd, 1il6 and 1k40

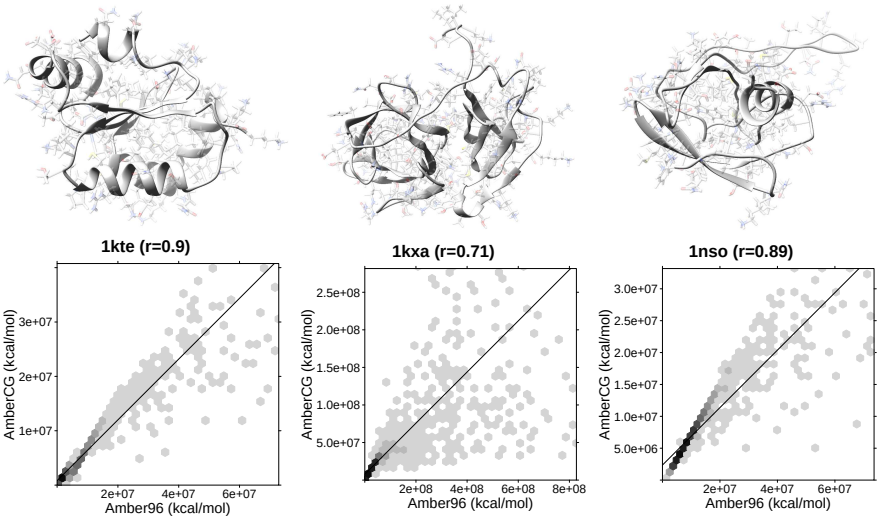


Figure 21: Correlations for structures 1kte, 1kxa and 1nso

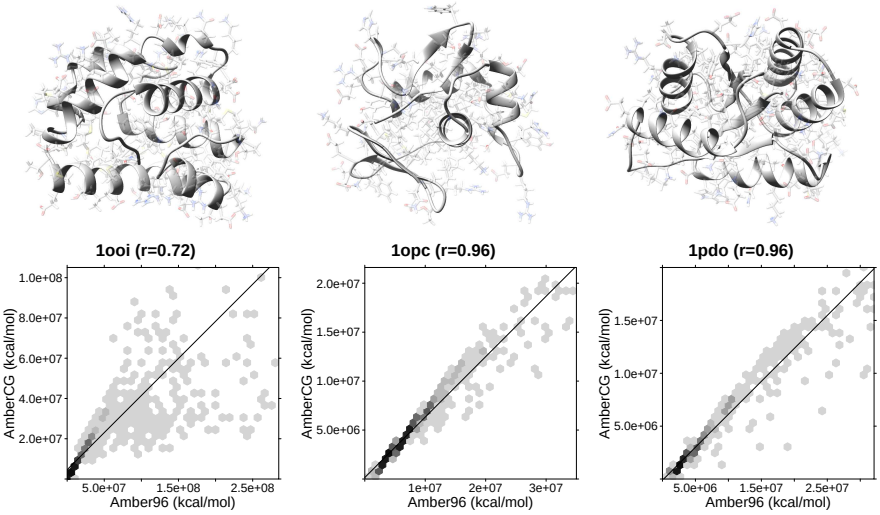


Figure 22: Correlations for structures 1ooi, 1opc and 1pdo

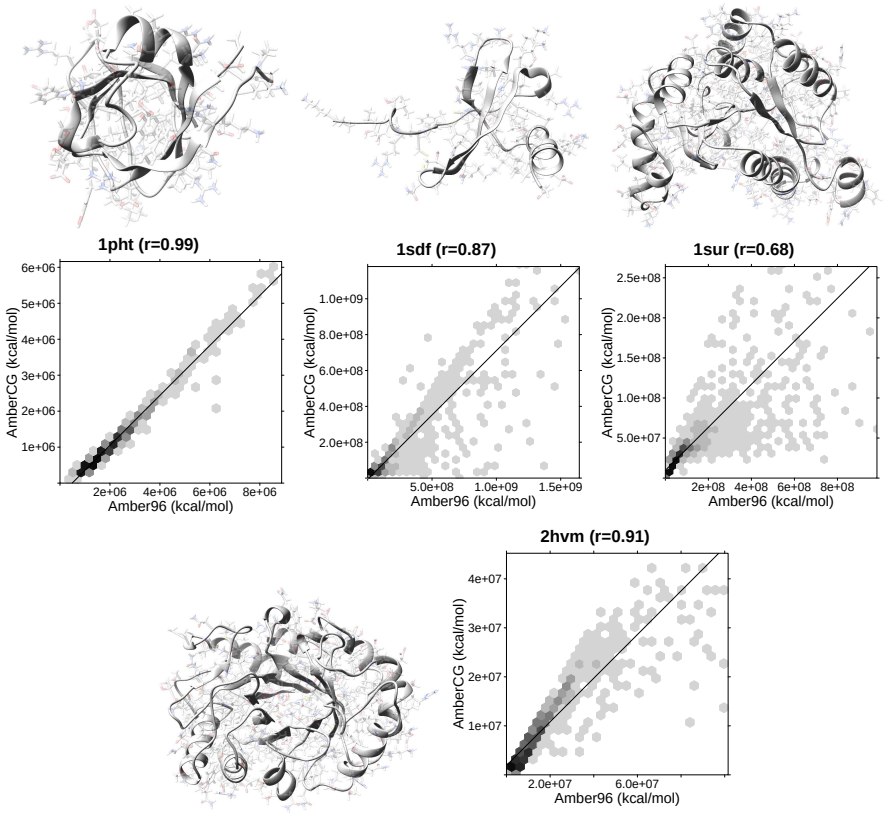


Figure 23: Correlations for structures 1pht, 1sdf, 1sur and 2hvm

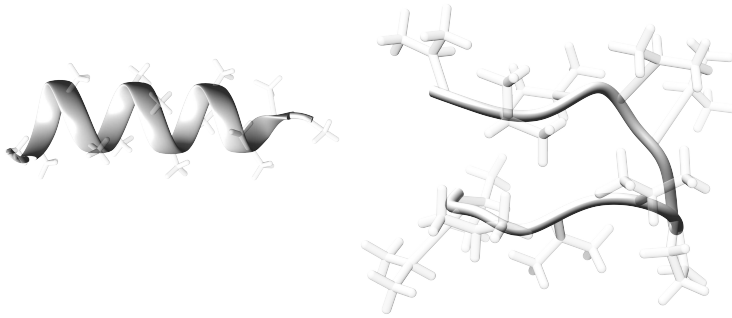


Figure 24: AmberCG structures from the free energy minima for $(Ala)_{15}$ (left) and $(Val)_5ProGly(Val)_5$ (right)

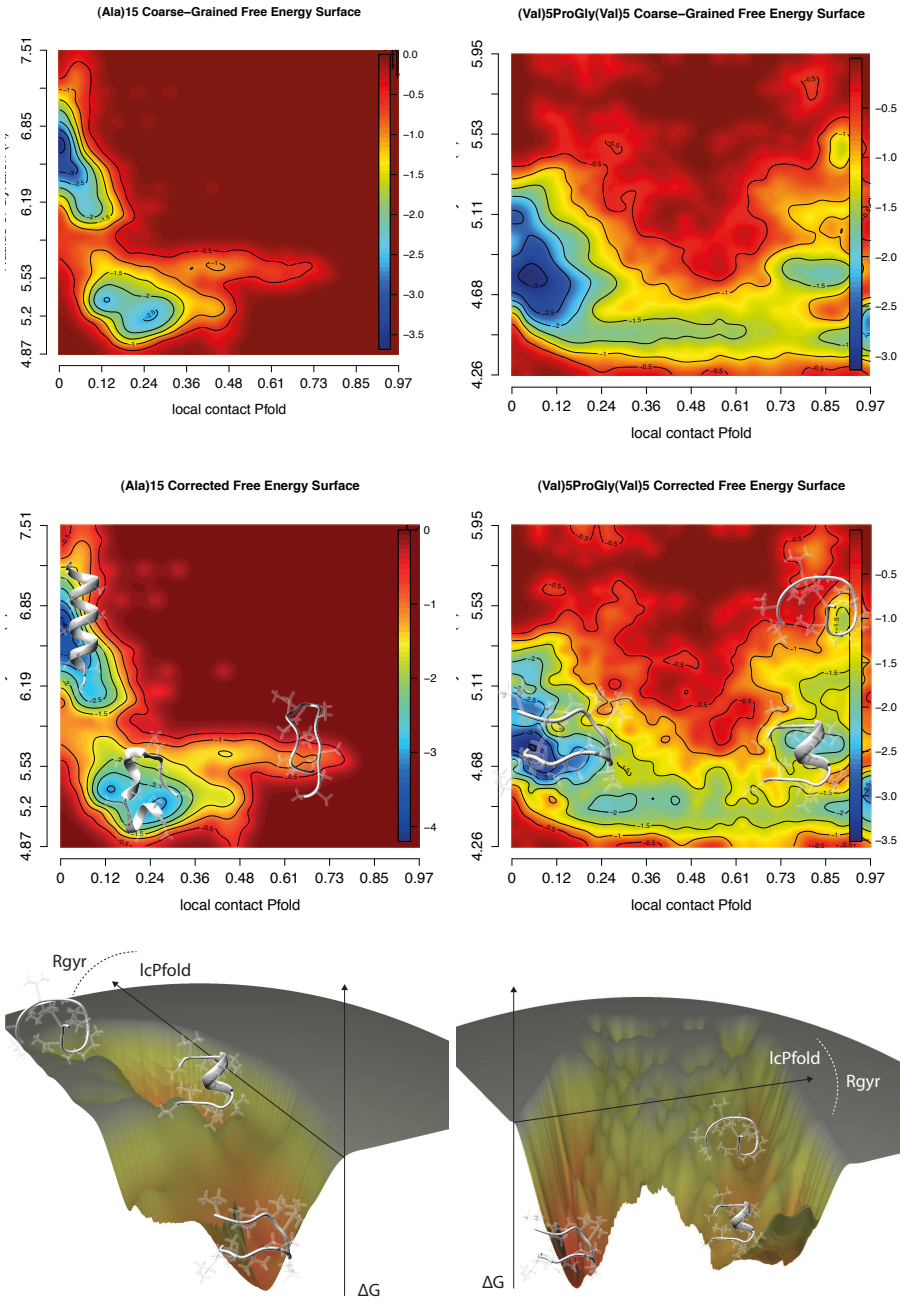


Figure 25: Peptide folding FES. Data for $(Ala)_{15}$ (left) and $(Val)_5ProGly(Val)_5$ (right) are depicted. The coordinates are calculated against a model α–helix and β–hairpin respectively. The top row shows coarse-grained free energy surfaces, the middle row corrected surfaces, and the bottom row corrected surfaces from a funnel perspective.

PDB code	AmberCG Correlation	EnzymixCG Correlation
1agi	0.72	0.09
1bfg	0.96	0.00
1bj7	0.88	0.01
1bsn	0.88	0.1
1chn	0.63	-0.01
1csp	0.98	-0.14
1czt	0.96	0.08
1fas	0.94	-0.07
1fvq	0.98	-0.18
1gnd	0.82	0.05
1ilj	0.55	0.08
1k40	0.55	-0.1
1kte	0.90	0.05
1kxa	0.71	0.08
1nso	0.89	0.00
1ooi	0.72	0.08
1opc	0.96	0.00
1pdo	0.96	0.02
1pht	0.99	-0.06
1sdf	0.87	0.00
1sur	0.68	0.08
2hvm	0.91	0.15
average	0.83	0.01

Table 1: Protein set with correlation coefficients for AmberCG and EncymixCG to all-atom energies

4.3.6 *Folding study of the Villin Headpiece and discussion*

The new AmberCG force-field is now tested against the 35 residue segment of the Chicken Villin Headpiece. The Villin is a target, which has been intensely studied in the past - which is why it is a perfect target for benchmarking. We have performed three sequential replica exchange simulations for a total of a bit above 3 µs coarse grained simulation time. The folding free energy surface is shown in figure 26. We can see that very quickly a condensed state is reached that already shares more similarity to the native conformations than to denatured conformations. The protein is successively compressed more, until a very stable state is reached from which the protein didn't seem to be able to escape. The state is not part of the folded ensemble, even though the general form is similar. The structure of the lowest free

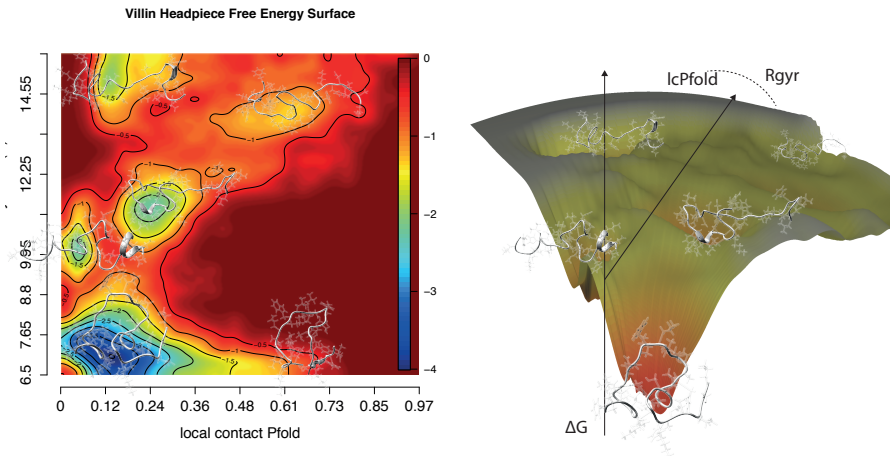


Figure 26: Folding landscape of the Chicken Villin Headpiece with AmberCG

energy bin is depicted in figure 27 as the solid structure, in which the fully folded Villin is overlaid for comparison. We can clearly see that the force-field is able to only partly fold the protein. It generally assumes the correct three dimensional shape and in principle also partly forms the correct two dimensional structure - which is why we see good lcpfold scores for this basin. However, it seems that frustration causes an imbalance and a struggle between the right tertiary and secondary structure, such that it is not able to fully collapse into the folded conformation.

It might be the case that the reference structures for the force-field parametrization were too simple, in the sense that trp-cage and trp-zipper exhibit a lot of secondary structure, but the force-field is missing tertiary components. In the quantitative energy correlation plots, where we compared a series of different folds, it is observable that we attain good correlations only in structures with at least some content of beta sheets, which was not the case for Villin. The conformation closest to the native conformation exhibits a distance of about 4.9 Å RMSD, which is about 1 Å larger than has been reported for other coarse-grained simulations[207–209] and some all-atom simulations[210;211], but much farther away than other all-atom simulations[212].

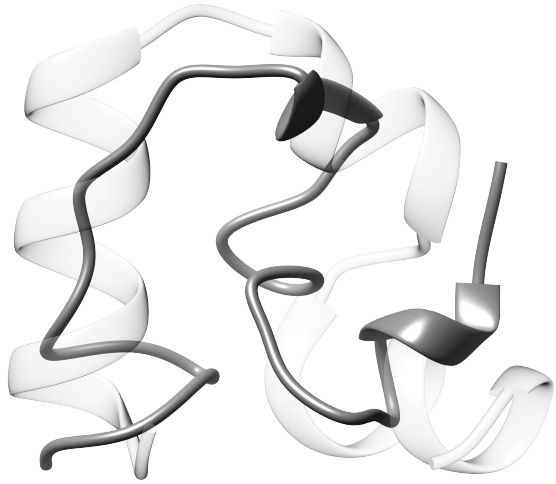
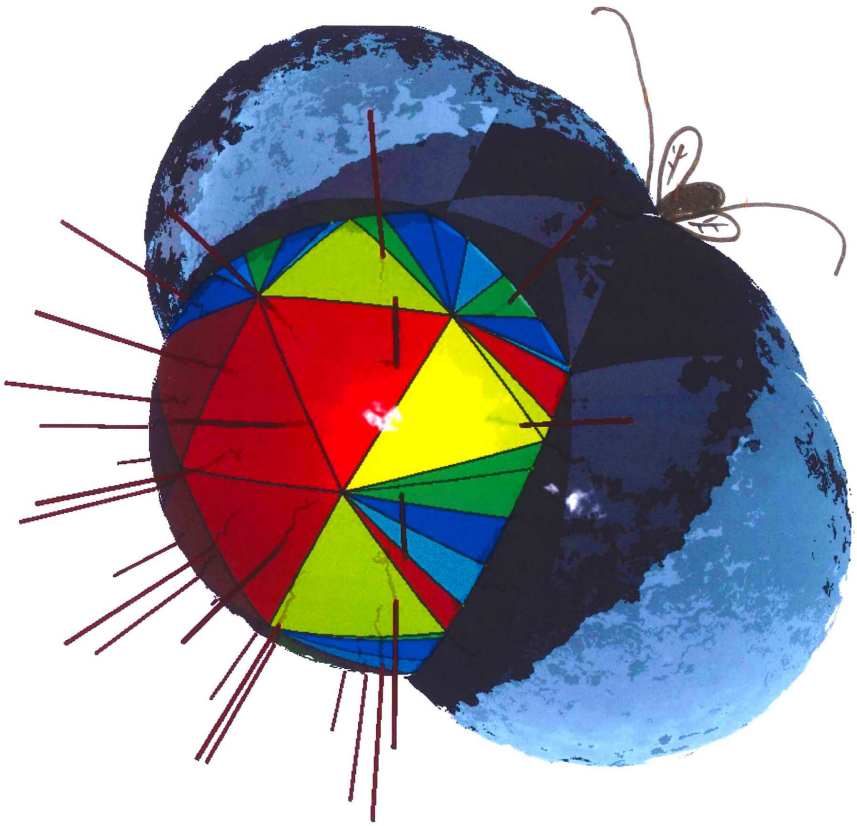


Figure 27: The lowest free energy conformation compared to folded Villin

The next two results introduce TRIFORCE, an algorithm to perform functional value integration on bounded spherical surfaces, as well as an algorithm to quickly detect the bounds of these surfaces. Its development was inspired by the necessity to integrate Lennard-Jones parameters as functional values on exposed areas in the context of the Semi-Explicit Assembly method. The algorithm is presented in its special case, in which the functional value is simply 1, which corresponds to the calculation of surface areas.



This picture was a test print of the first pentakisdodecahedron-based tessellation algorithm (a pre-TRIFORCE so to say), with modifications from Ignasi Buch

4.4 TRIFORCE: TESSELLATED SEMI-ANALYTICAL SOLVENT ACCESSIBLE SURFACE AREAS AND THEIR DERIVATIVES

Nils J.D. Drechsel, Christopher J. Fennell, Ken A. Dill, and Jordi Villà-Freixa

"TRIFORCE: Tessellated Semi-Analytical Solvent Accessible Surface Areas And Their Derivatives"

Submitted to Journal of Computational Chemistry

TRIFORCE: Tessellated Semi-Analytical Solvent Exposed Surface Areas And Their Derivatives

Nils J. D. Drechsel[1,2], Christopher J. Fennell[2], Ken A. Dill[2], and Jordi Villà–Freixa[3]

[1]*Computational Biochemistry and Biophysics Laboratory, Universitat Pompeu Fabra, Research Unit on Biomedical Informatics, C/Doctor Aiguader, 88, 08003 Barcelona, Catalunya, Spain*

[2]*Laufer Center for Physical and Quantitative Biology, Stony Brook University, Stony Brook, New York 11794-5252, USA*

[3]*Escola Politècnica Superior, Universitat de Vic, C/ de la Laura, 13, 08500 Vic, Catalunya, Spain*

The calculation of solvent exposed areas of molecules is of key interest in molecular dynamics and adjacent disciplines due to their direct correlation to the nonpolar solvation free energy. In the last decades, numerous methods of analytical, numerical, probabilistic and statistical nature have been developed to provide ever faster ways to compute or approximate this quantity. We present in this article a novel algorithm for the calculation of the exposed area, with unique properties that go beyond simple area calculations, although those calculations are the focus of this article. The algorithm performs a special tessellation of the exposed area, and is able to integrate the area, as well as predefined surface quantities of the tessellated surface patches in a fast and efficient way. The integration is performed semi-analytically with the utilization of a precomputed look-up table. Derivatives are calculated in the same way, enabling its application to minimization or molecular dynamics simulations. The algorithm is available free of charge for academical purposes in a library written in C++, readibly interfaceable to molecular simulators, as well as accessible online through a webinterface.

Simulations of systems at molecular levels of detail involve approximations that remove complexity in favor of computational tractability. One common approximation that makes macromolecular simulations, like protein folding and association, tractable is to use implicit rather than explicit solvent. Implicit solvents trade the detailed accounting of water, lipid, or other surrounding solvent molecules for an averaged effective solvent representation. To do this, the many explicit degrees of solvent freedom are integrated out of the system and replaced with free energies of solvation. Free energies of solvation have been observed to be well-correlated with the interfacial or solvent accessible surface area (SASA) in the case of non-polar solutes[1–4]. Many methods for implicit solvation have been developed which take advantage of this observation to provide quantitative predictions for solvation free energies using the SASA[5–14]. To perform simulations involving molecular dynamics and minimization with such implicit solvents, we need not only per atom SASAs, but how those areas change with changes in the atomic coordinates. Several algorithms and methods have been developed which can compute these areas and their derivatives spanning from analytically exact approaches[15–18] to statistical or numerically approximate ones[19–22].

We are interested in performing molecular simulations in implicit solvents that report solvation free energies with a high degree of accuracy. Analytical approaches available are often performance limiting, and statistical approaches are often marked by substantial non-uniform errors. Additionally, many methods are physically limited in some way, such as the inability to treat hydrogen atom contributions to the SASA. Here we present an alternative semi-analytical approach for computing Lee-Richards SASAs and their derivatives from coarse-grained or all-atom representations of molecular systems[23]. This method, which we refer to as TRIFORCE, employs a precomputed look-up table that enables accelerated determination of component areas and derivatives as a function of the principle angles defining boundary interfaces. We evaluate the correctness of this approach with comparisons to analytical methods and show that the numerical accuracy is determined by the density of this look-up table, with the primary cost being one of memory rather than compute cycle utilization.

Despite the large amount of surface area algorithms, development of the method was also encouraged by the lack of readily accessible, working, and robust

methods, which are easily interfacable to other software like molecular simulators. It exists a gap between the theoretical landscape of such algorithms and their practical usage, which we aim to fill.

0.2 AREA CALCULATION

A molecule is modeled as an aggregation of intersecting spheres with radii corresponding to their extended Van der Walls radii, which include a term to express the average distance to the solvent. The solvent accessible surface area A of a molecule can then be calculated by adding up all solvent exposed areas of all aggregated spheres A_l.

$$A = \sum_l A^l \tag{1}$$

A^l is calculated by summing up areas of lesser complexity. These areas are triangular patches which are constructed by a special tessellation of the exposed area of a sphere, as depicted in figure 1. Each triangular patch consists of two sides which are segments of great circles, and a third side which is a spherical arc. The boundary of the exposed area

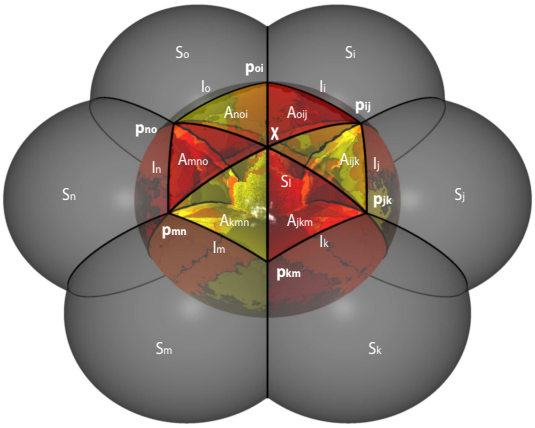


Figure 1: A tessellation of the exposed area of sphere S_l by triangles. The corner of the triangles are compromised by two consecutive intersection-points denoted by $\boldsymbol{p}$, and a point that is the same for all triangles of a sphere S_l. The point is located where the tessellation axis intersects with S_l. The intersection-points $\boldsymbol{p}$ are intersections of circular interfaces I on the surface of S_l. Two sides of each triangle are formed by segments of great circles, the other is a spherical arc.

is composed of these spherical arcs, which are segments of circular interfaces I_i between spheres S_i

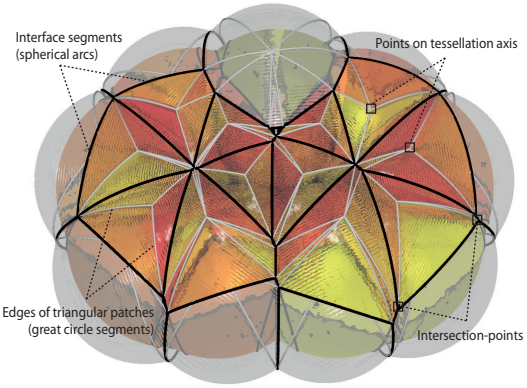


Figure 2: The tessellated molecule fluorene. Black lines depict circular interfaces, white lines segments of great circles. One interface segment and two great circle segments form a triangular patch for which the area will be computed. The three corners of the patch are formed by a point on the tessellation axis, and two intersection-points, which are the result of an interface-interface intersection.

that intersect with S_l (see figure 3). A circular interface I_i is located in the plane of intersection between S_l and S_i where the hulls touch each other. Intersection-points $\boldsymbol{p}_{ij}$ between these circular interfaces exist whenever the interfaces I_i and I_j of spheres S_i and S_j overlap (see figure 5 and 2). Two consecutive intersection-points $\boldsymbol{p}_{ij}$ and $\boldsymbol{p}_{jk}$ on the boundary of the solvent accessible surface area, in conjunction with a point on the tessellation axis $\boldsymbol{\chi}$, which is a fixed vector for each atom for the duration of the calculations, form the three points of a triangular region A^l_{ijk}; With $ijk \in Seg_l$, and Seg_l denoting all tuples (ijk) contained in boundary segments.

$$\hat{A}^l = \sum_{(ijk)\in Seg_l} A^l_{ijk} r_l^2 \tag{2}$$

Areas A^l_{ijk} are calculated on a unit sphere and therefore have to be multiplied with the squared radius r_l of sphere S_l. An area $\hat{A}^l$ is either surface exposed or surface excluded. Depending on its exposition, it needs to be added or subtracted from the total exposed area of S_l respectively. If, at the conclusion of the summation of equation 2 the area is negative, excluded area has been counted. If that is the case, it needs to be added to the total area of S_l to reflect

exposed area.

$$A^l = \begin{cases} \hat{A}^l & \hat{A}^l \geqslant 0 \\ 4\pi r_l^2 + \hat{A}^l & \text{else} \end{cases} \tag{3}$$

The triangular region A^l_{ijk} consists of two sub-areas, which are looked up and interpolated from a precomputed integration table $T(\Phi, \psi, \lambda)$. The sub-areas are subtracted from each other and multiplied by the squared radius r_l of atom S_l. The parametrization of the integration table $T(\Phi, \psi, \lambda)$ is comprised by three principle angles of boundary interfaces: First, the angle Φ between the tessellation plane and a vector from the center of the interface I_j to an intersection point $\boldsymbol{p}$. The tessellation plane is spanned by the tessellation axis $\boldsymbol{\chi}$ and the interface normal $\boldsymbol{\mu}_k$. Second, the angle ψ_j between the tessellation axis and the center of the interface I_j. Third, the opening angle λ_j of the cone constructed from the interface I_j and the center of sphere S_l. From now on, whenever unambiguous, we omit the index l.

The interfaces are calculated in a similar manner as in e.g. Fraczkiewicz et al.[16]. We give the equations for completeness, the variables are explained in figure 3. Throughout the article we will use the symbol $\cdot$ to denote the dot product, symbol $\otimes$ to denote the inner vector product, and symbol $\times$ to denote the cross-product. Whenever the cross-product is used between a matrix and a vector, a column-wise cross-product is assumed.

The interface I_i is determined by two quantities: The amount of penetration g of S_i into S_l normalized to a unit sphere, and the interface type f_i, which is either positive for convex or negative for concave interfaces. g is determined by drawing a vector between the Cartesian coordinates $\boldsymbol{x}_l$ and $\boldsymbol{x}_i$ of spheres S_l and S_i respectively, and subsequent calculation of the plane of intersection.

$$\boldsymbol{v}_i = \boldsymbol{x}_i - \boldsymbol{x}_l \tag{4}$$

$$d_i = |\boldsymbol{v}_i| \tag{5}$$

$$g_i^* = \frac{d_i^2 + r_l^2 - r_i^2}{2 d_i r_l}| \tag{6}$$

$$f_i = \text{sign}(g_i^*) \tag{7}$$

$$g_i = |g_i^*| \tag{8}$$

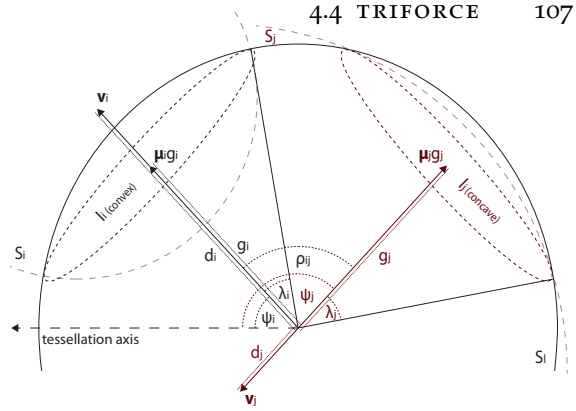


Figure 3: Sphere S_l intersects with two neighbor spheres S_i and S_j. The intersections result in one concave and one convex circular interface respectively, to which normalized vectors $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ point. Angles λ_i and λ_j represent the opening angle of the cones which are formed by the interfaces. ρ is the angle between their centers.

$$\boldsymbol{\mu}_i = f_i \frac{\boldsymbol{v}_i}{d_i} \tag{9}$$

$$\rho_{ij} = \arccos(\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j) \tag{10}$$

$$\lambda_i = \arccos(g_i) \tag{11}$$

Equations 4 to 11 represent a circular interface I_i on the surface of sphere S_l. The normal vector from the center of sphere S_l to the center of the interface I_i is denoted by $\boldsymbol{\mu}_i$, and the amount of penetration of one sphere into the other by g_i. Multiple circular interfaces might intersect, if the sum of their conical opening angles λ_i and λ_j fall below their radial distance ρ_{ij} and if one does not entirely contain the other.

The intersection between these circular interfaces will then result in two intersection-points $\boldsymbol{p}_{ij}$ and $\boldsymbol{p}_{ji}$ (see figures 1, 2, 4 and 5). The cases in which opening angles exactly match their radial distance, or two interfaces are identical, is treated as if there were no intersection at all. The circular interfaces can either be convex or concave, depending on whether g_i is positive or negative respectively.

The three look-up parameters are derived from these circular interfaces. Φ angles are calculated with respect to a certain interface (see figure 4); i.e. Φ angles for an intersection-point $\boldsymbol{p}_{ij}$ differ with respect to interface I_i or I_j. The curved arrows in

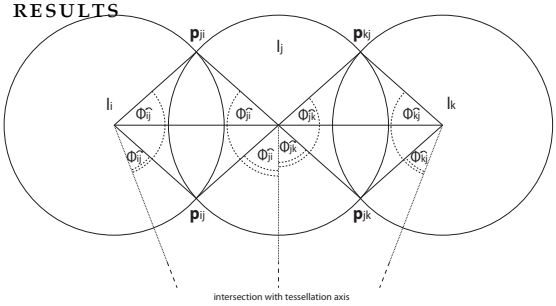


Figure 4: Intersections between interfaces I_i, I_j and I_k result in intersection-points $\boldsymbol{p}_{ij}$, $\boldsymbol{p}_{jk}$, $\boldsymbol{p}_{kj}$ and $\boldsymbol{p}_{ji}$. Φ angles are calculated with respect to their tessellation planes. For a certain intersection-point, Φ angles are different for both participating interfaces.

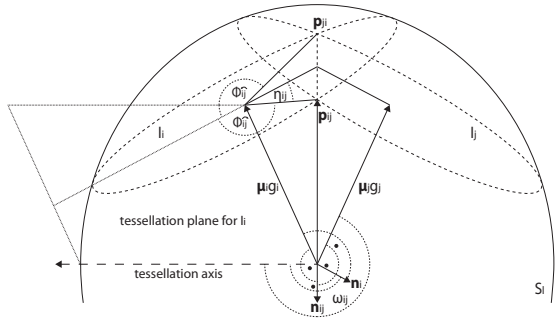


Figure 5: Intersections between circular interfaces I_i and I_j result in intersection-points $\boldsymbol{p}_{ij}$ and $\boldsymbol{p}_{ji}$. A tessellation plane is formed between $\boldsymbol{\mu}_j$ and the tessellation axis $\boldsymbol{\chi}$. ω_{ij} denotes the angle between the tessellation plane and the vector between the interfacial centers I_i and I_j, and is a combination of angles η_{ij} and ω_{ij}

equations 23 and 24 represent the direction of the intersection. Seen from interface I_i, intersection point $\boldsymbol{p}_{ij}$ is viewed as an outgoing point (connecting I_i with I_j), and $\boldsymbol{p}_{ji}$ as an incoming point (connecting I_j with I_i). The respective angles for interface I_i are an outgoing angle $\Phi_{ij}^{\curvearrowright}$ and an incoming angle $\Phi_{ji}^{\curvearrowleft}$ (see figure 5).

$$\psi_i = \arccos(\boldsymbol{\chi} \cdot \boldsymbol{\mu}_i) \tag{12}$$

$$\eta_{ij} = \arccos(\cot(\lambda_i)\cot(\rho_{ij}) - \cos(\lambda_j)\csc(\lambda_i)\csc(\rho_{ij})) \tag{13}$$

$$\boldsymbol{\nu}_i = \boldsymbol{\chi} \times \boldsymbol{\mu}_i \tag{14}$$

$$\boldsymbol{n}_i = \frac{\boldsymbol{\nu}_i}{|\boldsymbol{\nu}_i|} \tag{15}$$

$$\boldsymbol{\nu}_{ij} = \boldsymbol{\mu}_i \times \boldsymbol{\mu}_j \tag{16}$$

$$\boldsymbol{n}_{ij} = \frac{\boldsymbol{\nu}_{ij}}{|\boldsymbol{\nu}_{ij}|} \tag{17}$$

$$\varpi_{ij} = \arccos(\boldsymbol{n}_i \cdot \boldsymbol{n}_{ij}) \tag{18}$$

$$q_{ij} = -\text{sign}\left(\boldsymbol{n}_{ij} \cdot \boldsymbol{\chi}\right) \tag{19}$$

$$\omega_{ij} = q_{ij}\varpi_{ij} \tag{20}$$

$$q_{ij}^{\curvearrowright} = \pi\left(\text{sign}\left(\pi - (\omega_{ij} + \eta_{ij})\right) - 1\right) \tag{21}$$

$$q_{ij}^{\curvearrowleft} = -\pi\left(\text{sign}\left(-\pi - (\omega_{ij} - \eta_{ij})\right) - 1\right) \tag{22}$$

$$\Phi_{ij}^{\curvearrowright} = q_{ij}^{\curvearrowright} + \omega_{ij} + f_i f_j \eta_{ij} \tag{23}$$

$$\Phi_{ij}^{\curvearrowleft} = q_{ij}^{\curvearrowleft} + \omega_{ij} - f_i f_j \eta_{ij} \tag{24}$$

Equations 11 to 23 construct the parameters for the look-up table $T(\Phi, \psi, \lambda)$. Each sphere could possibly have multiple contributions to A^l in form of multiple discontinuous segments of its circular interface. Each segment can be uniquely identified by its corresponding intersection points $\boldsymbol{p}_{ij}$ and $\boldsymbol{p}_{jk}$ which denote the intersection between spheres S_i, S_j and S_k on the surface of sphere S_l whereas the solvent accessible surface area is on the right-hand side of the arcs containing this intersection point for convex interfaces, and on the left-hand side for concave interfaces. Segments containing $\boldsymbol{p}_{ij}$ belong to the boundary of the solvent accessible surface area, if, generally speaking, they are not covered by any other interface. The area of the triangular region A_{ijk}^l can now be given as:

$$q_{ijk} = \text{sign}\left(\Phi_{jk}^{\curvearrowright} - \Phi_{ji}^{\curvearrowleft}\right) \tag{25}$$

$$M_j = T(\pi, \psi_j, \lambda_j) \tag{26}$$

$$A^l_{ijk} = -f_j \left[M_j \, (q_{ijk} - 1) - T(f_j \, \Phi_{\widehat{jk}}, \psi_j, \lambda_j) + T(f_j \, \Phi_{\widehat{ji}}, \psi_j, \lambda_j) \right] \quad (27)$$

0.3 EXPOSED-BOUNDARY SEGMENTS

To calculate the three principle angles Φ, ψ and λ, for each intersection point of the boundary of the exposed area, it is necessary to find all segments (i, j, k) that contain these points. Each interface can possibly contain multiple segments for which the calculation is performed separately.
For each interface, a cyclic sorted list is established. In the course of the algorithm, the intersection points of all intersecting interfaces are added to the list. After each addition, the list is pruned and occluded intersection points removed. The pruning is performed by removing every node between two corresponding intersection points. Corresponding intersection points are either the two points of the newly added interface (double-dashed area in figure 6), or two points that pre-existed (single-dashed area in figure 6). At the conclusion of the algorithm, the list contains only intersection-points that are on the boundary of the exposed surface. The segments are then collected and their enclosing areas calculated.

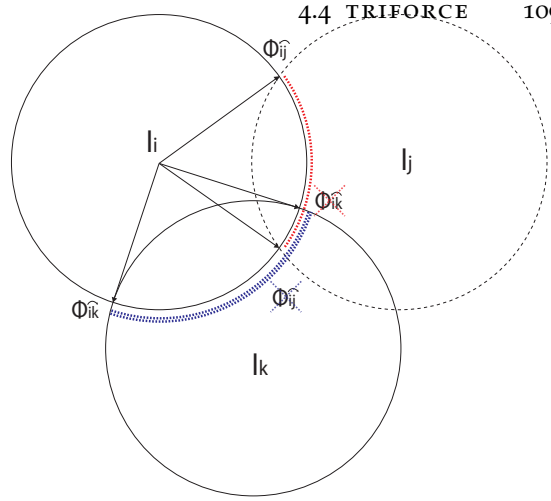


Figure 6: Interface I_k is added to the cyclic list of interface I_i, which already contains data from an intersection with I_j. Intersections $\Phi_{\widehat{ij}}$ and $\Phi_{\widehat{ij}}$ cause newly added intersection $\Phi_{\widehat{ik}}$ to be deleted (single-dashed area). Intersections $\Phi_{\widehat{ik}}$ and $\Phi_{\widehat{ik}}$ cause old intersection $\Phi_{\widehat{ij}}$ to be deleted (double-dashed area). Intersections that remain are $\Phi_{\widehat{ik}}$ and $\Phi_{\widehat{ij}}$.

0.4 INTEGRATION TABLES

Integration tables are computed for a predefined grid of Φ, ψ and λ values. ψ and λ are parametrizations which correspond to a virtual circular interface that has the two properties. The integration over these simplistic triangular surface areas is formally done by considering a vanishingly small patch Λ, and integrating over θ and φ.

$$A = \int_{\varphi_i}^{\varphi_j} \int_{0}^{\Theta(\varphi,\psi,\lambda)} d\psi d\theta \, \sin(\theta) \left| \frac{\partial \Lambda}{\partial \theta} \times \frac{\partial \Lambda}{\partial \psi} \right| \quad (28)$$

θ is the angle between the integration axis and the intersection of a radial line, drawn from the axis towards the interface, and its boundary. ϕ describes a rotation of aforementioned radial line around the integration axis, whereas a value of 0 corresponds to a radial line that intersects with the center of the interface. Consequently, integration limits for θ depend on the parametrization of the interface, as well as the rotation ϕ and is given by equation 29.

$$\Theta(\varphi, \psi, \lambda) = \arccos\left(\frac{\cos(\lambda)\cos(\psi) \pm \tau}{\kappa} \right) \quad (29)$$

with $\tau = \sqrt{\kappa - \cos(\psi)^2 \, (-\cos(\lambda)^2 + \kappa)}$ and $\kappa = \cos(\psi)^2 + \cos(\varphi)^2 \sin(\psi)^2$.

0.5 DERIVATIVES

Forces are calculated with respect to the change in the atomic coordinates and its effect on surface area. We have to take into account each atom $\boldsymbol{x}_l$ and the change of coordinates of intersecting atoms $\boldsymbol{x}^l_\delta$. For the sake of readability, we will omit index l where possible. We will denote the index set of interfaces of S_l by Int_l.

0.5.1 *Area*

$$\frac{\partial A}{\partial \boldsymbol{x}_\delta} = \sum_l \frac{\partial A^l}{\partial \boldsymbol{x}_\delta} \quad (30)$$

If $\delta \notin \mathrm{Int}_l$ or $\delta \neq l$, the derivative will be zero.

$$\frac{\partial A^l}{\partial \boldsymbol{x}_\delta} = r_l^2 \sum_{(ijk) \in \mathrm{Seg}_l} \frac{\partial A^l_{(ijk)}}{\partial \boldsymbol{x}_\delta} \quad (31)$$

If $\delta \notin \{i, j, k\}$ or $\delta \neq l$, the derivative will be zero.

$$\frac{\partial \mathcal{A}^{l}_{(ijk)}}{\partial \boldsymbol{x}_\delta} = -(q_{ijk} - 1)\frac{\partial M_j}{\partial x_\delta} + \left(\frac{\partial T(P_{jk})}{\partial \boldsymbol{x}_\delta} - \frac{\partial T(P_{ji})}{\partial \boldsymbol{x}_\delta}\right) \tag{32}$$

With $P_{jk} = (\Phi^{\curvearrowright}_{jk}, \psi_j, \lambda_j)$ and $P_{ji} = (\Phi^{\curvearrowright}_{ji}, \psi_j, \lambda_j)$.

0.5.2 *Integration tables*

$$\frac{\partial T(P_{ij})}{\partial \boldsymbol{x}_\delta} = \sum_{F} \frac{\partial T(P_{ij})}{\partial F}\frac{F}{\partial \boldsymbol{x}_\delta} \tag{33}$$

With $F = \{\Phi_{ij}, \psi_j, \lambda_j\}$. The derivatives of the integration table with respect to the look-up parameters, is itself looked-up and interpolated from 3 different look-up tables $G^{(\Phi)}$, $G^{(\psi)}$ and $G^{(\lambda)}$ for the respective derivatives.

$$\frac{\partial T(P_{ij})}{\partial F} = G^{(\partial F)}(P_{ij}) \tag{34}$$

Derivatives of the look-up parameters themselves with respect to atomic coordinates are calculated analytically.

0.5.3 *psi*

$$\frac{\partial \psi_a}{\partial \boldsymbol{x}_\delta} = \frac{\partial \psi_a}{\partial \boldsymbol{\mu}_a}\frac{\partial \boldsymbol{\mu}_a}{\partial \boldsymbol{x}_\delta} \tag{35}$$

$$\frac{\partial \psi_a}{\partial \boldsymbol{\mu}_a} = -\frac{\boldsymbol{\chi}}{\sqrt{1 - (\boldsymbol{\mu}_a \cdot \boldsymbol{\chi})^2}} \tag{36}$$

$$\frac{\partial \boldsymbol{\mu}_a}{\partial \boldsymbol{x}_a} = \frac{1}{d_a}(\boldsymbol{I} - \boldsymbol{\mu}_a \otimes \boldsymbol{\mu}_a)^{\intercal} \tag{37}$$

$$\frac{\partial \boldsymbol{\mu}_a}{\partial \boldsymbol{x}_l} = -\frac{1}{d_a}(\boldsymbol{I} - \boldsymbol{\mu}_a \otimes \boldsymbol{\mu}_a)^{\intercal} \tag{38}$$

0.5.4 *lambda*

$$\frac{\partial \lambda_a}{\partial \boldsymbol{x}_\delta} = \frac{\partial \lambda_a}{\partial g_a}\frac{\partial g_a}{\partial \boldsymbol{x}_\delta} \tag{39}$$

$$\frac{\partial \lambda_a}{\partial g_a} = -\frac{1}{\sqrt{1 - g_a^2}} \tag{40}$$

$$\frac{\partial g_a}{\partial \boldsymbol{x}_\delta} = \frac{\partial g_a}{\partial d_a}\frac{\partial d_a}{\partial \boldsymbol{x}_\delta} \tag{41}$$

$$\frac{\partial g_a}{\partial d_a} = -\frac{g_a}{d_a} + \frac{f_a}{r_l} \tag{42}$$

$$\frac{\partial d_a}{\partial \boldsymbol{x}_a} = \boldsymbol{\mu}_a \tag{43}$$

$$\frac{\partial d_a}{\partial \boldsymbol{x}_l} = -\boldsymbol{\mu}_a \tag{44}$$

0.5.5 *PHI*

$$\frac{\partial \Phi^{\curvearrowright}_{ab}}{\partial \boldsymbol{x}_\delta} = \frac{\partial \eta_{ab}}{\partial \boldsymbol{x}_\delta} + \frac{\partial \omega_{ab}}{\partial \boldsymbol{x}_\delta} \tag{45}$$

$$\frac{\partial \Phi^{\curvearrowright}_{ab}}{\partial \boldsymbol{x}_\delta} = -\frac{\partial \eta_{ab}}{\partial \boldsymbol{x}_\delta} + \frac{\partial \omega_{ab}}{\partial \boldsymbol{x}_\delta} \tag{46}$$

eta

$$\frac{\partial \eta_{ab}}{\partial \boldsymbol{x}_\delta} = \frac{\partial \eta_{ab}}{\partial \lambda_a}\frac{\partial \lambda_a}{\partial \boldsymbol{x}_\delta} + \frac{\partial \eta_{ab}}{\partial \lambda_b}\frac{\partial \lambda_b}{\partial \boldsymbol{x}_\delta} + \frac{\partial \eta_{ab}}{\partial \rho_{ab}}\frac{\partial \rho_{ab}}{\partial \boldsymbol{x}_\delta} \tag{47}$$

Let $\rho_{ab} = \cot(\lambda_a)\cot(\rho_{ab})$, $\sigma_{ab} = \cos(\lambda_b)\csc(\lambda_a)\csc(\rho_{ab})$ and $\sigma_{ab} = \rho_{ab} - \sigma_{ab}$

$$\frac{\partial \eta_{ab}}{\partial \lambda_a} = \left[\frac{\csc(\lambda_a)\rho_{ab}}{\cos(\lambda_a)} - \sigma_{ab}\tan(\lambda_a)\right] / \sqrt{1 - \sigma_{ab}^2} \tag{48}$$

$$\frac{\partial \eta_{ab}}{\partial \lambda_b} = -\left[\sigma_{ab}\tan(\lambda_b)\right] / \sqrt{1 - \sigma_{ab}^2} \tag{49}$$

$$\frac{\partial \eta_{ab}}{\partial \rho_{ab}} = \left[\frac{\csc(\rho_{ab})\rho_{ab}}{\cos(\rho_{ab})} - \sigma_{ab}\tan(\rho_{ab})\right] / \sqrt{1 - \sigma_{ab}^2} \tag{50}$$

rho

The following identity applies: $\rho_{ab} = \rho_{ba}$

$$\frac{\partial \rho_{ab}}{\partial \boldsymbol{x}_a} = \frac{\partial \rho_{ab}}{\partial \boldsymbol{\mu}_a}\frac{\partial \boldsymbol{\mu}_a}{\partial \boldsymbol{x}_a} \tag{51}$$

$$\frac{\partial \rho_{ab}}{\partial \boldsymbol{\mu}_a} = -\frac{\boldsymbol{\mu}_b}{\sqrt{1 - (\boldsymbol{\mu}_a \cdot \boldsymbol{\mu}_b)^2}} \tag{52}$$

omega

$$\frac{\partial \omega_{ab}}{\partial \boldsymbol{x}_\delta} = \frac{\partial \omega_{ab}}{\partial \varpi_{ab}}\frac{\partial \varpi_{ab}}{\partial \boldsymbol{x}_\delta} \tag{53}$$

$$\frac{\partial \omega_{ab}}{\partial \varpi_{ab}} = q_{ab} \tag{54}$$

$$\frac{\partial \varpi_{ab}}{\partial \boldsymbol{x}_\delta} = \frac{\varpi_{ab}}{\partial \boldsymbol{n}_a}\frac{\partial \boldsymbol{n}_a}{\partial \boldsymbol{x}_\delta} + \frac{\partial \varpi_{ab}}{\partial \boldsymbol{n}_{ab}}\frac{\partial \boldsymbol{n}_{ab}}{\partial \boldsymbol{x}_\delta} \tag{55}$$

$$\frac{\partial \varpi_{ab}}{\partial \boldsymbol{n}_a} = \frac{\boldsymbol{n}_{ab}}{\sqrt{1-(\boldsymbol{n}_a \cdot \boldsymbol{n}_{ab})^2}} \tag{56}$$

$$\frac{\partial \varpi_{ab}}{\partial \boldsymbol{n}_{ab}} = \frac{\boldsymbol{n}_i}{\sqrt{1-(\boldsymbol{n}_a \cdot \boldsymbol{n}_{ab})^2}} \tag{57}$$

$$\frac{\partial \boldsymbol{n}_a}{\partial \boldsymbol{x}_\delta} = \frac{\partial \boldsymbol{n}_a}{\partial \boldsymbol{\nu}_a}\frac{\partial \boldsymbol{\nu}_a}{\partial \boldsymbol{\mu}_a}\frac{\partial \boldsymbol{\mu}_a}{\partial \boldsymbol{x}_\delta} \tag{58}$$

$$\frac{\partial \boldsymbol{n}_a}{\partial \boldsymbol{\nu}_a} = -\frac{1}{d\nu_a}(\boldsymbol{I} - \boldsymbol{n}_a \otimes \boldsymbol{n}_a)^\intercal \tag{59}$$

$$\frac{\partial \boldsymbol{\nu}_a}{\partial \boldsymbol{\mu}_a} = \boldsymbol{I} \times \boldsymbol{\chi} \tag{60}$$

$$\frac{\partial \boldsymbol{n}_{ab}}{\partial \boldsymbol{x}_\delta} = \frac{\partial \boldsymbol{n}_{ab}}{\partial \boldsymbol{\nu}_{ab}}\frac{\partial \boldsymbol{\nu}_{ab}}{\partial \boldsymbol{\mu}_a}\frac{\partial \boldsymbol{\mu}_a}{\partial \boldsymbol{x}_\delta} + \frac{\partial \boldsymbol{n}_{ab}}{\partial \boldsymbol{\nu}_{ab}}\frac{\partial \boldsymbol{\nu}_{ab}}{\partial \boldsymbol{\mu}_b}\frac{\partial \boldsymbol{\mu}_b}{\partial \boldsymbol{x}_\delta} \tag{61}$$

$$\frac{\partial \boldsymbol{n}_{ab}}{\partial \boldsymbol{\nu}_{ab}} = -\frac{1}{d\nu_{ab}}(\boldsymbol{I} - \boldsymbol{n}_{ab} \otimes \boldsymbol{n}_{ab})^\intercal \tag{62}$$

$$\frac{\partial \boldsymbol{\nu}_{ab}}{\partial \boldsymbol{\mu}_a} = \boldsymbol{I} \times \boldsymbol{\mu}_b \tag{63}$$

$$\frac{\partial \boldsymbol{\nu}_{ab}}{\partial \boldsymbol{\mu}_b} = -\boldsymbol{I} \times \boldsymbol{\mu}_a \tag{64}$$

0.6 CONTACT POINT FORCE DISCONTINUITY

Let us imagine the following scenario: Two non-intersecting atoms S_l and S_i approach one another. In the moment when the hulls just touch each other, i.e. when $g_i = 1.0$, the derivative in the direction of the axis of separation abruptly jumps from zero to some non-zero value (see figure 7). This discontinuity will cause insuperable barriers during a minimization, or lead to a heat up in molecular dynamics simulations[24]. To compensate for these undesired artifacts, the force and area around the contact point is smoothed with a logistic function $\Gamma(x)$ which is dependent on the distance between the spheres as shown in figure 7 and equation 65.

$$\widetilde{A}^l_{(ijk)} = \Gamma(t_j) A^l_{(ijk)} \tag{65}$$

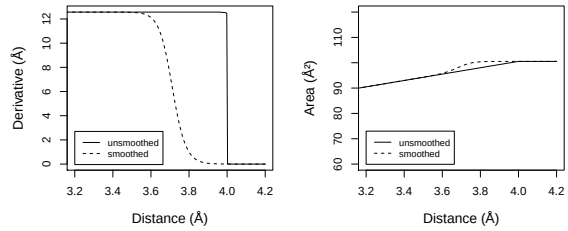


Figure 7: When two spheres (in this example with radii of 2Å) meet, derivatives jump from zero to some non-zero value. By using a distance dependent logistic smoother, this discontinuity can be smoothed in force-space, which will add a small hill in area-space.

$$t_j = p_0\left(1 - \frac{d_j}{r_j + r_l}\right) \tag{66}$$

$$\Gamma(x) = \frac{1}{1 + \exp(-2xp_1 - p_1)} \tag{67}$$

p_0 and p_1 are parameters used to adjust the shape of the smoothing function. In our implementation they were set to $p_0 = 0.15$ and $p_1 = 8.0$. The smoothing has impact on all derivatives that depend on interface I_j.

$$\frac{\partial \widetilde{A}^l_{(ijk)}}{\partial \boldsymbol{x}_\delta} = \Gamma(t_j)\frac{\partial A^l_{(ijk)}}{\partial \boldsymbol{x}_\delta} + \frac{\partial \Gamma(t_j)}{\partial \boldsymbol{x}_\delta}A^l_{(ijk)} \tag{68}$$

$$\frac{\partial \Gamma(t_j)}{\partial \boldsymbol{x}_\delta} = \frac{\partial \Gamma(t_j)}{\partial \boldsymbol{t}_j}\frac{\partial t_j}{\partial \boldsymbol{x}_\delta} \tag{69}$$

$$\frac{\partial \Gamma(t_j)}{\partial \boldsymbol{t}_j} = \frac{p_1}{1 + \cosh(p_1 - 2xp_1)} \tag{70}$$

$$\frac{\partial t_j}{\partial \boldsymbol{x}_j} = -\frac{\boldsymbol{\mu}_j p_0}{r_j + r_l} \tag{71}$$

$$\frac{\partial t_j}{\partial \boldsymbol{x}_l} = -\frac{\partial t_j}{\partial \boldsymbol{x}_j} \tag{72}$$

The smoothing adds a small hill in area-space, but manages to overcome the discontinuous gap in force-space with a smooth bridge.

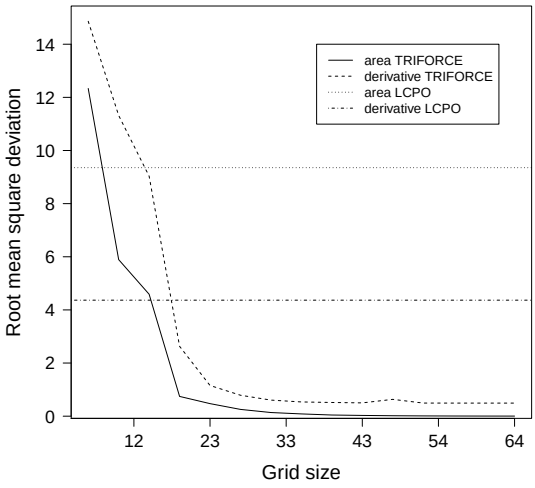


Figure 8: Atom-wise comparison for areas and derivatives has been performed on a set of 42 proteins and various grid-sizes which increase with a factor of 2.

Table 1: Protein set

PDB code	Heavy atoms	TRIFORCE RMSD area (Å^2)	TRIFORCE RMSD grad (Å)	LCPO RMSD area (Å^2)	LCPO RMSD grad (Å)
1PLX	40	0.003	0.011	15.860	3.026
1CBH	260	0.002	0.258	7.642	3.941
1SP2	269	0.003	0.199	11.098	4.374
5RXN	422	0.003	0.022	9.694	4.330
1I6F	436	0.003	0.015	11.343	4.475
4PTI	454	0.002	0.330	9.076	4.340
1FAS	468	0.002	0.067	9.554	4.532
1SN3	492	0.002	0.671	11.560	4.544
1CSP	505	0.003	1.345	10.732	4.427
2CRO	520	0.002	0.262	9.602	4.269
1FVQ	545	0.002	0.043	8.225	4.376
1SDF	550	0.003	0.059	10.005	4.484
1UBQ	602	0.003	0.088	10.175	4.540
1HIP	617	0.002	0.145	9.005	4.209
1PHT	666	0.002	2.160	8.669	4.375
1J5D	721	0.002	0.388	8.973	4.690
2CDV	801	0.003	0.443	12.592	5.281
1OPC	805	0.002	0.140	8.093	4.274
1KTE	818	0.002	0.095	9.484	4.458
1NSO	858	0.003	0.164	9.196	4.389
2PAZ	932	0.002	1.414	10.273	4.612
1CHN	966	0.002	0.109	9.570	4.467

continued in table 2

0.7 RESULTS

0.7.1 *Correctness of areas and derivatives*

The correctness of the area and derivatives is shown by comparing to exact analytical values for a set of proteins and integration tables of different grid sizes. We compared against the output of http://curie.utmb.edu/getarea.html. For the set of proteins we chose 6 structures of increasing size that were mentioned in reference[16], 16 structures from Decoys 'R' Us that proved challenging for other area calculation algorithms like `LCPO`, 30 structures from reference[25] that represent a set of different folds, and ubiquitin, because of its prevalence. The list of structures is shown in table 2.

0.7.2 *Minimization*

A minimization was performed on the fluorene molecule to test correctness of the derivatives in a dynamic simulation. No repulsive force was added, as such the simulation was expected to minimize the molecule into a single sphere, which is what was observable. We show in figure 9 the course of the simulation. No change is observed after around 51 steps, in which the system has nearly completely collapsed into the inflated sphere (VdW + water distance) of one singe carbon with radius of 3.09Å. The simulation ends in a local minimum, which is why we do not observe the complete collapse.

0.8 DISCUSSION

TRIFORCE has several features which distinguish it from algorithms that perform equivalent tasks. It has the ability to segment the exposed surface efficiently into triangular patches. This ability can not just be exploited to accumulate precomputed areas, but indeed may be used for any precomputed quantity. For such a purpose, the algorithm seamlessly allows for two adjustments: The selection of an arbitrary tessellation axis χ, to adjust for geometrical requirements, and the option to add additional dimensions to the integration tables, to allow the modeling of a functional surface with higher complexity. In contrast to some algorithms that have to rely on united atom approaches, our algorithm is capable of handling all topologies, including concave circular interfaces. This is crucial because such interfaces frequently occur in molecules, especially in the case when hydrogens are buried in the large Van der Waals radii of their bonded atoms.
Its tabular integration procedure can be efficiently implemented onto GPUs (development is in

continued from table 1

PDB code	Heavy atoms	TRIFORCE RMSD area (Å^2)	TRIFORCE RMSD grad (Å)	LCPO RMSD area (Å^2)	LCPO RMSD grad (Å)
1K40	976	0.002	0.378	9.600	4.399
1OOI	986	0.002	0.049	9.290	4.519
1PDO	988	0.002	0.666	8.284	4.331
6LYZ	1001	0.002	0.081	8.191	4.384
1LIT	1045	0.002	0.204	8.644	4.036
1BJ7	1208	0.002	0.159	8.604	4.325
2I1B	1219	0.002	0.268	9.878	4.501
1MBS	1223	0.002	0.093	11.019	4.879
1EMR	1232	0.002	0.167	9.382	4.679
1CZT	1311	0.002	0.204	9.521	4.195
2PTN	1629	0.002	0.278	8.717	4.150
5PAD	1655	0.002	1.279	7.687	4.257
1SUR	1739	0.002	0.287	8.093	4.501
2HVM	2087	0.002	0.263	7.924	4.091
2CYP	2299	0.002	0.221	7.992	4.220
1RHD	2319	0.002	0.207	7.814	4.458
2TMN	2432	0.002	0.484	6.911	4.124
2TS1	2457	0.002	0.115	8.683	4.438
1FRG	3361	0.002	0.166	8.251	4.232
1MCP	3401	0.002	0.102	7.870	4.288
Average		0.002	0.336	9.352	4.367

Table 2: Comparison between analytical and TRIFORCE surface areas. The root mean square error of the areas and their derivatives are given for a set of 42 proteins.

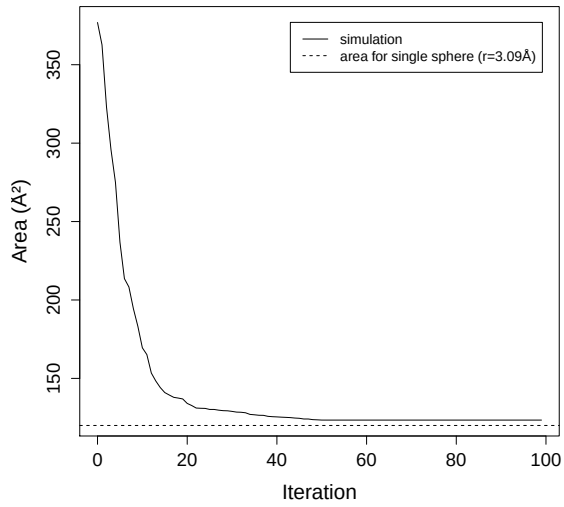


Figure 9: The molecule fluorene has been subject to a minimization of its solvent accessible surface area. No repulsive force was used and in consequence the molecule minimized into a single sphere of approximately radius 3.09 Å, which corresponds to the radius of the largest sphere used in the simulation.

progress) in which this task becomes simple texture look-up. The whole algorithm has been designed to allow for parallel processing on the level of circular interfaces, in contrast to an atomic level. Although not implemented, this should utilize the GPUs to full capacity. Integration tables are stored in an architecture independent way, enabling usage of the library on different platforms right away.
All calculations can be made arbitrarily precise by increasing the grid size, as figure 8 shows. Increasing the grid-size will result in additional memory demand, but not in a decrease of speed (except for the initial loading procedure). TRIFORCE can be accessed as a web-service from http://lavandula.imim.es/triforce, where also a downloadable version of the sources are available, free of charge under the GPL license.

0.9 CONCLUSION

We have presented a novel algorithm for the calculation of the solvent accessible surface area and its derivatives for arbitrary molecules. Although the method is partly numerical, errors in both quantities are marginal. Its correctness has been tested through an extensive test of a database of 42 proteins of various sizes and folds, and its robustness through a BFGS-minimization, which managed to area-minimize the fluorene molecule into a single ball. To compensate for discontinuities, which occur due to the discrete nature of spherical objects in a three dimensional space, we have introduced a logistical smoothing function which is able to bridge the gap in force-space between two distant spheres and their intersecting counterparts. Future improvement focuses on the parallelization of the algorithm, by its implementation on GPU architecture.

0.10 ACKNOWLEDGMENTS

We thank Montserrat Corbera for fruitful discussions and proof-reading of the manuscript.

[1] H. H. Uhlig. The solubilities of gases and surface tension. *J. Phys. Chem*, 41(9):1215–1226, 1937.

[2] R. B. Hermann. Theory of hydrophobic bonding. ii. the correlation of hydrocarbon solubility in water with solvent cavity surface area. *J. Phys. Chem.*, 76(19):2754–2759, 1972.

[3] Anthony Nicholls, Kim A. Sharp, and Barry Honig. Protein folding and association: Insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins: Struct., Funct., Genet.*, 11:281–296, 1991.

[4] E. Gallicchio, M. M. Kubo, and R. M. Levy. Enthalpy-entropy and cavity decomposition of alkane hydration free energies: Numerical results and implications for theories of hydrophobic solvation. *J. Phys. Chem. B*, 104(26): 6271–6285, 2000.

[5] D. Eisenberg and A. D. McLachlan. Solvation energy in protein folding and binding. *Nature*, 319:199–203, 1986.

[6] Gregory D. Hawkins, Christopher J. Cramer, and Donald G. Truhlar. Parametrized models of aqueous free energies of solvation based on pairwise descreening of solute atomic charges from a dielectric medium. *J. Phys. Chem.*, 100(51):19824–19839, January 1996. doi: 10.1021/jp961710n. URL http://dx.doi.org/10.1021/jp961710n.

[7] Gregory D. Hawkins, Christopher J. Cramer, and Donald G. Truhlar. Parametrized model for aqueous free energies of solvation using geometry-dependent atomic surface tensions with implicit electrostatics. *J. Phys. Chem. B*, 101(36):7147–7157, 1997.

[8] Junmei Wang, Wei Wang, Shuanghong Huo, Matthew Lee, and Peter A. Kollman. Solvation model based on weighted solvent accessible surface area. *J. Phys. Chem. B*, 105(21):5055–5067, 2001.

[9] E. Gallicchio, L. Y. Zhang, and R. M. Levy. 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.*, 23(5): 517–529, 2002.

[10] J. A. Wagoner and N. A. Baker. Assessing implicit models for nonpolar mean solvation forces: The importance of dispersion and volume terms. *Proc. Natl. Acad. Sci. USA*, 103(22):8331–8336, 2006.

[11] C. Tan, Y.-H. Tan, and R. Luo. Implicit nonpolar solvent models. *J. Phys. Chem. B*, 111:12263–12274, 2007.

[12] Emilio Gallicchio, Kristina Paris, and Ronald M. Levy. The agbnp2 implicit solvation model. *J. Chem. Theory Comput.*, 5(9):2544–2564, 2009.

[13] Christopher J. Fennell and Ken A. Dill. Oil/Water transfer is partly driven by molecular shape, not just size. *J. Am. Chem. Soc.*, 132(1):234–240, 2010. doi: 10.1021/ja906399e. URL http://pubs.acs.org/doi/abs/10.1021/ja906399e.

[14] C. J. Fennell, C. W. Kehoe, and K. A. Dill. Modeling aqueous solvation with semi-explicit assembly. *Proc. Natl. Acad. Sci. USA*, 108(8):3234–3239, 2011.

[15] G. Perrot, B. Cheng, K. D. Gibson, J. Vila, K. A. Palmer, A. Nayeem, B. Maigret, and H. A. Scheraga. Mseed: A program for the rapid analytical determination of accessible surface areas and their derivatives. *J. Comput. Chem.*, 13(1):1–11, 1992.

[16] Robert Fraczkiewicz and Werner Braun. Exact and efficient analytical calculation of the accessible surface areas and their gradients for macromolecules. *J. Comput. Chem.*, 19(3):319–333, 1998.

[17] Shura Hayryan, Chin-Kun Hu, Jaroslav Skřivánek, Edik Hayryan, and Imrich Pokorný. A new analytical method for computing solvent-accessible surface area of macromolecules and its gradients. *J. Comput. Chem.*, 26(4):334–343, 2005.

[18] Konstantin V. Klenin, Frank Tristram, Timo Strunk, and Wolfgang Wenzel. Derivatives of molecular surface area and volume: Simple and exact analytical formulas. *J. Comput. Chem.*, 32(12):2647–2653, 2011.

[19] S. J. Wodak and J. Janin. Analytical approximation to the accessible surface area of proteins. *Proc. Natl. Acad. Sci. USA*, 77(4):1736–1740, 1980.

[20] S. Sridharan, Anthony Nicholls, and Kim A. Sharp. A rapid method for calculating derivatives of solvent accessible surface areas of molecules. *J. Comput. Chem.*, 16(8): 1038–1044, 1995.

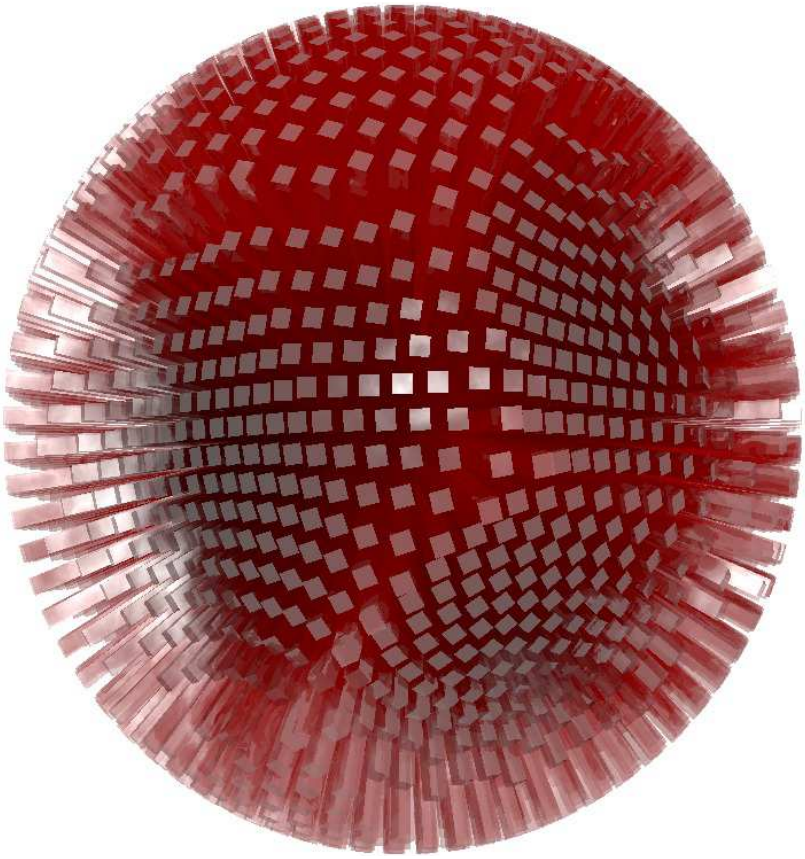
[21] Jörg Weiser, Peter S. Shenkin, and W. Clark Still. Approximate atomic surfaces from linear combinations of pairwise overlaps (LCPO). *J. Comput. Chem.*, 20(2):217–230, 1999.

[22] Georgy Rychkov and Michael Petukhov. Joint neighbors approximation of macromolecular solvent accessible surface area. *J. Comput. Chem.*, 28(12):1974–1989, 2007.

[23] B. Lee and R. M. Richards. The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Biol.*, 55(3):379–400, 1971.

[24] Peter J. Steinbach and Bernhard R. Brooks. New Spherical-Cutoff methods for Long-Range forces in macromolecular simulation. *Journal of Computational Chemistry*, 15(7), 1994.

[25] M. Rueda, C. Ferrer-Costa, T. Meyer, A. Perez, J. Camps, A. Hospital, J. L. Gelpi, and M. Orozco. A consensus view of protein dynamics. *Proceedings of the National Academy of Science*, 104:796–801, January 2007. doi: 10.1073/pnas.0605534104.



Artistic (and not realistic) representation of a spherical Depth-Buffer

4.5 A MULTI-LAYERED DEPTH-BUFFER FOR THE DETECTION OF ATOMS CONTRIBUTING TO THE BOUNDARIES OF EXPOSED SURFACE AREAS

Nils J.D. Drechsel, Christopher J. Fennell, Ken A. Dill, and Jordi Villà-Freixa

"A Multi-Layered Depth-Buffer for the Detection of Atoms Contributing to the Boundaries of Exposed Surface Areas"

Submitted to Journal of Chemical Theory and Computation

A Multi-Layered Depth-Buffer for the Detection of Atoms Contributing to the Boundaries of Exposed Surface Areas

Nils J. D. Drechsel[1,2], Christopher J. Fennell[2], Ken A. Dill[2], and Jordi Villà–Freixa[3]

[1]*Computational Biochemistry and Biophysics Laboratory, Universitat Pompeu Fabra, Research Unit on Biomedical Informatics, C/Doctor Aiguader, 88, 08003 Barcelona, Catalunya, Spain*

[2]*Laufer Center for Physical and Quantitative Biology, Stony Brook University, Stony Brook, New York 11794-5252, USA*

[3]*Escola Politècnica Superior, Universitat de Vic, C/ de la Laura, 13, 08500 Vic, Catalunya, Spain*

Algorithms which speed up solvent accessible surface area algorithms based on Gauss-Bonnet paths, remove buried atoms that do not contribute to the exposed areas. These algorithms are rare, because identification of such atoms is a non-trivial task. We propose here another kind of algorithm, simple and efficient, that removes many more atoms and is still able to calculate a path within arbitrary precision. The algorithm is based on the idea of a Multi-Layered Depth-Buffer that efficiently stores coverage information of where an atom is occluded and where it is exposed, in a way that takes into account the complicated shape of the exposed area. A main feature is the removal of non-buried, redundant atoms that are not a necessity to the formation of such paths. Removal of these atoms greatly improves performance, but creates additional paths, which are subsequently probabilistically removed.

Solvent accessible surface areas play a crucial role in implicit solvation methods[1–4]. This general type of algorithms approximate all-atom solvation free energies in order to, amongst other reasons, decrease computational demand. It is therefore imperative to calculate areas rapidly, since failure in doing so would undermine their advantage. Exposed area algorithms are usually inter-convertible, which gave rise to a whole continuum of methods. The optimal algorithm would be both precise and fast. Since these objectives however are in competition, many different algorithms have been developed that favor one objective more than the other.

The continuum of algorithms can be coarsely partitioned into arbitrarily exact, and approximate methods. Amongst the exact we find purely numerical methods[5–10], methods relying on Gauss-Bonnet paths[11–18], and methods utilizing alpha-shape theory[19]. Under approximate methods fall statistical and probabilistic methods[20–22] as well as heuristical methods[23]. Numerical methods generally provide none or finite difference derivatives at best, causing speed barriers in molecular dynamics with implicit solvent calculations. Methods utilizing Gauss-Bonnet paths are arguably the most accurate, but the detection of the paths is computationally demanding and poses a bottleneck in the overall calculation.

In molecular surface area calculations, atoms are usually modelled as spheres with radii corresponding to their Van der Waals radii plus the average distance to the solvent. A Gauss-Bonnet path is then a sequence of spherical arcs, each of which originate from an intersection-point of two neighbour spheres on the surface of a sphere of interest. These arcs are connected and form a cycle which can be used for evaluation of the Gauss-Bonnet theorem[12], or exploited with other approaches[17;18]. The intrinsic problem of calculating the paths lies within the quadratic amount of potential intersection-points. Any neighbour sphere can possibly intersect with any other neighbour sphere on the surface of the sphere of interest. Any algorithm that is set out to identify which intersection-points belong to the path and which not, needs to process also a great amount of atoms which are buried inside the molecule, their intersections always non Gauss-Bonnet and their calculation superfluous.

Several methods have been introduced to speed-up calculation of the intersection-points by removing aforementioned buried atoms. Fraczkiewicz et al.[16] introduced a method that relies on the partition of

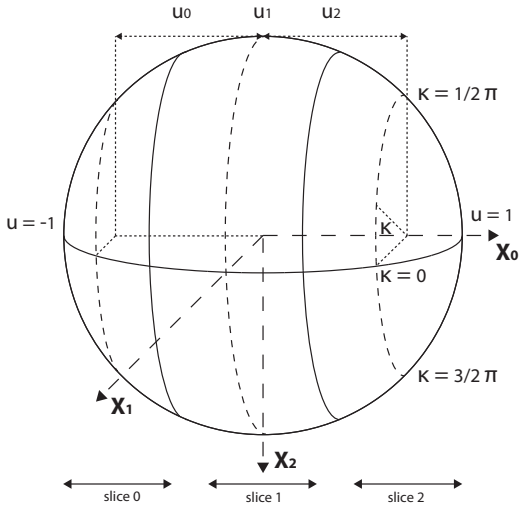


Figure 1: A sphere is cut into slices of equal width. Each slice is referenced by its distance u to the center along axis χ_0 and its angle κ to the (χ_0, χ_1)-plane.

the molecular space into half-spaces. Gibson et al.[24] described a method to reduce the amount of higher order intersections, which is a similar approach to Weiser et al.[25] who are pruning buried atoms from the neighbour list, which are occluded by 3 or 4 neighbour spheres. These algorithms are exact in the sense that they do not remove any exposed atoms from the lists. They use geometrical considerations to base their decisions. A more statistical approach was proposed by the same authors[26] which is an advancement of the idea of Stouten et al.[27]. In this approach, the neighbour density in four tetrahedral directions is calculated. If the density falls above a certain precomputed limit, the atom is considered buried and removed. The statistical nature give cause to a small error whenever exposed atoms are deemed buried and subsequently removed.

We present here another approach to speed up computation of Gauss-Bonnet paths. Our method does not only remove buried atoms, but instead removes all atoms that do not actively form the path, i.e. their intersections do not produce intersection-points that belong to a Gauss-Bonnet path. This methodology removes more spheres than any of the aforementioned algorithms, with the introduction of a small error due to its numerical nature. The error however is adjustable, dependent on the amount of speed-up which is desired.

0.2 METHOD

For each atom in a molecule, a special data structure called Multi-Layered Depth-Buffer is created. The molecule is modelled as an aggregation of spheres, which are defined by the Cartesian coordinates of the atoms and their extended Van der Walls radii. A sphere, denoted by S_l, is potentially intersected by a number of neighbour spheres S_i. The area that S_i covers on S_l through their intersection is termed buried. The complement of all covered areas, i.e. the area which is not covered by any neighbour sphere, is termed the exposed area. Neighbour spheres that adjoin to the exposed area are termed boundary spheres. They are those spheres that the Depth-Buffer attempts to detect. The intersection between those spheres and sphere S_l, is called the exposed boundary, and it encloses the exposed area. The rest of the spheres are called internal spheres, or buried spheres in case they have no own exposed area, i.e. are completely buried by their neighbour spheres.

For a sphere S_l, all neighbour spheres are sequentially added to the buffer. Once all spheres are added, the buffer has adopted its final shape and will be able to distinguish boundary spheres from internal and buried spheres. The spheres are then checked for either case, and removed if they are found to be non-boundary. The removal of internal spheres causes the creation of additional false exposed boundaries. In a third step, these boundaries are validated, and removed if found to be invalid.

The structure of the Multi-Layered Depth-Buffer is created by cutting up a unit sphere into slices of equal width as depicted in figure 1. We will refer to the number of slices as the resolution of the buffer. Three orientation axes (χ_0, χ_1 and χ_2), are arbitrarily chosen, such that they are orthogonal to each other and all slices are orthogonal to χ_0. To simplify some of the equations, in this article these axes are identical with canonical unit vectors.

A radial line is drawn where a plane intersects the exposed surface of the slice midways. The distance of the plane to the center of the sphere is denoted by u, and can take on positive or negative values in the interval $[-1, 1]$, depending on its orientation to χ_0 (see figure 1). All points located on these lines can be exactly described by the Depth-Buffer, all points in-between the lines only approximately. A point on a line with distance u is identified by its angle κ to the (χ_0, χ_1)-plane, as depicted in 1.

All points in the Depth-Buffer can be unambiguously identified by their κ and u values. To increase accuracy of the method, more Depth-Buffers, theo-

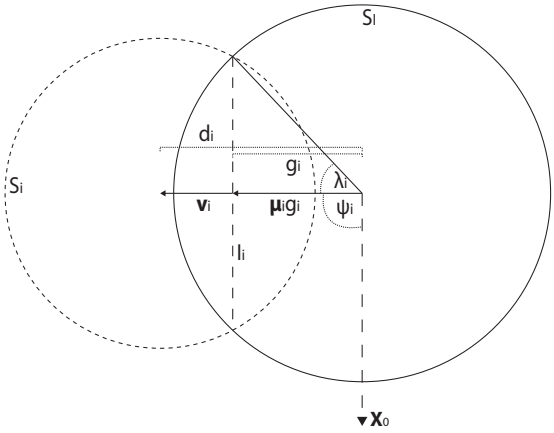


Figure 2: An interface I_i between two spheres S_l and S_i is created through their plane of intersection. It is parametrized by the distance g_i from the center of S_l to the plane and its orientation $\boldsymbol{\mu}_i$. Its angle to $\boldsymbol{\chi}_0$ is denoted by ψ_i and its opening angle by λ_i.

retically an infinite number, can be included. However, the gain in accuracy must be bought with higher computational demand. For this reason, we describe here only the usage of two Depth-Buffers, with the second buffer orthogonal to the first, which will be called B^0 and B^1 respectively. Whenever mathematically both are equivalent, we will omit their indexes.

A sphere S_i is added to the Depth-Buffer in three steps: First, its principle angles are determined. Second, its depth information is looked up from a precomputed table using aforementioned principle angles. Third, the buffer is updated with the information from the table. The principle angles ψ, λ and κ are derived by calculation of the interface I_i between S_l and S_i (see figure 2). Throughout this article, the symbol $\times$ will refer to the cross, and $\cdot$ to the scalar product.

An interface I_i is determined with the following: First, a vector $\boldsymbol{v}_i$ is drawn between Cartesian coordinates $\boldsymbol{x}_l$ and $\boldsymbol{x}_i$ of spheres S_l and S_i.

$$\boldsymbol{v}_i = \boldsymbol{x}_i - \boldsymbol{x}_l \tag{1}$$

The amount of intrusion g_i^* of S_i into S_l can be determined using their distance d_i and their radii r_l and r_i.

$$d_i = |\boldsymbol{v}_i| \tag{2}$$

$$g_i^* = \frac{d_i^2 + r_l^2 - r_i^2}{2 d_i r_l} \tag{3}$$

We are only interested in the location of I_i, which is the plane of intersection where the hulls of S_l and S_i touch. For this purpose, a vector $\boldsymbol{\mu}_i$ is created that points into the direction of I_i.

$$f_i = \operatorname{sign}(g_i^*) \tag{4}$$

$$g_i = |g_i^*| \tag{5}$$

$$\boldsymbol{\mu}_i = f_i \frac{\boldsymbol{v}_i}{d_i} \tag{6}$$

Multiplication by the normalized g_i yields a vector to the interface. The angle between $\boldsymbol{\chi}_0$ and $\boldsymbol{\mu}_i$ gives the first principle angle ψ_i.

$$\psi_i = \arccos(\boldsymbol{\chi}_0 \cdot \boldsymbol{\mu}_i) \tag{7}$$

The second principle angle λ_i is determined as if I_i was part of a cone originating from the center of S_l. λ_i is then the opening angle of that cone.

$$\lambda_i = \arccos(g_i) \tag{8}$$

For the third principle angle $\bar{\kappa}$, the rotation of S_i around $\boldsymbol{\chi}_0$ has to be determined. We start by creating a vector $\boldsymbol{n}_i$ which resides simultaneously in the $(\boldsymbol{\chi}_1, \boldsymbol{\chi}_2)$-plane as well as the plane spanned by $\boldsymbol{\mu}_i$ and $\boldsymbol{\chi}_0$. $\bar{\kappa}$ is then the angle between n_i and $\boldsymbol{\chi}_1$, in the interval $[0..2\pi)$ (see figure 5).

$$\boldsymbol{n}_i^* = \boldsymbol{\chi}_0 \times (\boldsymbol{\mu}_i \times \boldsymbol{\chi}_0) \tag{9}$$

$$\boldsymbol{n}_i = \frac{\boldsymbol{n}_i^*}{|\boldsymbol{n}_i^*|} \tag{10}$$

q_i^0 is a quantity calculated to distinguish between positive and negative angles.

$$q_i^0 = \operatorname{sign}(\boldsymbol{n}_i \cdot \boldsymbol{\chi}_2) \tag{11}$$

$$\bar{\kappa}_i = q_i^0 \arccos(\boldsymbol{\chi}_1 \cdot \boldsymbol{n}_i) + (1 - q_i^0)\pi \tag{12}$$

Once the principle angles are determined, depth information for S_i is looked up from a table $K(u, \psi_i, \lambda_i)$, with compatible resolution. The table contains angular data of the coverage of an interface with parameters ψ_i and λ_i for all Depth-Buffer lines. The tables are precomputed with a fixed $\bar{\kappa} = 0$, as such the information has to be rotated before it can be applied to the buffer. $\bar{\kappa} = 0$ refers to an interface that has a normal vector orthogonal to $\boldsymbol{\chi}_2$. The

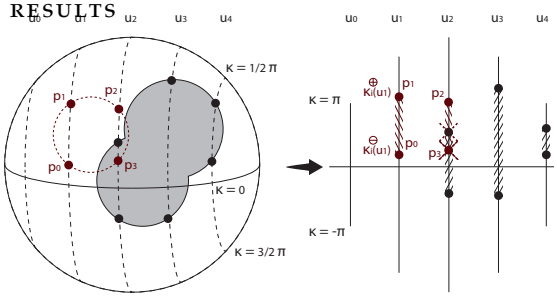


Figure 3: A neighbor sphere (in dashed lines) is added into a Depth-Buffer with preexisting depth-information. The sphere crosses Depth-Buffer lines u_1 and u_2. Depth-Information is looked up from a precomputed table, in which only u_1 and u_2 contain values. The new information is then added to the buffer by creating a new layer for u_1 and extending the preexisting layer by inserting $\boldsymbol{p}_2$ and removal of the preexisting bound.

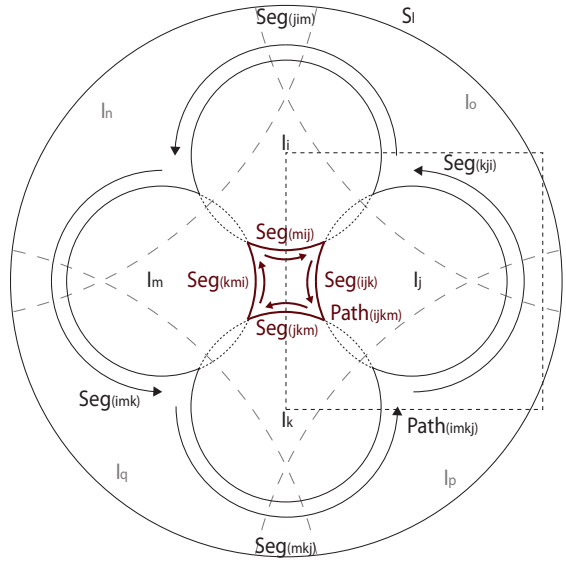


Figure 4: An exposed boundary is shown. The boundary is formed by interfaces I_i, I_j, I_k and I_m and contains the segments Seg_{ijk}, Seg_{jkm}, Seg_{kmi} and Seg_{mij}. During the algorithm, interfaces I_n, I_o, I_p and I_q are detected as internal and removed. This causes the internal boundary with segments Seg_{jim}, Seg_{imk}, Seg_{mkj} and Seg_{kji} to emerge.

rotation consists of adding $\bar{\kappa}_i$ to the depth information from the table. To compensate for some errors due to the finite resolution, the opening angles λ_i are slightly decreased while building the buffer. The amount of reduction is called slack and will be denoted by ξ.

For each Depth-Buffer line with distance u, we calculate two points: $\kappa_i^{\oplus}(u)$ and $\kappa_i^{\ominus}(u)$

$$\kappa_i^{\oplus}(u) = \bar{\kappa}_i + K(u, \psi_i, \lambda_i - \xi) \tag{13}$$

$$\kappa_i^{\ominus}(u) = \bar{\kappa}_i - K(u, \psi_i, \lambda_i - \xi) \tag{14}$$

These points are limiting points on a radial line, between which S_l is covered by S_i. Existing information in the buffer needs to be updated with this new information. An update will either extend an existing layer, add a new layer, or keep the buffer unchanged. The procedure is visualized in figure 3: A new circle (dashed) is added to the buffer, which already contains information. The insertion of p_0 and p_1 causes a new layer in the line at u_1, and the insertion of p_2 and p_3 the deletion of a limiting point at line u_2. After all neighbor spheres have been added to the buffer, another pass is performed to determine whether an interface is internal, and should be removed, or boundary and should be kept. For this pass, the spheres are added again into the buffer with disregard of the slack. If the (simulated) addition would cause the buffer to change, the interface is considered boundary and is kept.

This procedure removes not just buried interfaces, but also interfaces which are internal. As such, any subsequent algorithm used to identify the Gauss-Bonnet paths from spherical intersections, will be confronted with segments that belong to internal interfaces. The problem is shown in figure 4. Here, the true exposed area which is spanned by $Path_{ijkm}$ is shown. In this scenario, internal interfaces I_n, I_o, I_p, and I_q have been removed by the algorithm, causing $Path_{imkj}$ to emerge. This path is internal and does not bound an exposed area. It is possible to determine which segments are truly boundary, and which not, by analysis of the paths which connect them. Each segment belongs to a single unique path, which either takes completely part in the boundary of an exposed surface, or is completely internal. To do such distinction, the probability that a path is exposed or internal is calculated. Points are created on what the Depth-Buffer knows are exposed segments. A distance operator is established, which assigns each segment of each path a distance to the exposed surface. Intuitively, paths with small cumulative distances are more likely to be exposed than such with large cumulative distances.

First, we will present equations to convert arbitrary points from Depth-Buffer-space into Cartesian space and subsequently into $(\Phi, \psi_j, \lambda_j)$ space. The reason for doing so is that in $(\Phi, \psi_j, \lambda_j)$-space we are able to compare intersection-points that belong

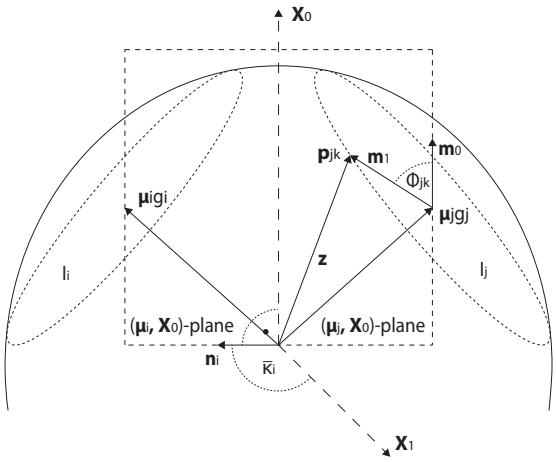


Figure 5: Determination of angles $\bar{\kappa}_i$ and Φ_{jk} is depicted. To calculate $\bar{\kappa}_i$, a vector n_i is laid into the (μ_i, χ_0)-plane, so that it is orthogonal to χ_0. $\bar{\kappa}_i$ is then the angle to χ_1. For a vector z which is pointing to intersection-point p_{jk}, Φ_{jk} is calculated by simultaneously laying a vector m_0 into the (μ_j, χ_0)-plane and a plane orthogonal to μ_j. Φ_{jk} is then the angle between this vector and vector m_1, which points from the center of I_j to $p_j k$.

to different segments with each other, enabling the quantification of angular distances to the exposed surface. For an arbitrary point on the boundary of interface I_j, Φ is the angle between a vector from the center of I_j to that point and the (χ_0, χ_1)-plane. ψ_j and λ_j are determined by equations 8 and 7

The projection into Cartesian space depends on the Depth-Buffer. For B^0 the conversion is as follows: For any point in Depth-Buffer space with coordinates (u,κ) the radius h of the radial line with distance u is calculated.

$$h = \sqrt{1 - u^2} \tag{15}$$

Then, the conversion is a simple rotation of χ_1 around χ_0 using κ, and translation along χ_0 using h.

$$z = \begin{pmatrix} u \\ h\cos(\kappa) \\ h\sin(\kappa) \end{pmatrix} \tag{16}$$

And for B^1 likewise:

$$z = \begin{pmatrix} h\sin(\kappa) \\ u \\ h\cos(\kappa) \end{pmatrix} \tag{17}$$

Now that we have Cartesian coordinates, we are interested in projecting them into the interface plane of I_j. Projection of an arbitrary Cartesian vector z into $(\Phi, \psi_j, \lambda_j)$-space of interface I_j is performed with the following sequence of equations: First, we create a vector m_0, which is simultaneously in the plane of interface I_j and the plane spanned by μ_j and χ_0 (see figure 5).

$$m_0 = \mu_j \times (\chi_0 - \mu_j) \times \mu_j \tag{18}$$

The vector m_1 points to vector z in the plane of interface I_j

$$m_1 = z - g_j \mu_j \tag{19}$$

q^1 is a quantity calculated to distinguish between positive and negative angles.

$$q^1 = \text{sign}\,(m_1 \cdot m_0) \tag{20}$$

$$\Phi = q^1 \arccos\left(\frac{m_1 \cdot m_0}{|m_1||m_0|}\right) \tag{21}$$

Equations 15 to 21 sequentially convert arbitrary points from Depth-Buffer space into $(\Phi, \psi_j, \lambda_j)$-space. Now, rays are generated which originate from the center of S_l and intersect with some point on each segment (see figure 6). It was already described that an interface passes the Depth-Buffer test if it changes the buffer in the second testing pass. As such, the change in the buffer for Depth-Buffer line u_a can be quantified by the vector (u_a, κ_a), which is pointing to a spot on a segment that would cause the deletion of previous bounds. The precision of these spots depend on the resolution of the buffer and the resolution of the look-up table. To minimize computational demand, all spots for a segment are collected and subsequently pruned, so that only their centroids C remain.

Let's denote a segment by Seg_{ijk}, where ijk are indexes to interfaces I_i, I_j and I_k, such that there is an arc between an intersection-point formed by I_i, I_j and an intersection-point formed by I_j and I_k (see figure 4). The centroid C_{ijk} for interface I_j and Seg_{ijk} as well as intersection-points p_{ij}, p_{jk}, etc.. are now projected into $(\Phi_{ijk}, \psi_j, \lambda_j)$-space as Φ_{ijk}, Φ_{ij}, Φ_{jk}, etc... Note the different use of the indexes. We denote the Φ value of centroid C_{ijk} with index ijk and intersection-points p_{ij}, p_{jk} with indexes ij and jk respectively. A distance operator will now be defined as follows:

$$\mathcal{D}\begin{pmatrix} \Phi_c \\ \Phi_{up} \\ \Phi_{lo} \end{pmatrix} = \begin{cases} 0 & \Phi_{up} \geqslant \Phi_c \geqslant \Phi_{lo} \\ R & \text{else} \end{cases} \tag{22}$$

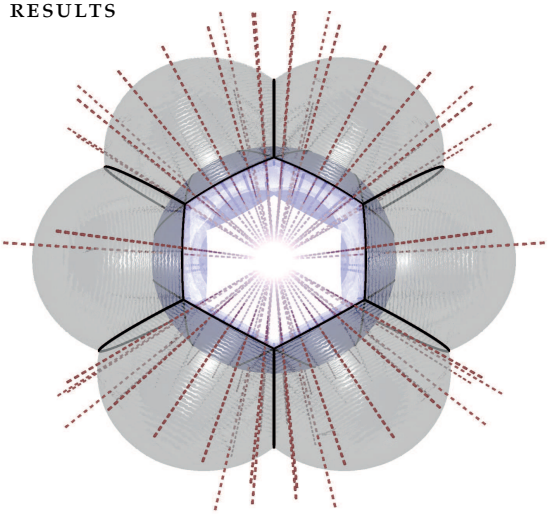


Figure 6: Rays are shot from the center of the sphere through its exposed boundary. All rays are shown, however they are subsequently pruned, so that just one ray remains per segment. Their accuracy largely depend on the resolution of the buffer.

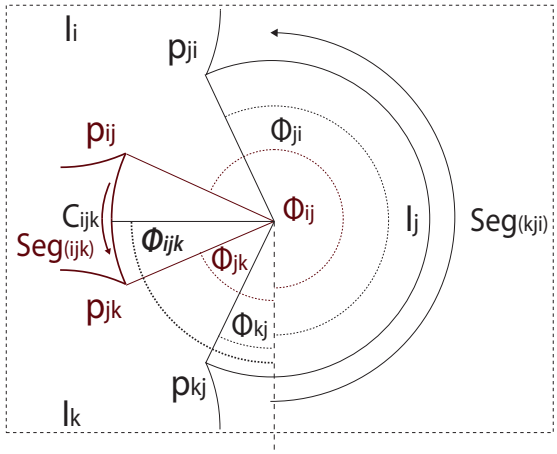


Figure 7: Subset of figure 4. Angles for intersection-points belonging to segments Seg_{ijk} and Seg_{ijk} are calculated to a fixed axis. Centroid C_{ijk} with angle Φ_{ijk} is located in between Intersection-points p_{ij} and p_{jk} with angles Φ_{ij} and Φ_{jk}; therefore, the distance from Seg_{ijk} to the centroid is devised to be zero. Seg_{kji} does not contain the centroid and will have a distance greater than zero.

With:

$$R = \min\left(|\Phi_c - \Phi_{up}|, |\Phi_c - \Phi_{lo}|\right) \tag{23}$$

In which Φ_{up} and Φ_{lo} are upper and lower segment limits respectively, and Φ_c a centroid. The various Φ values are shown as an example in figure 7, which is a cutout of the dashed box in figure 4. Two Segments are shown: Exposed Seg_{ijk} belongs to $Path_{ijkm}$ and connects intersection-points p_{ij} and p_{jk}, while internal Seg_{kji} belongs to $Path_{imkj}$ and connects p_{kj} and p_{ji}, yielding angles Φ_{ij}, Φ_{jk}, Φ_{kj} and Φ_{ji} to a fixed reference vector respectively. The reference vector is identical to $\boldsymbol{m}_0$ from figure 5.

For Seg_{ijk}, the angles Φ_{ij} and Φ_{jk} are used as an upper and lower segment limit respectively, yielding distance $\mathcal{D}(\Phi_{ij}, \Phi_{jk}, \Phi_{ijk})$. Since Seg_{ijk} contains the centroid, its distance is zero. Likewise, for Seg_{kji}, upper and lower segment limits Φ_{kj} and Φ_{ji} are used in $\mathcal{D}(\Phi_{ji}, \Phi_{kj}, \Phi_{ijk})$. Seg_{kji} does not contain the centroid; therefore the evaluation will result in a non-zero distance, decreasing likelihood that $Path_{ijkm}$ is exposed.

For a path P_t, which contains segments Seg_{ijk} ... Seg_{mno}, a set of distances is defined:

$$\boldsymbol{d}_t = \left\{\mathcal{D}(\Phi_{ijk}, \Phi_{ij}, \Phi_{jk}) \ldots \mathcal{D}(\Phi_{mno}, \Phi_{mn}, \Phi_{no})\right\} \tag{24}$$

The probability that any path P_t belongs to the exposed boundary or is internal, given the distances of

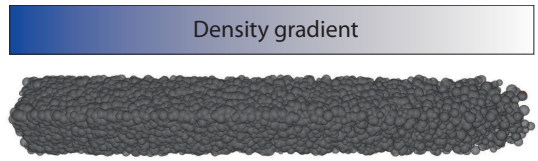


Figure 8: The artificial molecule that was used to calculate unbiased probability distributions for $p(\boldsymbol{e}|\boldsymbol{d})$. Density of the spheres are lowest to the right of the picture and increase towards the left. The radii of the spheres have been drawn randomly from a biologically meaningful range.

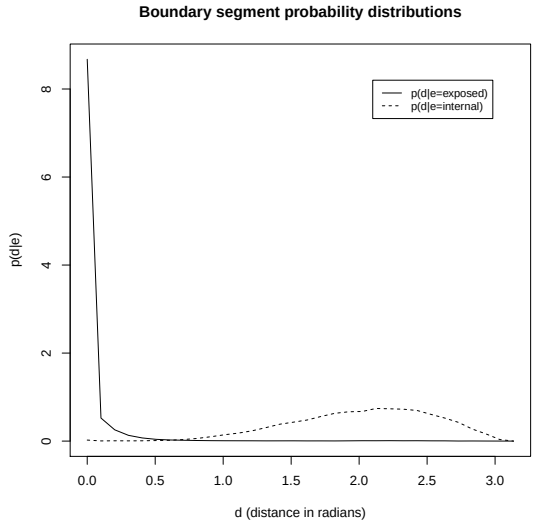


Figure 9: Probability distributions for $p(\boldsymbol{d}|\boldsymbol{e})$. Exposed segments usually contain centroids and therefore are assigned a distance of zero. Locations of centroids are however fuzzy and depend on the resolution of the buffer. Thus, small deviations from zero are occuring at times, giving shape to the exposed distribution. Internal segments almost never enclose centroids. The mode of the internal distribution is far from zero.

its segments, will be denoted by $p(e = \text{exposed}|\boldsymbol{d}_t)$ and $p(e = \text{internal}|\boldsymbol{d}_t)$ respectively. The distances per segment and path are truly independent. As such, we can formulate the above probability as:

$$p(\boldsymbol{e}|\boldsymbol{d}_t) = \frac{p(\boldsymbol{e}) \prod_{d \in \boldsymbol{d}_t} p(d|\boldsymbol{e})}{\sum_{e \in \boldsymbol{e}} \left(p(e) \prod_{d \in \boldsymbol{d}_t} p(d|e) \right)} \tag{25}$$

In which $p(\boldsymbol{e})$, with $\boldsymbol{e} = \{\text{exposed}, \text{internal}\}$, is a prior probability that path P_t is exposed and $p(d|\boldsymbol{e})$ a likelihood that describes the probability of encountering a distance d if it is known that the path is either exposed or internal. In the above equation we have use the fact that $\sum_{e \in \boldsymbol{e}} p(e|\boldsymbol{d}) = 1$. In order to quantify $p(\boldsymbol{e})$ and $p(d|\boldsymbol{e})$ without biasing for any kind of molecule type, we have created a purely artificial molecule which is depicted in figure 8.
In this molecule, spheres with radii that were randomly chosen from a biological meaningful range were added with increasing density towards one of its ends. Subsequently, exposed boundaries were detected without utilization of the Depth-Buffer, and later on with activated Depth-Buffer. Both results were compared to identify true exposed segments. Afterward, distances were collected and assigned their respective classes, which gave rise to the two probability densities shown in figure 9. Priors were determined by accumulating frequencies for paths that were exposed and internal. Once probability $p(\boldsymbol{e}|\boldsymbol{d})$ is calculated, the path is determined exposed if the probability $p(e = \text{exposed}|\boldsymbol{d})$ is equal or greater than 0.5.

0.3 RESULTS

Tests were performed on a set of 56 proteins. Plots 10 and 11 show how error and speed-up behaves for different values of the slack parameter. Speed-ups were calculated just for the identification of the exposed boundary, which excluded the calculation of the area. The timing starts with a list of spheres, which has already been pruned by a neighbor list, so to not to include speed-ups generated by the neighbor-list. As expected, the higher the slack, the more interfaces pass the Depth-Buffer test and slow down calculations. The error behaves differently, in the sense that it remains relatively constant for the duration of the decrease in slack, until too many exposed interfaces are removed, spiking the error. The oscillation in the constant phase is mainly due to small cancellation of errors, when the subsequent algorithm deals with different sets of interfaces which passed the Depth-Buffer test.
The amount of speed-up which can be achieved mildly depends on the size of the molecule ($r \sim 0.6$), with larger molecules generally facilitating higher speed-ups, but strongly correlates with the percentage of removed interfaces ($r \sim 0.8$). Larger molecules have larger buried volumes than small molecules and therefore more completely buried atoms, which are removed right after the Depth-Buffer test. As such, no time is spent on the identification the exposed area - which is nonexistent in these cases. For very small molecules the amount of removed interfaces decreases tremendously, resulting in almost neglectible speed-ups.

0.4 DISCUSSION

Algorithms, like TRIFORCE[18], Richmond's method[11], Connolly's method[12], GETAREA[16], MSEED[13], SASAD[15], Gogonea's method[14], ACCAR[17], POWERSASA[28], and many more, principally only rely on the exposed boundary for their calculations. Most of these methods use the Gauss-Bonnet theorem, with which it is possible to

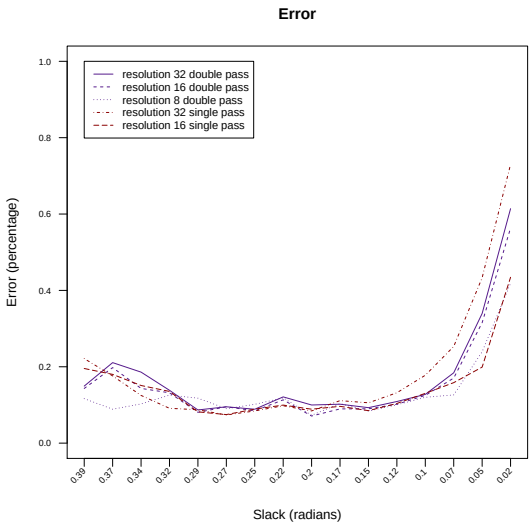


Figure 10: Development of the error for a decreasing amount of slack. For the most part, the error is relatively stable, but abruptly rises once the slack is too low and the algorithm removes too many exposed interfaces. Single pass and double pass refer to usage of one or two (orthogonal) Depth-Buffers respectively

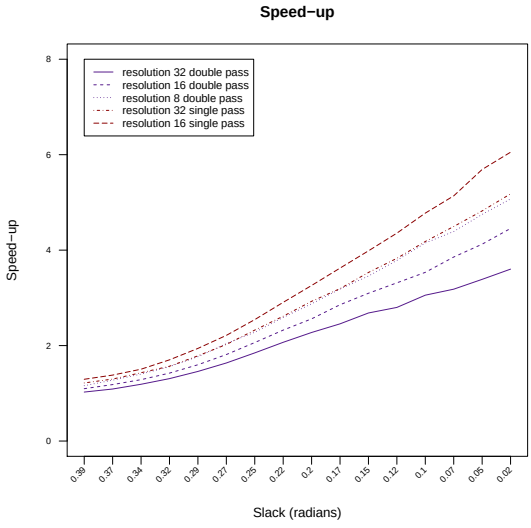


Figure 11: The speed-up is almost linear with the amount of slack and is much higher when only one buffer is used. Single pass and double pass refer to usage of one or two (orthogonal) Depth-Buffers respectively

Table 1: Protein set

PDB code	atoms	error (percentage)	speed-up	removed interfaces (percentage)
1PLX	66	0.038	1.250	66.40
1CBH	451	0.095	4.696	89.31
1SP2	470	0.057	4.227	84.98
5RXN	703	0.216	5.966	90.47
1I6F	754	0.205	3.809	88.86
4PTI	796	0.401	5.676	89.88
1FAS	817	0.266	4.234	89.67
1SN3	845	0.295	5.806	89.85
1CSP	888	0.190	6.079	90.87
2CRO	963	0.408	6.286	91.38
1FVQ	970	0.010	6.093	92.28
1SDF	1007	0.019	4.509	89.10
1HIP	1081	0.119	5.633	90.87
1UBQ	1089	0.843	6.523	91.87
1PHT	1157	0.227	5.593	91.15
1J5D	1294	0.056	5.031	90.89
2SSI	1385	0.247	4.500	90.03
2CDV	1407	0.019	4.328	86.48
1CQY	1425	0.045	6.603	92.22
1OPC	1453	0.151	5.653	91.65
1KTE	1488	0.104	5.694	92.45
1NSO	1543	0.225	4.576	90.27
1JLI	1626	0.212	5.107	91.87
1RN3	1677	0.250	5.392	91.62
2PAZ	1681	0.347	5.560	92.58
1OOI	1756	0.065	5.337	91.74
1CHN	1771	0.070	6.726	92.15
6LYZ	1777	0.109	5.889	91.87
1AGI	1779	0.519	5.068	91.03
1PDO	1794	0.551	5.788	92.15
1K40	1809	0.078	5.237	92.25
1BFG	1828	0.287	6.308	92.72
1LIT	1829	0.043	5.964	92.34
1BSN	1888	0.912	4.208	89.73

continued in table 2

calculate solvent exposed areas solely with information from vectors that are tangential to boundary arcs. In our analysis of the performance of the TRIFORCE algorithm, it became evident, that most computational time is spent on the construction of the exposed boundary, and a much smaller part on the actual calculation of the exposed area, i.e. integration over tessellated surface patches or evaluation of the Gauss-Bonnet theorem. Hence, the fast detection of the exposed boundary should speed-up area calculations significantly, and might also be used in algorithms, which do not directly rely on exposed boundaries like, e.g. LCPO[20].

Speed-ups are difficult to compare. As already stated by Weiser et al.[26] it depends much on the subsequent algorithms that identify the exposed boundary and the algorithms that calculate actual

continued from table 1

PDB code	atoms	error (percentage)	speed-up	removed interfaces (percentage)
1BJ7	2133	0.399	6.072	92.82
1KXA	2171	0.016	5.300	91.75
2I1B	2184	0.488	5.888	91.85
1MBS	2200	0.111	5.915	91.47
1EMR	2262	0.380	6.789	92.36
1CZT	2336	0.025	5.990	92.34
1IL6	2406	0.022	5.768	91.69
1LKI	2439	0.075	5.773	92.14
2PTN	2901	0.365	6.344	92.54
5PAD	2937	0.264	5.820	92.67
1SUR	3144	0.075	6.679	92.02
2HVM	3702	0.107	6.602	93.39
2CYP	3987	0.058	6.225	92.79
1RHD	4147	0.142	5.936	92.18
3APP	4161	0.133	6.663	93.37
2TMN	4281	0.068	6.552	93.40
2TS1	4378	0.146	6.468	92.68
1BNI	4477	0.081	6.310	92.03
1FRG	6007	0.139	6.750	93.00
1MCP	6031	0.022	6.866	92.81
1GND	6075	0.027	6.762	92.48
1CA0	7758	0.327	7.224	93.02

Table 2: Procentual errors, speed-ups and percentage of removed interfaces are shown for a set of 56 proteins.

areas. Nonetheless, we give speed-ups for our algorithm TRIFORCE, to show the behavior of the Depth-Buffer with different parameters. What figures 10, 11 and table 2 communicate is how increased performance can be bought by acceptance of some error.

The Multi-Layered Depth-Buffer algorithm determines a different set of spheres as in Fraczkiewicz[16] and Weiser[25,26]. These algorithms calculate the set of spheres excluding buried spheres, which do not contribute to the exposed area and can be safely removed. Weiser et al. report a removal of 73% of spheres using one of their methods[25] and 35% using the other[26]. They report a maximal error of about 0.5% for large molecules. This is comparable to our results, though in our method it is possible to adjust the amount of error by choosing different Depth-Buffer configurations. As is observable in table 2, the largest removal of spheres was 93% for protein 1CA0, but values up to 95% have been observed by the authors using different configurations.

In the algorithm used in GETAREA, a very elegant technique is employed to find the exposed boundary. Comparison with this method is problematic however, because capabilities are vastly different. That method divides the molecular space into half-spaces, wherever an atom-atom interface is located. With a subsequent geometric inversion and the creation of the convex hull of the inverted interface pointers, the exposed boundary is calculated. The problematic part of the algorithm is the geometric inversion. There, it is assumed that the distance to an interface will never be 0, or indeed, negative - which occurs for what we call concave interfaces. For a description see reference[18]. Concave interfaces frequently occur in e.g. molecular dynamics simulations for arbitrary atom-atom intersections, and almost always occur for hydrogens, since they mostly reside within the Van der Waals radius of the atom they are connected to.

The algorithm has been implemented in c++ and is packed together with our TRIFORCE library. It can be downloaded from http://lavandula.imim.es/triforce and used independently from TRIFORCE.

0.5 CONCLUSION

We have presented a novel - and to the authors' knowledge unique - algorithm to detect atoms which contribute to the solvent exposed area of a molecule. The algorithm processes the capability to speed-up calculations of a number of algorithms that rely on the exposed boundary to perform area calculations, and might accelerate other algorithms as well. Although it is not error free, we have shown in a study that a balance between accuracy and speed-up can be found, and that the overall error is small in comparison to the prefigured speed-ups.

0.6 ACKNOWLEDGMENTS

We thank Montserrat Corbera for fruitful discussions and proof-reading of the manuscript.

BIBLIOGRAPHY

[1] C. J. Fennell, C. W. Kehoe, and K. A. Dill. Modeling aqueous solvation with semi-explicit assembly. *Proc. Natl. Acad. Sci. USA*, 108(8):3234–3239, 2011.

[2] Michael Feig and Charles L Brooks III. Recent advances in the development and application of implicit solvent models in biomolecule simulations. *Current opinion in structural biology*, 14(2):217–224, 2004.

[3] Vickie Tsui and David A Case. Theory and applications of the generalized born solvation model in macromolecular simulations. *Biopolymers*, 56(4):275–291, 2000.

[4] Junmei Wang, Wei Wang, Shuanghong Huo, Matthew Lee, and Peter A Kollman. Solvation model based on weighted solvent accessible surface area. *The Journal of Physical Chemistry B*, 105(21):5055–5067, 2001.

[5] D. Eisenberg and A. D. McLachlan. Solvation energy in protein folding and binding. *Nature*, 319:199–203, 1986.

[6] E. Silla, F. Villar, O. Nilsson, J.L. Pascual-Ahuir, and O. Tapia. Molecular volumes and surfaces of biomacromolecules via gepol: A fast and efficient algorithm. *Journal of Molecular Graphics*, 8(3):168 – 172, 1990. ISSN 0263-7855. doi: 10.1016/0263-7855(90)80059-O. URL http://www.sciencedirect.com/science/article/pii/026378559080059O.

[7] André H. Juffer and Hans J. Vogel. A flexible triangulation method to describe the solvent-accessible surface of biopolymers. *Journal of Computer-Aided Molecular Design*, 12(3):289–299, 1998. ISSN 0920-654X. doi: 10.1023/A:1016089901704. URL http://dx.doi.org/10.1023/A%3A1016089901704.

[8] Ján Buša, Shura Hayryan, Chin-Kun Hu, Jaroslav Skřivánek, and Ming-Chya Wu. Enveloping triangulation method for detecting internal cavities in proteins and algorithm for computing their surface areas and volumes. *Journal of computational chemistry*, 30(3):346–357, 2009.

[9] Junping Xiang and Maolin Hu. An efficient method for sampling and computing molecular surface. In *Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics - Volume 01*, BMEI '08, pages 52–56, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3118-2. doi: 10.1109/BMEI.2008.153. URL http://dx.doi.org/10.1109/BMEI.2008.153.

[10] Byungjoo Kim, K Kim, and J Seong. Gpu accelerated molecular surface computing. *Appl Math Inf Sci*, 6(1S): 185S–194S, 2012.

[11] Timothy J. Richmond. Solvent accessible surface area and excluded volume in proteins: Analytical equations for overlapping spheres and implications for the hydrophobic effect. *Journal of Molecular Biology*, 178(1):63 – 89, 1984. ISSN 0022-2836. doi: 10.1016/0022-2836(84)90231-6. URL http://www.sciencedirect.com/science/article/pii/0022283684902316.

[12] M. L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16(5):548–558, Oct 1983. doi: 10.1107/S0021889883010985. URL http://dx.doi.org/10.1107/S0021889883010985.

[13] G. Perrot, B. Cheng, K. D. Gibson, J. Vila, K. A. Palmer, A. Nayeem, B. Maigret, and H. A. Scheraga. Mseed: A program for the rapid analytical determination of accessible surface areas and their derivatives. *J. Comput. Chem.*, 13(1):1–11, 1992.

[14] Valentin Gogonea and Eiji Ōsawa. An improved algorithm for the analytical computation of solvent-excluded volume. the treatment of singularities in solvent-accessible surface area and volume functions. *Journal of Computational Chemistry*, 16(7):817–842, 1995. ISSN 1096-987X. doi: 10.1002/jcc.540160703. URL http://dx.doi.org/10.1002/jcc.540160703.

[15] S. Sridharan, Anthony Nicholls, and Kim A. Sharp. A rapid method for calculating derivatives of solvent accessible surface areas of molecules. *J. Comput. Chem.*, 16(8): 1038–1044, 1995.

[16] Robert Fraczkiewicz and Werner Braun. Exact and efficient analytical calculation of the accessible surface areas and their gradients for macromolecules. *J. Comput. Chem.*, 19(3):319–333, 1998.

[17] Shura Hayryan, Chin-Kun Hu, Jaroslav Skřivánek, Edik Hayryan, and Imrich Pokorný. A new analytical method for computing solvent-accessible surface area of macromolecules and its gradients. *J. Comput. Chem.*, 26(4):334–343, 2005.

[18] Nils J. D. Drechsel, Christopher J. Fennell, Ken A. Dill, and Jordi Villà-Freixa. Triforce: Tessellated semi-analytical solvent exposed surface areas and their derivatives. *in preparation*, 2013.

[19] Jie Liang, Herbert Edelsbrunner, Ping Fu, Pamidighantam V. Sudhakar, and Shankar Subramaniam. Analytical shape computation of macromolecules: I. molecular area and volume through alpha shape. *Proteins: Structure, Function, and Bioinformatics*, 33(1):1–17, 1998. ISSN 1097-0134. doi: 10.1002/(SICI)1097-0134(19981001)33:1<1::AID-PROT1>3.0.CO;2-O. URL http://dx.doi.org/10.1002/(SICI)1097-0134(19981001)33:1<1::AID-PROT1>3.0.CO;2-O.

[20] Jörg Weiser, Peter S. Shenkin, and W. Clark Still. Approximate atomic surfaces from linear combinations of pairwise overlaps (LCPO). *J. Comput. Chem.*, 20(2):217–230, 1999.

[21] S. J. Wodak and J. Janin. Analytical approximation to the accessible surface area of proteins. *Proc. Natl. Acad. Sci. USA*, 77(4):1736–1740, 1980.

[22] Franca Fraternali and Luigi Cavallo. Parameter optimized surfaces (pops): analysis of key interactions and conformational changes in the ribosome. *Nucleic Acids Research*, 30(13):2950–2960, 2002. doi: 10.1093/nar/gkf373. URL http://nar.oxfordjournals.org/content/30/13/2950.abstract.

[23] Georgy Rychkov and Michael Petukhov. Joint neighbors approximation of macromolecular solvent accessible surface area. *J. Comput. Chem.*, 28(12):1974–1989, 2007.

[24] K.D. Gibson and H.A. Scheraga. Exact calculation of the volume and surface area of fused hard-sphere molecules with unequal atomic radii. *Molecular Physics*, 62(5):1247–1265, 1987. doi: 10.1080/00268978700102951. URL http://www.tandfonline.com/doi/abs/10.1080/00268978700102951.

[25] Jörg Weiser, Armin A. Weiser, Peter S. Shenkin, and W. Clark Still. Neighbor-list reduction: Optimization for computation of molecular van der waals and solvent-accessible surface areas. *Journal of Computational Chemistry*, 19(7):797–808, 1998. ISSN 1096-987X. doi: 10.1002/(SICI)1096-987X(199805)19:7<797::AID-JCC9>3.0.CO;2-L. URL http://dx.doi.org/10.1002/(SICI)1096-987X(199805)19:7<797::AID-JCC9>3.0.CO;2-L.

[26] Jörg Weiser, Peter S. Shenkin, and W. Clark Still. Fast, approximate algorithm for detection of solvent-inaccessible atoms. *Journal of Computational Chemistry*, 20(6):586–596, 1999. ISSN 1096-987X. doi: 10.1002/(SICI)1096-987X(19990430)20:6<586::AID-JCC4>3.0.CO;2-J. URL http://dx.doi.org/10.1002/(SICI)1096-987X(19990430)20:6<586::AID-JCC4>3.0.CO;2-J.

[27] Pieter F. W. Stouten, Cornelius Frömmel, Haruki Nakamura, and Chris Sander. An effective solvation term based on atomic occupancies for use in protein simulations. *Molecular Simulation*, 10(2-6):97–120, 1993. doi: 10.1080/08927029308022161. URL http://www.tandfonline.com/doi/abs/10.1080/08927029308022161.

[28] Konstantin V. Klenin, Frank Tristram, Timo Strunk, and Wolfgang Wenzel. Derivatives of molecular surface area and volume: Simple and exact analytical formulas. *J. Comput. Chem.*, 32(12):2647–2653, 2011.

4.6 OUTLOOK

4.6.1 *Towards Semi-Analytical Semi-Explicit Assembly*

We have seen that TRIFORCE can be used to efficiently calculate surface areas. But it can do more. Areas are a special case of surface integrals in which the functional value of the surface is 1. TRIFORCE can integrate over arbitrary functional values, as long as they are smooth and their complexity is limited. This can be used to create a semi-analytical version of the previous fully numerical Semi-Explicit Assembly.

The first step in Semi-Explicit Assembly is to take a target atom and probe its Lennard Jones field, to which the atom itself and all its neighbors contribute[126]. All atoms are modeled as spheres with an extended radius corresponding to their van der Waals radius plus average distance to the water. For the purpose of probing, rays are shot from the center of the sphere through its solvent accessible surface area. Now, the minimum potential along each ray is located, all minima averaged and combined into effective Lennard Jones parameter ϵ_{eff} and σ_{eff}.

Finding the minimum along each ray is the culprit of the method. It is purely numerical, not even for a field containing just two Lennard Jones sources there is an analytical closed form. One solution to the problem involes creating a precomputed function of minima and their well-depths, which would interpolate between adjacent minima.

This function however would be dependent on the constellation of all neighbors, thus the combinatorial requirements would be astronomical. We can however utilize a divide and conquer approach in which a table is built that consists of just two Lennard Jones sources. Subsequently the information in the table would be used to progressively combine minima of distinct Lennard Jones sources to a global minimum.

In a discrete approximation, where we not yet consider surfaces but keep working with rays, similar to the Semi-Explicit Assembly, we take all the rays, the sphere for which we calculate the Lennard Jones field S_l, and exactly one neighbor S_k. Now, using aforementioned precomputed table f, we look up the minimum $\sigma_j^{(i)}$ and well-depth $\epsilon_j^{(i)}$ for iteration i and ray j.

$$(\sigma_j^{(i)}, \epsilon_j^{(i)}) = f(\sigma_l, \sigma_k, \epsilon_l, \epsilon_k, d_{lk}, \text{ray}_j) \tag{64}$$

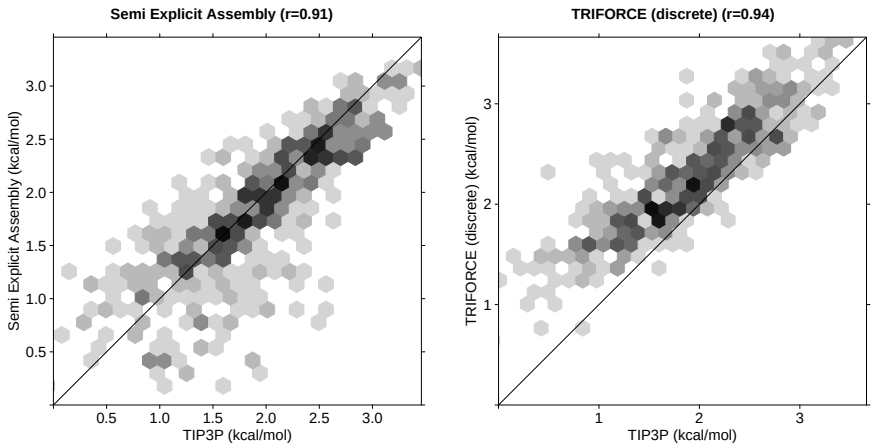


Figure 28: Semi-Explicit Assembly vs. TRIFORCE (discrete)

In which σ_l, σ_k, ϵ_l, ϵ_k are the "default" Lennard Jones parameters for standard non-bonded interactions, d_{lk} the distance between spheres S_l and S_k, and ray_j a vector pointing into the direction of the jth ray.

Then, just like in Semi-Explicit Assembly, we extract the effective parameter for iteration j by averaging to $\sigma_{eff}^{(i)}$ and well-depth $\epsilon_{eff}^{(i)}$. Now, we take another neighbor S_m and repeat

$$\left(\sigma_j^{(i+1)}, \epsilon_j^{(i+1)}\right) = f\left(\sigma_{eff}^{(i)}, \sigma_m, \epsilon_{eff}^{(i)}, \epsilon_m, d_{lm}, ray_j\right) \tag{65}$$

until it converges to the approximate true minimum. The extracted effective parameter outperform Semi-Explicit Assembly in terms of correlation to explicit water TIP3P (see correlation plots in figure 28).

The calculation of the ray-wise average is done in the following way:

$$\epsilon_{eff}^{(i)} = \frac{1}{N} \sum_{j \in rays} \epsilon_j^{(i+1)} \tag{66}$$

$$\sigma_{eff}^{(i)} = \frac{1}{N} \sum_{j \in rays} \sigma_j^{(i+1)} \tag{67}$$

In which N is the total number of rays. This can be written in continuous form:

$$\left(\epsilon_{eff}^{(i)}, \sigma_{eff}^{(i)}\right) = \frac{1}{A} \iint_D f\left(\sigma_{eff}^{(i)}, \sigma_m, \epsilon_{eff}^{(i)}, \epsilon_m, d_{lm}, ray(\phi, \theta)\right) dS \tag{68}$$

In which $\mathrm{ray}(\phi, \theta)$ is a function that yields the vector of a ray given its spherical coordinates ϕ and θ, and A is the exposed surface area. Equation 68 is essentially what TRIFORCE is able to compute. However, multiple adjustments have to be done. The integration tables $T(\Phi, \psi, \lambda)$ need to be extended by 3 additional dimensions to $T_t^{(s)}(\Phi, \psi, \lambda, d_{lk}, \epsilon_{\mathrm{eff}}, \sigma_{\mathrm{eff}})$, in which t specifies either σ or ϵ, i.e. we have separate tables for these quantities, s represents the species, which can take the form of e.g (carbon, oxygen, hydrogen, nitrogen) or more species if necessary, d_{lk} is the distance between the sphere of interest and a neighbor, and ϵ_{eff} and σ_{eff} are the current effective Lennard Jones parameters. The "default" Lennard Jones parameters for each species are fixed and do not change during the course of a simulation, so we create separate tables for each species between we will not interpolate.

The type of table parametrization requires that $\boldsymbol{v}_{lk}$, the vector connecting S_l and S_k coincides with the tessellation axis $\boldsymbol{\chi}$. Because we do not want to create tessellations for each neighbor (which would be prohibitively costly), we have to convert a current tessellation $\nabla(\boldsymbol{\chi}_i)$ with tessellation axis χ_i into a rotated tessellation $\nabla(\boldsymbol{\chi}_j)$ of tessellation axis $\boldsymbol{\chi}_j$. Fortunately this is very easy, we simply need to add the angle between the old and new tessellation planes to Φ and recalculate ψ based on the new tessellation axis:

$$\boldsymbol{n}_u^{*(i)} = \boldsymbol{\mu}_u \times \boldsymbol{\chi}_i \tag{69}$$

$$\boldsymbol{n}_u^{(i)} = \frac{\boldsymbol{n}_u^{*(i)}}{|\boldsymbol{n}_u^{*(i)}|} \tag{70}$$

$$\boldsymbol{n}_u^{*(j)} = \boldsymbol{\mu}_u \times \boldsymbol{\chi}_j \tag{71}$$

$$\boldsymbol{n}_u^{(j)} = \frac{\boldsymbol{n}_u^{*(j)}}{|\boldsymbol{n}_u^{*(j)}|} \tag{72}$$

$$\Phi_{uw}^{(j)} = \Phi_{uw}^{i} + \arccos\left(\boldsymbol{n}_u^i \cdot \boldsymbol{n}_u^j\right) \tag{73}$$

For some spherical interface u and segment w.

$$\psi_u^{(j)} = \arccos(\boldsymbol{\chi_j} \cdot \boldsymbol{\mu}_u) \tag{74}$$

4.6.2 *Towards GPU-TRIFORCE*

TRIFORCE has been designed with the intention of running on GPUs. All of its internal algorithms have been layout to efficiently port them onto graphics hardware. This includes porting on the interface level, i.e. interfaces are viewed as the smallest parallelizable entity. This has implications on the utilization of ever growing GPU core arrays, because the number of potential interfaces is quadratic in the number of spheres.

The list of modules to port is compromised of the following:

1. Neighbor-list [atoms]
2. Calculation of spherical interfaces [interfaces]
3. Filtering of spherical interfaces [interfaces]
4. Calculation of boundary segments [interfaces]
5. Surface integration [segments]

In brackets, we have given the smallest parallelizable entity of the respective step. Principally, porting neighbor-list efficiently onto GPUs is a difficult task. However, reference implementations are available[213]. Once the neighbor-list has been created, calculation of spherical interfaces can be performed on a per neighbor-atom basis, i.e. every neighbor will create its interface with S_1. Afterward every interface has to check if it is contained fully in any other interface and potentially buried. If so, it has to remove itself from the process. Calculation of boundary segments is likely the most time consuming part. Here, every interface needs to find out if it contains boundary segments or not, utilizing data from all intersecting interfaces. However, no crosstalk is necessary, and the process can be parallelized. Once the boundary segments have been identified, we move focus from interfaces to these boundaries (which are much fewer). Every segment can integrate itself, again no crosstalk is necessary. Forces can be calculated in the same manner. The problem here is that at the conclusion of the integration, all forces need to be collapsed into atom-wise forces - a process called reduction, which will take additional steps but is not a bottleneck in the computations.

5

DISCUSSION

A multiscale molecular dynamics protocol must consist of the following parts: Two force-fields (one all-atom, the other coarse-grained), a simulator to create trajectories for both, a linkage module to combine one with the other, and analysis tools to extract the quantities in question. In this thesis, we have extended the multiscale protocol from Fan, Warshel and co-workers for the Amber force-field, and we have created AmberCG, a coarse-grained force-field for protein folding based on the Amber series. We have envisioned a new collective variable to help in the post-processing of multiscale trajectories. We have worked on finding better descriptions for the solvent, and created a powerful tessellation software that is able to integrate continuous functional values over surfaces that are bounded by complex spherical geometries and topologies - and as a special case is able to calculate the solvent accessible surface area. And we have discovered a new algorithm for the removal of buried and superfluous neighbours in the discipline of calculating such areas, in a field in which recently only 4 such algorithms have existed.

Things could be better in molecular dynamics. Simulations could be faster, simpler to handle, especially for people without prior knowledge in the field. All-atom force-fields are powerful, and, except for the even more precise quantum mechanical calculations, they are the most accurate tool that we have today to study micro-molecular phenomena. But with these details come problems. Since more than 30 years, researchers try to find just the right parametrizations for the ever-same potentials. Moreover, while the force-fields become better and better, it feels like a struggle of finding the right balance between mutual-exclusive objectives. It is not surprising that force-fields are chosen based on the structure under study, because they tend to bias secondary and tertiary structure. It is the intractable sum of interactions, that needs to be in perfect harmony, which cause tendencies in a force-field that are difficult to control.

Most interesting molecular quantities are statistical by nature, and therefore intrinsically averaged. When discrete detail does not matter, coarse-grained force-fields are a step towards the right goal. They remove the water, reshape whole side-chains into giant balls, and throw away countless interactions. With these modifications, some problems disappear: stalled simulations that are trapped in local minima, frustration, solvent relaxation. Detail is replaced by potentials of mean force, with the potential ability to optimize the mean forces directly - a top to down approach in comparison to the bottom up approach of explicit force-fields.

Again, when discreteness does not matter, coarse-grained force-fields have the potential to provide better results, through simplification of the model. This can be demonstrated in an example. Let us suppose someone dropped a ball of known mass down a vacuum tube. The force acting on the ball is gravitation, and a simple formula exists to express it:

$$\boldsymbol{F} = \mathfrak{m}\boldsymbol{g} \tag{75}$$

In which $\boldsymbol{F}$ is the net force acting on the ball, $\mathfrak{m}$ its mass and $\boldsymbol{g}$ the vector and magnitude of the direction of gravitation. A simulation utilizing this force is incredibly cheap to compute and an accurate description of what happens to the ball. With this simplification, running a simulation will not yield insight into the dynamics of every atom of the ball, but if the question we were asking is how quickly the ball would reach the ground, it is more precise and more feasible to compute than calculating the trajectory explicitly all-atom.

Of course, we shouldn't oversimplify. We are interested in the dynamics of the atoms of proteins, but we are not so much interested in single conformations. Phase space is compromised by clusters of conformations, weighted by the probability of their occurrence. The goal is to sample all of phase space with a coarse-grained force-field, ideally with the same distribution of an all-atom forcefield. This requirement can be relaxed: the two just need to be compatible, such that regions of high probability in one do not correspond to regions of low probability of the other and vice-versa.

If we can assure these criteria, then the error of sampling in the wrong distribution can be corrected through free energy perturbation or similar techniques. In chapter 4.3 we have seen that this is possible for small peptides. An alpha helix and a beta hairpin were successfully folded through the multiscale protocol. It was shown that the coarse-

grained force-field AmberCG yields good energy correlations to all-atom force-fields for a series of different folds - but it is not perfect. Some correlations are just above 0.5, and as we were able to see in our folding study of the 35 residue segment of the chicken villin headpiece, this disagreement does matter. AmberCG induces the correct meta-fold, but is yet unable to stabilize the secondary structure to fully collapse into the native basin. Obviously, more reparametrization runs need to be performed, which must include more complicated folds to release the observed frustration.

Free energy surfaces can be created in various ways, optimally with coordinates that have a physical meaning, are orthogonal, are readily calculated, separate well the product, reactant and transition states, and cause the transition state distribution to be sharply peaked. It is not an easy task, because some goals seem to be mutually repulsive. Perfect orthogonal coordinates can be calculated by principal component analysis - which results in nonphysical descriptions. Coordinates based on transition path sampling, including p_{fold}, are computationally and temporally demanding. Throughout the thesis, we have mainly used two coordinates. Lcpfold, which is based on contact maps and aims to improve separation of product and reactant states, and radius of gyration, which is a totally intrinsic measurement based on the expansion of the protein.

Lcpfold performs a linearization of the presumed folding trajectory by simultaneously comparing the ability of a protein to fold into a (native) reference structure, or to lose its established secondary and tertiary composition. We have shown that it surpasses native contacts when the separation of the equilibrium distributions are compared against each other. However, we did not address the matter of improving the shape of the transition path distribution. This would require transition path sampling, and analysis of its data, while none was available from the source trajectories that we used in our comparison.

While force-field models and solute representations are an important part in coarse-graining, solvent descriptions, and the effect of water need to be addressed as well. Although non-polar solvation free energy is smaller in comparison to its polar counterpart, it has important implications to the effect of hydration. Van der Waals interactions, induced by fluctuating and temporal polarizations, are weak, but their influence grows with the area of the dielectric boundary, and can be strong enough to enable geckos to hanging from walls or ceilings - not an effect of hydration, but an illustration to make the point. In protein

folding, both states, native and denatured are energetically speaking not so far apart. Enthalpically, the native state is very advantageous, offering protection from the water to the hydrophobic core, internal and external hydrogen bonds, secondary structure synergies, electrostatic balance, but it is entropically situated in a disadvantageous part of phase space. The side chains are tightly packed against each other, and the majority of the degrees of freedom is lost. In such a scenario, the difference in geometry, e.g. between a flat structure and a round structure, and its effect on the accessibility of solute atoms with solvent atoms, and their van der Waals interactions, can play a crucial role.

Due to its good correlation to the solvent accessible surface area (at least for cycloalkanes), there are not many non-polar solvation methods out there, and the ones that are available, have the same functional form and differ only in their parametrization. All of these models have in common, that they do not take the geometry of the structure under investigation into account. Semi-Explicit Assembly, a method envisioned by Christopher J. Fennel and Ken A. Dill, is able to do so. It calculates an improved non-polar boundary around the molecule, such that its geometry is explicitly taken into account. The method only has one flaw. The non-polar boundary is calculated as the minimum of the Lennard-Jones field emanating from the solute with respect to the water, which is itself a two dimensional manifold surrounding the solute. Finding this manifold is difficult. The obvious solution is its approximation through a scanning of the field with rays. This is not unproblematic. Firstly, it is expensive, secondly, the dependence on rays and on-the-fly optimization allows the calculation of only numerical derivatives - too slow for molecular dynamics.

As an answer, we created TRIFORCE. A machinery to integrate functional values over surface areas “in real time”. It uses a set of precomputed tables to store functional values and their derivatives. In the presented case in chapter4.4, the functional value is 1, which causes calculation of surface areas. In the case of semi-analytical Semi-Explicit Assembly, the functional values represent segments of the manifold surrounding just one atom S_l, ignoring all neighbors except one. Through a series of iterations over all the neighbors, the true manifold is constructed.

We have shown, that in the example of surface areas, TRIFORCE is almost as accurate as the fully analytical solution GETAREA, and much more accurate than the widely used algorithm LCPO. Even

though we couldn't perform any speed benchmarks with none of them, mainly because of their unavailability as stand-alone applications, TRIFORCE is likely to outperform GETAREA due to its reduced set of algorithmics, and the possibility to port it onto GPUs. It however will not outperform LCPO in terms of speed, the improved accuracy must be paid with this price. One important question is, what is more important, accuracy or speed? The question is ongoing in the scientific community. Fact is that speed is very important, up to the point were inaccuracies would cause the sampling of incorrect distributions. Forces derived from LCPO are very coarse. They principally point into the right hemisphere (r~0.6), but otherwise do not make much sense. It is arguable that they cause more frustration than synergy.

TRIFORCE spends around 90 % of its time calculating so called intersection-points. Special points on the surface of a sphere, marking the interconnection between spherical arcs which make up the boundary of the exposed area. Without this "warming up", LCPO would not be faster than TRIFORCE. How to accelerate the calculation of these points? A first step is the removal of buried atoms, because these atoms would be included in their calculations, but naturally would not posses an exposed surface, and as such no intersection-points. 4 algorithms exist to date to perform such task, 3 have been introduced in the introduction and the other is a precursor to one of the three. Two of the methods are analytical, the first, based on half-spaces, extremely elegant, yet not robust. It fails in certain toplogies, where atoms reside in the van der Waals radius of another atom by more than 50 %. The second, based on geometrical considerations of three and four body intersections, and their implications on buried atoms. The thrid and fourth aproximate algorithms based on neighbor densities estimated by a gaussian kernel.

All the methods have one goal: to speed-up post-calculations by removing these buried atoms. To remove the atoms, time has to be invested. If the time exceeds the amount that is saved, the method is useless. There's also a fine line between accuracy and the number of atoms to be removed. The analytical versions generally will only remove buried atoms, but might not catch all of them. The approximate versions might also remove non-buried atoms, but arguably do it faster. We have introduced a new algorithm which is approximative, that let's the user decide the balance beetween accuracy and speed-up. In the approach, a special grid is drawn over the surface of a sphere. If neighbor spheres intersect with that grid, the intersection is saved and

used to identify buried atoms. However, if intersections fall between the grid, i.e. if the grid is too coarse to accurately catch all intersection, some non-buried atoms will be removed as well. On the good side, the user can chose the grid size and with it the maximum error he is willing to accept. Furthermore, it will not ust remove buried atoms, but all atoms that do not contribute to the boundary of exposed surface areas, and are simply balast for methods based on the Gauss-Bonnet Theorem. On the bad side, with a very large grid, the method itself becomes more expensive, possibly negating what is saved through the removal of the atoms.

All of the methods contained in this thesis are part of the multiscale protocol. Some of them were specifically designed to work cooperativly, others can be embedded in parts of the protocol in which they were yet excluded. This is the case for TRIFORCE, which could be linked within the framework of a coarse-grained force-field. In that context, the multiscale protocol is very flexible. It allows changing components in all places. Starting with the force-field functions, the parameters to optimize, to the parameters of the genetic algorithm, different free energy integration methods and collective coordinates, solvation models, and atom removal algorithms. We have shown, that the protocol works on a number of examples. The multiscale protocol must be understood as a constant cycle, an iterative pipeline, in which results are generated, analyzed and improved through resubmission. It it the foundation that will enable new research in many ways, because new methodologies can be tested, used and build on top of it.

6

CONCLUSIONS

The main contributions of this thesis can be summarized as follows:

1. A Multiscale Molecular Dynamics protocol has been established, implemented, and tested in all of its parts, which enables the study of protein dynamics in a simplified and accelerated form.

2. AmberCG v1.0, a coarse-grained force-field, based on the Amber series of force-fields has been parameterized, and shown to be able to fold small peptides, and to provide reasonable energies for larger proteins. The framework can be used to reparameterize the force-field to be some day able to fold larger proteins.

3. A new collective variable, based on contact maps, designed to linearize the phase space in order to separate product and reactant states was envisioned. The separation has been found to be superior to contact maps and RMSD and suggests that the method can be used instead of the two examples, in studies where separation is desireable.

4. TRIFORCE, a library to perform on-the-fly integrations of functional terms and their derivatives, on spheres bounded by complex solvent accessible surface areas, has been invented, implemented, and tested. It achieves comparable accuracy to full analytical solutions, much better accuracies than an approximative technique, and is designed to run on GPUs and be generally be faster than previous analytical methods, by precalculating expensive quantities.

5. A novel algorithm based on one dimensional Depth-Buffers was envisioned and developed to identify the atoms contributing to the solvent accessible surface area, with the potential to speed up methods based on Gauss-Bonnet Paths. The method was found to outperform existing methods, mainly because it allows more atoms to be removed from any post-processing.

A

MATHEMATICA NOTEBOOKS (TRIFORCE)

This chapter contains Mathematica notebooks, which, once inserted into Mathematica, provide key quantities that were used in the TRIFORCE method. These quantities compromise mainly surface integrals, which were saved in integration tables, and are used "as is" in the source-code and in the article, and might provide helpful insight. Integration limits, dependent on the shape of the triangular regions, are given at the end of this chapter.

Spherical Surface Integrals

Matrix preparations

```
Needs["VectorAnalysis`"]
```

- **unit vector**

```
v := {1, 0, 0}
```

- **rotation around z axis**

```
rotz[α_] := {{Cos[α], -Sin[α], 0}, {Sin[α], Cos[α], 0}, {0, 0, 1}}
```

```
rotz[θ] // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\theta] & -\text{Sin}[\theta] & 0 \\ \text{Sin}[\theta] & \text{Cos}[\theta] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **rotation around x-axis**

```
rotx[α_] := {{1, 0, 0}, {0, Cos[α], -Sin[α]}, {0, Sin[α], Cos[α]}}
```

```
rotx[φ] // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \text{Cos}[\varphi] & -\text{Sin}[\varphi] \\ 0 & \text{Sin}[\varphi] & \text{Cos}[\varphi] \end{pmatrix}$$

Surface Area Integral

- **vector to the surface points, dependent on the angles θ and φ**

```
(f = rotx[φ].rotz[θ].v) // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\theta] \\ \text{Cos}[\varphi]\,\text{Sin}[\theta] \\ \text{Sin}[\theta]\,\text{Sin}[\varphi] \end{pmatrix}$$

- **derivative with respect to θ**

```
(dfθ = D[f, θ]) // MatrixForm
```

$$\begin{pmatrix} -\text{Sin}[\theta] \\ \text{Cos}[\theta]\,\text{Cos}[\varphi] \\ \text{Cos}[\theta]\,\text{Sin}[\varphi] \end{pmatrix}$$

- **derivative with respect to φ**

```
(dfφ = D[f, φ]) // MatrixForm
```

$$\begin{pmatrix} 0 \\ -\text{Sin}[\theta]\,\text{Sin}[\varphi] \\ \text{Cos}[\varphi]\,\text{Sin}[\theta] \end{pmatrix}$$

- **magnitude of the normal vector to the surface points**

```
(c = CrossProduct[dfθ, dfφ]) // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\theta]\,\text{Cos}[\varphi]^2\,\text{Sin}[\theta] + \text{Cos}[\theta]\,\text{Sin}[\theta]\,\text{Sin}[\varphi]^2 \\ \text{Cos}[\varphi]\,\text{Sin}[\theta]^2 \\ \text{Sin}[\theta]^2\,\text{Sin}[\varphi] \end{pmatrix}$$

```
m = FullSimplify[Sqrt[c.c], Assumptions → θ ≥ 0 && θ ≤ Pi]
```

$$\text{Sin}[\theta]$$

- **for the surface integration, the shape of the circular interface is important, i.e. how far θ streches for each φ angle. The angle between the circular interface and the integration axis is given by ψ, its opening angle is given by λ.**

```
(n = rotz[ψ].v ) // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\psi] \\ \text{Sin}[\psi] \\ 0 \end{pmatrix}$$

```
FullSimplify[Quiet[Solve[λ == ArcCos[(f.n)], θ]]]
```

$$\left\{\left\{\theta \to -\text{ArcCos}\left[\left(\text{Cos}[\lambda]\,\text{Cos}[\psi] - \sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\left(-\text{Cos}[\lambda]^2 + \text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)}\right) \Big/ \left(\text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)\right]\right\},\right.$$

$$\left\{\theta \to \text{ArcCos}\left[\frac{\text{Cos}[\lambda]\,\text{Cos}[\psi] - \sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\left(-\text{Cos}[\lambda]^2 + \text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)}}{\text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2}\right]\right\},$$

$$\left\{\theta \to -\text{ArcCos}\left[\left(\text{Cos}[\lambda]\,\text{Cos}[\psi] + \sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\left(-\text{Cos}[\lambda]^2 + \text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)}\right) \Big/ \left(\text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)\right]\right\},$$

$$\left.\left\{\theta \to \text{ArcCos}\left[\frac{\text{Cos}[\lambda]\,\text{Cos}[\psi] + \sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\left(-\text{Cos}[\lambda]^2 + \text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)}}{\text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2}\right]\right\}\right\}$$

- **thetaConcave and thetaConvex give the maximal θ values per φ angle until a line on the surface would hit the end of the either convex or concave circular interface**

```
thetaConcave[φ_, ψ_, λ_] :=
 ArcCos[((Cos[λ] Cos[ψ] + Sqrt[Cos[φ]^2 Sin[ψ]^2 (-Cos[λ]^2 + Cos[ψ]^2 + Cos[φ]^2 Sin[ψ]^2)])/
    (Cos[ψ]^2 + Cos[φ]^2 Sin[ψ]^2))]
thetaConvex[φ_, ψ_, λ_] :=
 ArcCos[(Cos[λ] Cos[ψ] - Sqrt[Cos[φ]^2 Sin[ψ]^2 (-Cos[λ]^2 + Cos[ψ]^2 + Cos[φ]^2 Sin[ψ]^2)])/
   (Cos[ψ]^2 + Cos[φ]^2 Sin[ψ]^2)]
```

the integration can analytically only be performed over θ angles. For φ angles, the integral is evaluated numerically

- **the integral with respect to θ for a concave circular region**

```
FullSimplify[Integrate[m, {θ, 0, thetaConcave[φ, ψ, λ]}]]
```

$$1 - \frac{\text{Cos}[\lambda]\,\text{Cos}[\psi] + \sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\left(-\text{Cos}[\lambda]^2 + \text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)}}{\text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2}$$

- **the integral with respect to θ for a convex circular interface**

```
FullSimplify[Integrate[m, {θ, 0, thetaConvex[φ, ψ, λ]}]]
```

$$1 + \frac{-\text{Cos}[\lambda]\,\text{Cos}[\psi] + \sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\left(-\text{Cos}[\lambda]^2 + \text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)}}{\text{Cos}[\psi]^2 + \text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2}$$

- **this is how it looks**

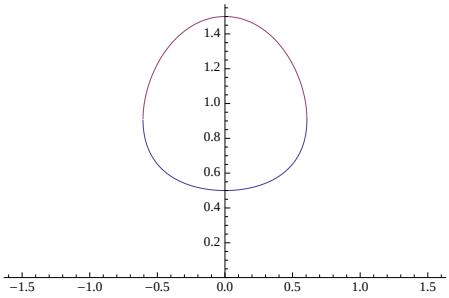
```
psi = 1
```

```
1
```

```
lambda = 0.5
```

```
0.5
```

```
Plot[{thetaConcave[x, psi, lambda], thetaConvex[x, psi, lambda]},
 {x, -Pi / 2, Pi / 2}, PlotRange → {0, Pi / 2}]
```



Integration Limits

Matrix preparations

```
Needs["VectorAnalysis`"]
```

- **unit vector**

```
ex := {1, 0, 0}
```

```
nOrigin := {0, 0, 1}
```

- **rotation around z axis**

```
rotz[α_] := {{Cos[α], -Sin[α], 0}, {Sin[α], Cos[α], 0}, {0, 0, 1}}
```

```
rotz[θ] // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\theta] & -\text{Sin}[\theta] & 0 \\ \text{Sin}[\theta] & \text{Cos}[\theta] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **rotation around x-axis**

```
rotx[α_] := {{1, 0, 0}, {0, Cos[α], -Sin[α]}, {0, Sin[α], Cos[α]}}
```

```
rotx[φ] // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \text{Cos}[\varphi] & -\text{Sin}[\varphi] \\ 0 & \text{Sin}[\varphi] & \text{Cos}[\varphi] \end{pmatrix}$$

PHI2phi conversion function

- **"bottom" of circular region**

```
border = rotz[-λ].ex
```

```
{Cos[λ], -Sin[λ], 0}
```

- **center of circular region**

```
v = rotz[-ψ].ex
```

```
{Cos[ψ], -Sin[ψ], 0}
```

- **"base" vector for the circular region**

```
a = border - ex
```

```
{-1 + Cos[λ], -Sin[λ], 0}
```

vector from center of circular region to intersection point

```
aRot = rotx[Φ].a
```

```
{-1 + Cos[λ], -Cos[Φ] Sin[λ], -Sin[λ] Sin[Φ]}
```

- **vector from origin to intersection point**

```
ip = ex + aRot
```

```
{Cos[λ], -Cos[Φ] Sin[λ], -Sin[λ] Sin[Φ]}
```

- **perpendicular normal vector to intersection point and center of circular region**

```
nIntersection = CrossProduct[v, ip]
```

```
{Sin[λ] Sin[Φ] Sin[ψ], Cos[ψ] Sin[λ] Sin[Φ], -Cos[Φ] Cos[ψ] Sin[λ] + Cos[λ] Sin[ψ]}
```

```
nIntersection = Normalize[nIntersection]
```

```
{(Sin[λ] Sin[Φ] Sin[ψ]) / (√(Abs[Cos[ψ] Sin[λ] Sin[Φ]]^2 +
      Abs[Sin[λ] Sin[Φ] Sin[ψ]]^2 + Abs[-Cos[Φ] Cos[ψ] Sin[λ] + Cos[λ] Sin[ψ]]^2)),
 (Cos[ψ] Sin[λ] Sin[Φ]) / (√(Abs[Cos[ψ] Sin[λ] Sin[Φ]]^2 + Abs[Sin[λ] Sin[Φ] Sin[ψ]]^2 +
      Abs[-Cos[Φ] Cos[ψ] Sin[λ] + Cos[λ] Sin[ψ]]^2)),
 (-Cos[Φ] Cos[ψ] Sin[λ] + Cos[λ] Sin[ψ]) / (√(Abs[Cos[ψ] Sin[λ] Sin[Φ]]^2 +
      Abs[Sin[λ] Sin[Φ] Sin[ψ]]^2 + Abs[-Cos[Φ] Cos[ψ] Sin[λ] + Cos[λ] Sin[ψ]]^2))}
```

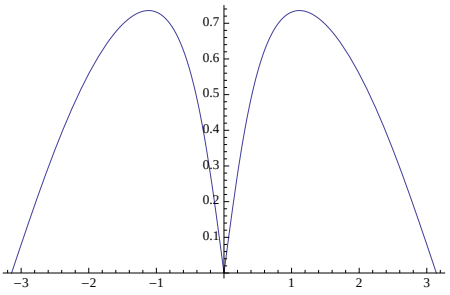
- **PHI2phi function gives angle between normal vector and x-y-plane**

```
PHI2phi[Φ_, ψ_, λ_] = FullSimplify[ArcCos[DotProduct[nIntersection, nOrigin]]]
```

```
ArcCos[(-Cos[Φ] Cos[ψ] Sin[λ] + Cos[λ] Sin[ψ]) /
   (√(Abs[Cos[ψ] Sin[λ] Sin[Φ]]^2 + Abs[Sin[λ] Sin[Φ] Sin[ψ]]^2 +
      Abs[Cos[Φ] Cos[ψ] Sin[λ] - Cos[λ] Sin[ψ]]^2))]
```

- **this is how it looks**

```
Plot[PHI2phi[x, 1, 0.6], {x, -Pi, Pi}]
```



phi2PHI inverse function

```
FullSimplify[Quiet[Solve[PHI2phi[Φ, ψ, λ] == φ, Φ]]]
```

Solve::ifun :
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. >>

$$\left\{\left\{\Phi \to -\text{ArcSec}\left[\left(2\left(\text{Cos}[\psi]^2+\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)\right)\Big/\left(\sqrt{2}\;\text{Csc}[\lambda]^2\sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\lambda]^2\left(-\text{Cos}[2\lambda]+\text{Cos}[\psi]^2+\text{Cos}[2\varphi]\,\text{Sin}[\psi]^2\right)}+\text{Cot}[\lambda]\,\text{Sin}[\varphi]^2\,\text{Sin}[2\psi]\right)\right]\right\},\right.$$

$$\left\{\Phi \to \text{ArcSec}\left[\left(2\left(\text{Cos}[\psi]^2+\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)\right)\Big/\left(\sqrt{2}\;\text{Csc}[\lambda]^2\sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\lambda]^2\left(-\text{Cos}[2\lambda]+\text{Cos}[\psi]^2+\text{Cos}[2\varphi]\,\text{Sin}[\psi]^2\right)}+\text{Cot}[\lambda]\,\text{Sin}[\varphi]^2\,\text{Sin}[2\psi]\right)\right]\right\},$$

$$\left\{\Phi \to -\text{ArcSec}\left[\left(2\left(\text{Cos}[\psi]^2+\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)\right)\Big/\left(-\sqrt{2}\;\text{Csc}[\lambda]^2\sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\lambda]^2\left(-\text{Cos}[2\lambda]+\text{Cos}[\psi]^2+\text{Cos}[2\varphi]\,\text{Sin}[\psi]^2\right)}+\text{Cot}[\lambda]\,\text{Sin}[\varphi]^2\,\text{Sin}[2\psi]\right)\right]\right\},$$

$$\left.\left\{\Phi \to \text{ArcSec}\left[\left(2\left(\text{Cos}[\psi]^2+\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)\right)\Big/\left(-\sqrt{2}\;\text{Csc}[\lambda]^2\sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\lambda]^2\left(-\text{Cos}[2\lambda]+\text{Cos}[\psi]^2+\text{Cos}[2\varphi]\,\text{Sin}[\psi]^2\right)}+\text{Cot}[\lambda]\,\text{Sin}[\varphi]^2\,\text{Sin}[2\psi]\right)\right]\right\}\right\}$$

```
phi2PHI[φ_, ψ_, λ_] = ArcSec[(2 (Cos[ψ]^2 + Cos[φ]^2 Sin[ψ]^2)) /
    (-Sqrt[2] Csc[λ]^2 Sqrt[Cos[φ]^2 Sin[λ]^2 (-Cos[2 λ] + Cos[ψ]^2 + Cos[2 φ] Sin[ψ]^2)] +
      Cot[λ] Sin[φ]^2 Sin[2 ψ])]
```

$$\text{ArcSec}\left[\left(2\left(\text{Cos}[\psi]^2+\text{Cos}[\varphi]^2\,\text{Sin}[\psi]^2\right)\right)\Big/\left(-\sqrt{2}\;\text{Csc}[\lambda]^2\sqrt{\text{Cos}[\varphi]^2\,\text{Sin}[\lambda]^2\left(-\text{Cos}[2\lambda]+\text{Cos}[\psi]^2+\text{Cos}[2\varphi]\,\text{Sin}[\psi]^2\right)}+\text{Cot}[\lambda]\,\text{Sin}[\varphi]^2\,\text{Sin}[2\psi]\right)\right]$$

phiLimit

- **φ limits are reached when the square root of both functions thetaConvex and thetaConcave of 'surfaceIntegral.nb' vanishes:**

```
Quiet[Solve[Cos[φ]^2 Sin[ψ]^2 (-Cos[λ]^2 + Cos[ψ]^2 + Cos[φ]^2 Sin[ψ]^2) == 0, φ]]
```

$$\left\{\left\{\varphi \to -\frac{\pi}{2}\right\},\ \left\{\varphi \to \frac{\pi}{2}\right\},\ \left\{\varphi \to -\text{ArcCos}\left[\sqrt{(\text{Cos}[\lambda]-\text{Cos}[\psi])\,(\text{Cos}[\lambda]+\text{Cos}[\psi])\,\text{Csc}[\psi]^2}\right]\right\},\right.$$

$$\left\{\varphi \to \text{ArcCos}\left[\sqrt{(\text{Cos}[\lambda]-\text{Cos}[\psi])\,(\text{Cos}[\lambda]+\text{Cos}[\psi])\,\text{Csc}[\psi]^2}\right]\right\},$$

$$\left\{\varphi \to -\text{ArcCos}\left[-\sqrt{(\text{Cos}[\lambda]^2-\text{Cos}[\psi]^2)\,\text{Csc}[\psi]^2}\right]\right\},\ \left.\left\{\varphi \to \text{ArcCos}\left[-\sqrt{(\text{Cos}[\lambda]^2-\text{Cos}[\psi]^2)\,\text{Csc}[\psi]^2}\right]\right\}\right\}$$

the 3rd and 4th solution give the limits for negative φ and positive φ respectively. We use only the positive solution

```
phiLimit[ψ_, λ_] = FullSimplify[ArcCos[Sqrt[(Cos[λ] - Cos[ψ]) (Cos[λ] + Cos[ψ]) Csc[ψ]^2]]]
```

```
ArcCos[Sqrt[(Cos[λ]^2 - Cos[ψ]^2) Csc[ψ]^2]]
```

PHILimit for $\psi+\lambda < \pi$

```
FullSimplify[phi2PHI[phiLimit[ψ, λ], ψ, λ]]
```

```
ArcSec[Cot[λ] Tan[ψ]]
```

```
PHILimit0[ψ_, λ_] = ArcSec[Cot[λ] Tan[ψ]]
```

```
ArcSec[Cot[λ] Tan[ψ]]
```

PHILimit for $\psi+\lambda >= \pi$

- **the limit for φ is constantly $\pi/2$ for this range of values**

```
FullSimplify[phi2PHI[Pi / 2, ψ, λ]]
```

```
ArcSec[Cot[ψ] Tan[λ]]
```

BIBLIOGRAPHY

[1] BJ Alder and TEf Wainwright. Phase transition for a hard sphere system. *The Journal of Chemical Physics*, 27(5):1208–1209, 1957.

[2] E. Meeron. *Physics of Many-particle Systems: Methods and Problems*. Number v. 1 in Many-body problem. Gordon and Breach, 1964.

[3] A Rahman. Correlations in the motion of atoms in liquid argon. *phys. Rev*, 136 (2A):405–411, 1964.

[4] Shneior Lifson and Arieh Warshel. Consistent force field for calculations of conformations, vibrational spectra, and enthalpies of cycloalkane and n-alkane molecules. *The Journal of Chemical Physics*, 49:5116, 1968.

[5] Michael Levitt. The birth of computational structural biology. *Nature Structural & Molecular Biology*, 8(5):392–393, 2001.

[6] M Bixon and S Lifson. Potential functions and conformations in cycloalkanes. *Tetrahedron*, 23(2):769–784, 1967.

[7] L. S. Bartell and D. A. Kohl. Structure and rotational isomerization of free hydrocarbon chains. *The Journal of Chemical Physics*, 39(11):3097–3105, 1963. doi: 10.1063/1.1734149.

[8] James B. Hendrickson. Molecular geometry. i. machine computation of the common rings. *Journal of the American Chemical Society*, 83(22):4537–4547, 1961. doi: 10.1021/ja01483a011.

[9] Kenneth B. Wiberg. A scheme for strain energy minimization. application to the cycloalkanes1. *Journal of the American Chemical Society*, 87(5):1070–1078, 1965. doi: 10.1021/ja01083a024.

[10] Kenneth S. Pitzer. Potential energies for rotation about single bonds. *Discuss. Faraday Soc.*, 10:66–73, 1951. doi: 10.1039/DF9511000066.

[11] A Warshel, M Levitt, and S Lifson. Consistent force field for calculation of vibrational spectra and conformations of some amides and lactam rings. *Journal of Molecular Spectroscopy*, 33(1):84–99, 1970.

[12] A Warshel and M Karplus. Calculation of ground and excited state potential surfaces of conjugated molecules. i. formulation and parametrization. *Journal of the American Chemical Society*, 94(16):5612–5625, 1972.

[13] Arieh Warshel and Michael Levitt. Theoretical studies of enzymic reactions: dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *Journal of molecular biology*, 103(2):227–249, 1976.

[14] Arieh Warshel and Robert M Weiss. An empirical valence bond approach for comparing reactions in solutions and in enzymes. *Journal of the American Chemical Society*, 102(20):6218–6226, 1980.

[15] Aneesur Rahman and Frank H. Stillinger. Molecular dynamics study of liquid water. *The Journal of Chemical Physics*, 55(7):3336–3359, 1971. doi: 10.1063/1.1676585.

[16] Frank H. Stillinger and Aneesur Rahman. Molecular dynamics study of temperature effects on water structure and kinetics. *The Journal of Chemical Physics*, 57(3):1281–1292, 1972. doi: 10.1063/1.1678388.

[17] Frank H. Stillinger and Aneesur Rahman. Improved simulation of liquid water by molecular dynamics. *The Journal of Chemical Physics*, 60(4):1545–1557, 1974. doi: 10.1063/1.1681229.

[18] J Andrew McCammon. Dynamics of folded proteins. *Nature*, 267:16, 1977.

[19] Martin Karplus. Molecular dynamics of biological macromolecules: A brief history and perspective. *Biopolymers*, 68(3):350–358, 2003.

[20] Michael Levitt, Arieh Warshel, et al. Computer simulation of protein folding. *Nature*, 253(5494):694, 1975.

[21] Arieh Warshel. Bicycle-pedal model for the first step in the vision process. *Nature (London)*, 260:678–683, 1976.

[22] B R Gelin and M Karplus. Sidechain torsional potentials and motion of amino acids in porteins: bovine pancreatic trypsin inhibitor. *Proceedings of the National Academy of Sciences*, 72(6):2002–2006, 1975.

[23] DA Case and M Karplus. Dynamics of ligand binding to heme proteins. *Journal of molecular biology*, 132(3):343–368, 1979.

[24] Glenn M Torrie and John P Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, 1977.

[25] S H Northrup, M R Pear, C Y Lee, J A McCammon, and M Karplus. Dynamical theory of activated processes in globular proteins. *Proceedings of the National Academy of Sciences*, 79(13):4035–4039, 1982.

[26] Shankar Kumar, John M Rosenberg, Djamal Bouzida, Robert H Swendsen, and Peter A Kollman. The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *Journal of Computational Chemistry*, 13(8):1011–1021, 1992.

[27] M Levitt and R Sharon. Accurate simulation of protein dynamics in solution. *Proceedings of the National Academy of Sciences*, 85(20):7557–7561, 1988.

[28] Jenn Kang Hwang and Arieh Warshel. Microscopic examination of free-energy relationships for electron transfer in polar solvents. *Journal of the American Chemical Society*, 109(3):715–720, 1987.

[29] A Warshel and G King. Polarization constraints in molecular dynamics simulation of aqueous solutions: the surface constraint all atom solvent (scaas) model. *Chemical physics letters*, 121(1):124–129, 1985.

[30] Lisa Pollack. Fashioning NAMD: A History of Risk and Reward. *https://www-s.ks.uiuc.edu/History/NAMD/*, 2012.

[31] Kit Fun Lau and Ken A. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10):3986–3997, 1989. doi: 10.1021/ma00200a030.

[32] H.J.C. Berendsen, D. van der Spoel, and R. van Drunen. Gromacs: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(1-3):43 – 56, 1995. ISSN 0010-4655. doi: http://dx.doi.org/10.1016/0010-4655(95)00042-E.

[33] Herman JC Berendsen, J Pl M Postma, Wilfred F van Gunsteren, ARHJ DiNola, and JR Haak. Molecular dynamics with coupling to an external bath. *The Journal of chemical physics*, 81:3684, 1984.

[34] William G. Hoover, Anthony J. C. Ladd, and Bill Moran. High-strain-rate plastic flow studied via nonequilibrium molecular dynamics. *Phys. Rev. Lett.*, 48:1818–1820, Jun 1982. doi: 10.1103/PhysRevLett.48.1818.

[35] Anthony J. C. Ladd and William G. Hoover. Plastic flow in close-packed crystals via nonequilibrium molecular dynamics. *Phys. Rev. B*, 28:1756–1762, Aug 1983. doi: 10.1103/PhysRevB.28.1756.

[36] Shuichi Nose. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 1984. doi: 10.1063/1.447334.

[37] S. A. Adelman and J. D. Doll. Generalized langevin equation approach for atom/solid-surface scattering: General formulation for classical scattering off harmonic solids. *The Journal of Chemical Physics*, 64(6):2375–2388, 1976. doi: 10.1063/1.432526.

[38] W. Clark Still, Anna Tempczyk, Ronald C. Hawley, and Thomas Hendrickson. Semianalytical treatment of solvation for molecular mechanics and dynamics. *Journal of the American Chemical Society*, 112(16):6127–6129, 1990. doi: 10.1021/ja00172a038.

[39] Kim T Simons, Charles Kooperberg, Enoch Huang, David Baker, et al. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of molecular biology*, 268(1):209–225, 1997.

[40] Christopher Jarzynski. Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78(14):2690, 1997.

[41] Kevin W Plaxco, Kim T Simons, and David Baker. Contact order, transition state placement and the refolding rates of single domain proteins. *Journal of molecular biology*, 277(4):985–994, 1998.

[42] Yong Duan and Peter A. Kollman. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, 282(5389):740–744, 1998. doi: 10.1126/science.282.5389.740.

[43] Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters*, 314(1-2):141 – 151, 1999. ISSN 0009-2614. doi: http://dx.doi.org/10.1016/S0009-2614(99)01123-9.

[44] Brian Kuhlman, Gautam Dantas, Gregory C. Ireton, Gabriele Varani, Barry L. Stoddard, and David Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649):1364–1368, 2003. doi: 10.1126/science.1089427.

[45] Isaiah T Arkin, Huafeng Xu, Morten O Jensen, Eyal Arbely, Estelle R Bennett, Kevin J Bowers, Edmond Chow, Ron O Dror, Michael P Eastwood, Ravenna Flitman-Tene, et al. Mechanism of na+/h+ antiporting. *Science Signaling*, 317 (5839):799, 2007.

[46] Daniela Röthlisberger, Olga Khersonsky, Andrew M Wollacott, Lin Jiang, Jason DeChancie, Jamie Betker, Jasmine L Gallaher, Eric A Althoff, Alexandre Zanghellini, Orly Dym, et al. Kemp elimination catalysts by computational enzyme design. *Nature*, 453(7192):190–195, 2008.

[47] Vincent A Voelz, Gregory R Bowman, Kyle Beauchamp, and Vijay S Pande. Molecular simulation of ab initio protein folding for a millisecond folder ntl9 (1- 39). *Journal of the American Chemical Society*, 132(5):1526–1528, 2010.

[48] Andreas W. Götz, Mark J. Williamson, Dong Xu, Duncan Poole, Scott Le Grand, and Ross C. Walker. Routine microsecond molecular dynamics simulations with amber on gpus. 1. generalized born. *Journal of Chemical Theory and Computation*, 8(5):1542–1555, 2012. doi: 10.1021/ct200909j.

[49] MJ Harvey, G Giupponi, and G De Fabritiis. Acemd: accelerating biomolecular dynamics in the microsecond time scale. *Journal of Chemical Theory and Computation*, 5(6):1632–1639, 2009.

[50] Berk Hess, Carsten Kutzner, David van der Spoel, and Erik Lindahl. Gromacs 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *Journal of Chemical Theory and Computation*, 4(3):435–447, 2008. doi: 10.1021/ct700301q.

[51] Alexey Onufriev, Donald Bashford, and David A Case. Exploring protein native states and large-scale conformational changes with a modified generalized born model. *Proteins: Structure, Function, and Bioinformatics*, 55(2):383–394, 2004.

[52] Christopher J Fennell, Charles W Kehoe, and Ken A Dill. Modeling aqueous solvation with semi-explicit assembly. *Proceedings of the National Academy of Sciences*, 108(8):3234–3239, 2011.

[53] Siewert J Marrink, H Jelger Risselada, Serge Yefimov, D Peter Tieleman, and Alex H de Vries. The martini force field: coarse grained model for biomolecular simulations. *The Journal of Physical Chemistry B*, 111(27):7812–7824, 2007.

[54] Sergei Izvekov and Gregory A. Voth. A multiscale coarse-graining method for biomolecular systems. *The Journal of Physical Chemistry B*, 109(7):2469–2473, 2005. doi: 10.1021/jp044629q. URL `http://pubs.acs.org/doi/abs/10.1021/jp044629q`.

[55] Benjamin M Messer, Maite Roca, Zhen T Chu, Spyridon Vicatos, Alexandra Vardi Kilshtain, and Arieh Warshel. Multiscale simulations of protein landscapes: Using coarse-grained models as reference potentials to full explicit models. *Proteins: Structure, Function, and Bioinformatics*, 78(5):1212–1227, 2010.

[56] Edward Lyman, F Marty Ytreberg, and Daniel M Zuckerman. Resolution exchange simulation. *Physical review letters*, 96(2):028105, 2006.

[57] Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002. doi: 10.1073/pnas.202427399.

[58] Michael Shirts and Vijay S. Pande. Screen savers of the world unite! *Science*, 290(5498):1903–1904, 2000. doi: 10.1126/science.290.5498.1903.

[59] M Harvey, G Giupponi, J Villà-Freixa, and G De Fabritiis. Ps3grid .net: Building a distributed supercomputer using the playstation 3, distributed and grid computing-science made transparent for everyone. *Principles, Applications and Supporting Communities*, 2007.

[60] I Buch, Matt J Harvey, T Giorgino, DP Anderson, and G De Fabritiis. High-throughput all-atom molecular dynamics simulations using distributed computing. *Journal of chemical information and modeling*, 50(3):397–403, 2010.

[61] Rhiju Das, Bin Qian, Srivatsan Raman, Robert Vernon, James Thompson, Philip Bradley, Sagar Khare, Michael D Tyka, Divya Bhat, Dylan Chivian, et al. Structure prediction for casp7 targets using extensive all-atom refinement with rosetta@ home. *Proteins: Structure, Function, and Bioinformatics*, 69(S8):118–128, 2007.

[62] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307): 756–760, 2010.

[63] Robert A Alberty. Principle of detailed balance in kinetics. *Journal of Chemical Education*, 81(8):1206, 2004.

[64] Vasilios I Manousiouthakis and Michael W Deem. Strict detailed balance is unnecessary in monte carlo simulation. *The Journal of chemical physics*, 110:2753, 1999.

[65] Hidemaro Suwa and Synge Todo. Markov chain monte carlo method without detailed balance. *Phys. Rev. Lett.*, 105:120603, Sep 2010. doi: 10.1103/PhysRevLett.105.120603. URL `http://link.aps.org/doi/10.1103/PhysRevLett.105.120603`.

[66] Joseph D. Bryngelson, José Nelson Onuchic, Nicholas D. Socci, and Peter G. Wolynes. Funnels, pathways, and the energy landscape of protein folding: A synthesis. *Proteins: Structure, Function, and Bioinformatics*, 21(3):167–195, 1995. ISSN 1097-0134. doi: 10.1002/prot.340210302. URL `http://dx.doi.org/10.1002/prot.340210302`.

[67] James B Clarage, Tod Romo, B Kim Andrews, B Montgomery Pettitt, and George N Phillips. A sampling problem in molecular dynamics simulations of macromolecules. *Proceedings of the National Academy of Sciences*, 92(8):3288–3292, 1995.

[68] Cyrus Levinthal. Are there pathways for protein folding. *J. Chim. Phys*, 65(1): 44–45, 1968.

[69] F Marty Ytreberg, Robert H Swendsen, and Daniel M Zuckerman. Comparison of free energy methods for molecular systems. *The Journal of chemical physics*, 125:184114, 2006.

[70] Michael R. Shirts and Vijay S. Pande. Comparison of efficiency and bias of free energies computed by exponential averaging, the bennett acceptance ratio, and thermodynamic integration. *The Journal of Chemical Physics*, 122(14): 144107, 2005. doi: 10.1063/1.1873592. URL `http://link.aip.org/link/?JCP/122/144107/1`.

[71] Gavin E Crooks. Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. *Physical Review E*, 60(3):2721, 1999.

[72] F Marty Ytreberg and Daniel M Zuckerman. Single-ensemble nonequilibrium path-sampling estimates of free energy differences. *The Journal of chemical physics*, 120:10876, 2004.

[73] Charles H Bennett. Efficient estimation of free energy differences from monte carlo data. *Journal of Computational Physics*, 22(2):245–268, 1976.

[74] Peter Hunter, Peter V. Coveney, Bernard de Bono, Vanessa Diaz, John Fenner, Alejandro F. Frangi, Peter Harris, Rod Hose, Peter Kohl, Pat Lawford, Keith McCormack, Miriam Mendes, Stig Omholt, Alfio Quarteroni, John Skår, Jesper Tegner, S. Randall Thomas, Ioannis Tollis, Ioannis Tsamardinos, Johannes H. G. M. van Beek, and Marco Viceconti. A vision and strategy for the virtual physiological human in 2010 and beyond. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1920):2595–2614, 2010. doi: 10.1098/rsta.2010.0048.

[75] ZZ Fan, J-K Hwang, and A Warshel. Using simplified protein representation as a reference potential for all-atom calculations of folding free energy. *Theoretical Chemistry Accounts*, 103(1):77–80, 1999.

[76] Hans Martin Senn and Walter Thiel. Qm/mm methods for biomolecular systems. *Angewandte Chemie International Edition*, 48(7):1198–1229, 2009.

[77] Valentina Tozzini. Coarse-grained models for proteins. *Current opinion in structural biology*, 15(2):144–150, 2005.

[78] Hiroshi Taketomi, Yuzo Ueda, and Nobuhiro Gō. Studies on protein folding, unfolding and fluctuations by computer simulation. *International journal of peptide and protein research*, 7(6):445–459, 1975.

[79] Marc Delarue and Y-H Sanejouand. Simplified normal mode analysis of conformational transitions in dna-dependent polymerases: the elastic network model. *Journal of molecular biology*, 320(5):1011–1024, 2002.

[80] Shoji Takada. Coarse-grained molecular simulations of large biomolecules. *Current opinion in structural biology*, 22(2):130–137, 2012.

[81] Michael Levitt. A simplified representation of protein conformations for rapid simulation of protein folding. *Journal of molecular biology*, 104(1):59–107, 1976.

[82] B Smit, PAJ Hilbers, K Esselink, LAM Rupert, NM Van Os, and AG Schlijper. Computer simulations of a water/oil interface in the presence of micelles. *Nature*, 348(6302):624–625, 1990.

[83] Sergei V Krivov and Martin Karplus. Hidden complexity of free energy surfaces for peptide (protein) folding. *Proceedings of the National Academy of Sciences of the United States of America*, 101(41):14766–14770, 2004.

[84] Angel E Garcia. Large-amplitude nonlinear motions in proteins. *Physical review letters*, 68(17):2696, 1992.

[85] Miguel L Teodoro, George N Phillips Jr, and Lydia E Kavraki. Understanding protein flexibility through dimensionality reduction. *Journal of Computational Biology*, 10(3-4):617–634, 2003.

[86] Payel Das, Mark Moll, Hernan Stamati, Lydia E Kavraki, and Cecilia Clementi. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences*, 103(26):9885–9890, 2006.

[87] Robert B Best and Gerhard Hummer. Reaction coordinates and rates from transition paths. *Proceedings of the National Academy of Sciences of the United States of America*, 102(19):6732–6737, 2005.

[88] Rose Du, Vijay S Pande, Alexander Yu Grosberg, Toyoichi Tanaka, and Eugene S Shakhnovich. On the transition coordinate for protein folding. *The Journal of chemical physics*, 108:334, 1998.

[89] Vladimir N Maiorov and Gordon M Crippen. Significance of root-mean-square deviation in comparing three-dimensional structures of globular proteins. *Journal of molecular biology*, 235(2):625–634, 1994.

[90] John Kuszewski, Angela M Gronenborn, and G Marius Clore. Improving the packing and accuracy of nmr structures with a pseudopotential for the radius of gyration. *Journal of the American Chemical Society*, 121(10):2337–2338, 1999.

[91] David Chandler. Interfaces and the driving force of hydrophobic assembly. *Nature*, 437(7059):640–647, 2005.

[92] Sowmianarayanan Rajamani, Thomas M Truskett, and Shekhar Garde. Hydrophobic hydration from small to large lengthscales: Understanding and manipulating the crossover. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9475–9480, 2005.

[93] X Huang, CJ Margulis, and BJ Berne. Dewetting-induced collapse of hydrophobic particles. *Proceedings of the National Academy of Sciences*, 100(21):11953–11958, 2003.

[94] Thomas Simonson and Charles L Brooks. Charge screening and the dielectric constant of proteins: insights from molecular dynamics. *Journal of the American Chemical Society*, 118(35):8452–8458, 1996.

[95] Peter H Poole, Francesco Sciortino, Tor Grande, H Eugene Stanley, and C Austen Angell. Effect of hydrogen bonds on the thermodynamic behavior of liquid water. *Physical review letters*, 73(12):1632, 1994.

[96] David van der Spoel, Paul J van Maaren, and Herman JC Berendsen. A systematic study of water models for molecular simulation: Derivation of water models optimized for use with a reaction field. *The Journal of chemical physics*, 108:10220, 1998.

[97] Frank H Stillinger and Aneesur Rahman. Improved simulation of liquid water by molecular dynamics. *The Journal of Chemical Physics*, 60:1545, 1974.

[98] William L Jorgensen, Jayaraman Chandrasekhar, Jeffry D Madura, Roger W Impey, and Michael L Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of chemical physics*, 79:926, 1983.

[99] William L Jorgensen. Quantum and statistical mechanical studies of liquids. 10. transferable intermolecular potential functions for water, alcohols, and ethers. application to liquid water. *Journal of the American Chemical Society*, 103(2):335–340, 1981.

[100] Pekka Mark and Lennart Nilsson. Structure and dynamics of the tip3p, spc, and spc/e water models at 298 k. *The Journal of Physical Chemistry A*, 105(43): 9954–9960, 2001.

[101] Jan Zielkiewicz. Structural properties of water: Comparison of the spc, spce, tip4p, and tip5p models of water. *The Journal of chemical physics*, 123:104501, 2005.

[102] G Makov and MC Payne. Periodic boundary conditions in ab initio calculations. *Physical Review B*, 51(7):4014, 1995.

[103] E Spohr. Effect of electrostatic boundary conditions and system size on the interfacial properties of water and aqueous solutions. *The Journal of chemical physics*, 107:6342, 1997.

[104] Lawrence R Pratt and Steven W Haan. Effects of periodic boundary conditions on equilibrium properties of computer simulated fluids. i. theory. *The Journal of Chemical Physics*, 74:1864, 1981.

[105] Gregory King and Arieh Warshel. A surface constrained all-atom solvent model for effective simulations of polar solutions. *The Journal of Chemical Physics*, 91: 3647, 1989.

[106] Ronald M Levy, Linda Y Zhang, Emilio Gallicchio, and Anthony K Felts. On the nonpolar hydration free energy of proteins: surface area and continuum solvent models for the solute-solvent interaction energy. *Journal of the American Chemical Society*, 125(31):9523–9530, 2003.

[107] Ingo Muegge, Holly Tao, and Arieh Warshel. A fast estimate of electrostatic group contributions to the free energy of protein-inhibitor binding. *Protein engineering*, 10(12):1363–1372, 1997.

[108] Yuk Yin Sham, Zhen Tao Chu, Holly Tao, and Arieh Warshel. Examining methods for calculations of binding free energies: Lra, lie, pdld-lra, and pdld/s-lra calculations of ligands binding to an hiv protease. *Proteins: Structure, Function, and Bioinformatics*, 39(4):393–407, 2000.

[109] Arieh Warshel, Stephen T Russell, et al. *Calculations of electrostatic interactions in biological systems and in solutions*. Cambridge Univ Press, 1984.

[110] Kim A Sharp and Barry Honig. Electrostatic interactions in macromolecules: theory and applications. *Annual review of biophysics and biophysical chemistry*, 19 (1):301–332, 1990.

[111] A Warshel, G Naray-Szabo, F Sussman, and JK Hwang. How do serine proteases really work? *Biochemistry*, 28(9):3629–3637, 1989.

[112] Kim A Sharp and Barry Honig. Calculating total electrostatic energies with the nonlinear poisson-boltzmann equation. *Journal of Physical Chemistry*, 94(19): 7684–7692, 1990.

[113] Stephen C Harvey. Treatment of electrostatic effects in macromolecular modeling. *Proteins: Structure, Function, and Bioinformatics*, 5(1):78–92, 1989.

[114] Michael K Gilson, Malcolm E Davis, Brock A Luty, and J Andrew McCammon. Computation of electrostatic forces on solvated molecules using the poisson-boltzmann equation. *The Journal of Physical Chemistry*, 97(14):3591–3600, 1993.

[115] Ulrich Essmann, Lalith Perera, Max L Berkowitz, Tom Darden, Hsing Lee, and Lee G Pedersen. A smooth particle mesh ewald method. *Journal of Chemical Physics*, 103(19):8577–8593, 1995.

[116] Alexander A Rashin and Barry Honig. Reevaluation of the born model of ion hydration. *The journal of physical chemistry*, 89(26):5588–5593, 1985.

[117] W Clark Still, Anna Tempczyk, Ronald C Hawley, and Thomas Hendrickson. Semianalytical treatment of solvation for molecular mechanics and dynamics. *Journal of the American Chemical Society*, 112(16):6127–6129, 1990.

[118] Wonpil Im, Michael S Lee, and Charles L Brooks. Generalized born model with a simple smoothing function. *Journal of computational chemistry*, 24(14): 1691–1702, 2003.

[119] Gregory D Hawkins, Christopher J Cramer, and Donald G Truhlar. Pairwise solute descreening of solute charges from a dielectric medium. *Chemical Physics Letters*, 246(1):122–129, 1995.

[120] Arieh Warshel and Arno Papazyan. Electrostatic effects in macromolecules: fundamental concepts and practical modeling. *Current opinion in structural biology*, 8(2):211–217, 1998.

[121] Claudia N Schutz and Arieh Warshel. What are the dielectric âconstantsâ of proteins and how to validate electrostatic models? *Proteins: Structure, Function, and Bioinformatics*, 44(4):400–417, 2001.

[122] Robert B Hermann. Theory of hydrophobic bonding. ii. correlation of hydrocarbon solubility in water with solvent cavity surface area. *The Journal of Physical Chemistry*, 76(19):2754–2759, 1972.

[123] Christopher J Cramer and Donald G Truhlar. Implicit solvation models: equilibria, structure, spectra, and dynamics. *Chemical Reviews*, 99(8):2161–2200, 1999.

[124] Doree Sitkoff, Kim A Sharp, and Barry Honig. Accurate calculation of hydration free energies using macroscopic solvent models. *The Journal of Physical Chemistry*, 98(7):1978–1988, 1994.

[125] Byungkook Lee and Frederic M Richards. The interpretation of protein structures: estimation of static accessibility. *Journal of molecular biology*, 55(3):379–IN4, 1971.

[126] Christopher J Fennell, Charlie Kehoe, and Ken A Dill. Oil/water transfer is partly driven by molecular shape, not just size. *Journal of the American Chemical Society*, 132(1):234–240, 2009.

[127] Kim A Sharp, Anthony Nicholls, Richard Friedman, and Barry Honig. Extracting hydrophobic free energies from experimental data: relationship to protein folding and theoretical models. *Biochemistry*, 30(40):9686–9697, 1991.

[128] Jed W Pitera and Wilfred F van Gunsteren. The importance of solute-solvent van der waals interactions with interior atoms of biopolymers. *Journal of the American Chemical Society*, 123(13):3163–3164, 2001.

[129] Sergei V Krivov, Stefanie Muff, Amedeo Caflisch, and Martin Karplus. One-dimensional barrier-preserving free-energy projections of a β-sheet miniprotein: New insights into the folding process. *The Journal of Physical Chemistry B*, 112(29):8701–8714, 2008.

[130] Jeffrey K Myers, C Nick Pace, and J Martin Scholtz. Denaturant m values and heat capacity changes: relation to changes in accessible surface areas of protein unfolding. *Protein Science*, 4(10):2138–2148, 1995.

[131] Frank Eisenhaber, Philip Lijnzaad, Patrick Argos, Chris Sander, and Michael Scharf. The double cubic lattice method: efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. *Journal of Computational Chemistry*, 16(3):273–284, 1995.

[132] Ariel A Chialvo and Pablo G Debenedetti. On the use of the verlet neighbor list in molecular dynamics. *Computer Physics Communications*, 60(2):215–224, 1990.

[133] Jörg Weiser, Armin A Weiser, Peter S Shenkin, and W Clark Still. Neighbor-list reduction: optimization for computation of molecular van der waals and solvent-accessible surface areas. *Journal of computational chemistry*, 19(9):1110, 1998.

[134] Jörg Weiser, Peter S Shenkin, and W Clark Still. Fast, approximate algorithm for detection of solvent-inaccessible atoms. *Journal of computational chemistry*, 20 (6):586–596, 1999.

[135] TP Straatsma, HJC Berendsen, and JPM Postma. Free energy of hydrophobic hydration: A molecular dynamics study of noble gases in water. *The Journal of chemical physics*, 85:6720, 1986.

[136] Robert W Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, 22:1420, 1954.

[137] G De Fabritiis, PV Coveney, and J Villà-Freixa. Energetics of k+ permeability through gramicidin a by forward-reverse steered molecular dynamics. *Proteins: Structure, Function, and Bioinformatics*, 73(1):185–194, 2008.

[138] AK Soper. Empirical potential monte carlo simulation of fluid structure. *Chemical Physics*, 202(2):295–306, 1996.

[139] Dirk Reith, Mathias Pütz, and Florian Müller-Plathe. Deriving effective mesoscale potentials from atomistic simulations. *Journal of computational chemistry*, 24(13):1624–1636, 2003.

[140] Alessandra Villa, Christine Peter, and Nico FA van der Vegt. Self-assembling dipeptides: conformational sampling in solvent-free coarse-grained simulation. *Physical Chemistry Chemical Physics*, 11(12):2077–2086, 2009.

[141] Andrzej J Rzepiela, Martti Louhivuori, Christine Peter, and Siewert J Marrink. Hybrid simulations: combining atomistic and coarse-grained force fields using virtual sites. *Physical Chemistry Chemical Physics*, 13(22):10437–10448, 2011.

[142] Furio Ercolessi and James B Adams. Interatomic potentials from first-principles calculations: the force-matching method. *EPL (Europhysics Letters)*, 26(8):583, 1994.

[143] Sergei Izvekov, Michele Parrinello, Christian J Burnham, and Gregory A Voth. Effective force fields for condensed phase systems from ab initio molecular dynamics simulation: A new method for force-matching. *The Journal of chemical physics*, 120:10896, 2004.

[144] Naomi Siew, Arne Elofsson, Leszek Rychlewski, and Daniel Fischer. Maxsub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, 16(9):776–785, 2000.

[145] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[146] Yue-Feng Shen, Bo Li, and Zhi-Ping Liu. Protein structure alignment based on internal coordinates. *Interdisciplinary Sciences: Computational Life Sciences*, 2(4): 308–319, 2010.

[147] David Pelta, Natalio Krasnogor, Carlos Bousono-Calzon, José L Verdegay, J Hirst, and Edmund Burke. A fuzzy sets based generalization of contact maps for the overlap of protein structures. *Fuzzy Sets and Systems*, 152(1):103–123, 2005.

[148] Samuel S Cho, Yaakov Levy, and Peter G Wolynes. P versus q: Structural reaction coordinates capture protein folding on smooth landscapes. *Proceedings of the National Academy of Sciences of the United States of America*, 103(3):586–591, 2006.

[149] Ming Li, Jonathan H Badger, Xin Chen, Sam Kwong, Paul Kearney, and Haoyong Zhang. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149–154, 2001.

[150] Chris S. Wallace and David L. Dowe. Minimum message length and kolmogorov complexity. *The Computer Journal*, 42(4):270–283, 1999.

[151] Natalio Krasnogor and David A Pelta. Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, 20(7): 1015–1021, 2004.

[152] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[153] Alexandros Altis, Phuong H Nguyen, Rainer Hegger, and Gerhard Stock. Dihedral angle principal component analysis of molecular dynamics simulations. *The Journal of chemical physics*, 126:244111, 2007.

[154] Lars Onsager. Electric moments of molecules in liquids. *Journal of the American Chemical Society*, 58(8):1486–1493, 1936.

[155] Frederick S Lee and Arieh Warshel. A local reaction field method for fast evaluation of long-range electrostatic interactions in molecular simulations. *The Journal of chemical physics*, 97(5):3100, 1992.

[156] Ilario G Tironi, René Sperb, Paul E Smith, and Wilfred F van Gunsteren. A generalized reaction field method for molecular dynamics simulations. *The Journal of chemical physics*, 102:5451, 1995.

[157] Markus Deserno and Christian Holm. How to mesh up ewald sums. i. a theoretical and numerical comparison of various particle mesh routines. *The Journal of chemical physics*, 109:7678, 1998.

[158] Alexey Onufriev, David A Case, and Donald Bashford. Effective born radii in the generalized born approximation: the importance of being perfect. *Journal of computational chemistry*, 23(14):1297–1304, 2002.

[159] Gregory D Hawkins, Christopher J Cramer, and Donald G Truhlar. Parametrized models of aqueous free energies of solvation based on pairwise descreening of solute atomic charges from a dielectric medium. *The Journal of Physical Chemistry*, 100(51):19824–19839, 1996.

[160] A_ Bondi. van der waals volumes and radii. *The Journal of Physical Chemistry*, 68(3):441–451, 1964.

[161] Jiang Zhu, Emil Alexov, and Barry Honig. Comparative study of generalized born models: Born radii and peptide folding. *The Journal of Physical Chemistry B*, 109(7):3008–3022, 2005.

[162] Shoshana J Wodak and Joël Janin. Analytical approximation to the accessible surface area of proteins. *Proceedings of the National Academy of Sciences*, 77(4): 1736–1740, 1980.

[163] Jörg Weiser, Peter S Shenkin, and W Clark Still. Approximate atomic surfaces from linear combinations of pairwise overlaps (lcpo). *Journal of Computational Chemistry*, 20(2):217–230, 1999.

[164] Timothy J. Richmond. Solvent accessible surface area and excluded volume in proteins: Analytical equations for overlapping spheres and implications for the hydrophobic effect. *Journal of Molecular Biology*, 178(1):63 – 89, 1984. ISSN 0022-2836. doi: 10.1016/0022-2836(84)90231-6. URL `http://www.sciencedirect.com/science/article/pii/0022283684902316`.

[165] M. L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16(5):548–558, Oct 1983. doi: 10.1107/S0021889883010985. URL `http://dx.doi.org/10.1107/S0021889883010985`.

[166] Robert Fraczkiewicz and Werner Braun. Exact and efficient analytical calculation of the accessible surface areas and their gradients for macromolecules. *J. Comput. Chem.*, 19(3):319–333, 1998.

[167] G. Perrot, B. Cheng, K. D. Gibson, J. Vila, K. A. Palmer, A. Nayeem, B. Maigret, and H. A. Scheraga. Mseed: A program for the rapid analytical determination of accessible surface areas and their derivatives. *J. Comput. Chem.*, 13(1):1–11, 1992.

[168] S. Sridharan, Anthony Nicholls, and Kim A. Sharp. A rapid method for calculating derivatives of solvent accessible surface areas of molecules. *J. Comput. Chem.*, 16(8):1038–1044, 1995.

[169] Shura Hayryan, Chin-Kun Hu, Jaroslav Skřivánek, Edik Hayryan, and Imrich Pokorný. A new analytical method for computing solvent-accessible surface area of macromolecules and its gradients. *J. Comput. Chem.*, 26(4):334–343, 2005.

[170] Manfredo Perdigao Do Carmo and Manfredo Perdigao Do Carmo. *Differential geometry of curves and surfaces*, volume 2. Prentice-Hall Englewood Cliffs, 1976.

[171] Karl Friedrich Gauss. General investigations of curved surfaces of 1827 and 1825. *Raven Press Books Ltd.*, 1965.

[172] Eric W Weisstein. Principal curvatures. *MathWorld–A Wolfram Web Resource*, 2013. URL http://mathworld.wolfram.com/PrincipalCurvatures.html.

[173] Todd Rowland. Compact surface. *MathWorld–A Wolfram Web Resource*, 2013. URL http://mathworld.wolfram.com/CompactSurface.html.

[174] Mary A Rohrdanz, Wenwei Zheng, and Cecilia Clementi. Discovering mountain passes via torchlight: Methods for the definition of reaction coordinates and pathways in complex macromolecular reactions. *Annual review of physical chemistry*, 64:295–316, 2013.

[175] Christoph Dellago, Peter Bolhuis, and Phillip L Geissler. Transition path sampling. *Advances in Chemical Physics*, 123:1–78, 2002.

[176] Ken A Dill, Klaus M Fiebig, and Hue Sun Chan. Cooperativity in protein-folding kinetics. *Proceedings of the National Academy of Sciences*, 90(5):1942–1946, 1993.

[177] S Banu Ozkan, G Albert Wu, John D Chodera, and Ken A Dill. Protein folding by zipping and assembly. *Proceedings of the National Academy of Sciences*, 104(29): 11987–11992, 2007.

[178] Andrew Blake, Pushmeet Kohli, and Carsten Rother. *Markov random fields for vision and image processing*. The MIT Press, 2011.

[179] D. Doria. Loopy belief propagation on mrfs in itk. *Insight Journal*, 03 2011.

[180] Peter G Bolhuis, David Chandler, Christoph Dellago, and Phillip L Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual review of physical chemistry*, 53(1):291–318, 2002.

[181] Gerhard Hummer. From transition paths to transition states and rate coefficients. *The Journal of chemical physics*, 120:516, 2004.

[182] Daniel L Ensign, Peter M Kasson, and Vijay S Pande. Heterogeneity even at the speed limit of folding: large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. *Journal of molecular biology*, 374(3):806–816, 2007.

[183] Todd K Moon. The expectation-maximization algorithm. *Signal processing magazine, IEEE*, 13(6):47–60, 1996.

[184] *A distance measure between GMMs based on the unscented transform and its application to speaker recognition*, INTERSPEECH, 2005.

[185] Katharine Miller. Bringing the fruits of computation to bear on human health: Its a tough job, but the NIH has to do it. *Biomed. Comput. Rev.*, 5(2):18–28, 2009.

[186] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–10, November 2002. doi: 10.1038/nature01254.

[187] Arieh Warshel and Michael Levitt. Theoretical studies of enzymic reactions: dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *J. Mol. Biol.*, 103(2):227–249, 1976.

[188] Paul Sherwood, Bernard R. Brooks, and Mark S. P. Sansom. Multiscale methods for macromolecular simulations. *Curr. Opin. Struct. Biol.*, 18(5):630–40, October 2008. doi: 10.1016/j.sbi.2008.07.003.

[189] Hao Hu and Weitao Yang. Free energies of chemical reactions in solution and in enzymes with ab initio quantum mechanics/molecular mechanics methods. *Annu. Rev. Phys. Chem.*, 59(1):573–601, 2008. ISSN 0066-426X. doi: 10.1146/annurev.physchem.59.032607.093618. URL http://dx.doi.org/10.1146/annurev.physchem.59.032607.093618.

[190] Z. Z. Fan, J. K. Hwang, and Arieh Warshel. Using simplified protein representation as a reference potential for all-atom calculations of folding free energy. *Theor. Chim. Acta*, 103(1):77–80, 1999.

[191] Gary S Ayton, Will G Noid, and Gregory A Voth. Multiscale modeling of biomolecular systems: in serial and in parallel. *Curr. Opin. Struct. Biol.*, 17(2): 192–8, April 2007. doi: 10.1016/j.sbi.2007.03.004.

[192] Benjamin M. Messer, Maite Roca, Zhen T. Chu, Spyridon Vicatos, Alexandra Vardi-Kilshtain, and Arieh Warshel. Multiscale simulations of protein landscapes: using coarse-grained models as reference potentials to full explicit models. *Proteins: Struct., Funct., Bioinf.*, 78(5):1212–27, April 2010. doi: 10.1002/prot.22640.

[193] Normand Mousseau and Philippe Derreumaux. Exploring energy landscapes of protein folding and aggregation. *Front. Biosci.*, 13:4495–516, January 2008.

[194] Ken A. Dill, S. Banu Ozkan, M. Scott Shell, and Thomas R. Weikl. The protein folding problem. *Annu. Rev. Biophys.*, 37(1):289–316, 2008. doi: 10.1146/annurev.biophys.37.092707.153558. URL http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.biophys.37.092707.153558. PMID: 18573083.

[195] César L. Ávila, Nils J. D. Drechsel, Raúl Alcántara, and Jordi Villà-Freixa. Multiscale molecular dynamics of protein aggregation. *Current Protein and Peptide Science*, 12(3):221–234, 2011.

[196] Michael A. Johnston, Ignacio Fdez. Galván, and Jordi Villà-Freixa. Framework-based design of a new all-purpose molecular simulation application: the adun simulator. *J. Comput. Chem.*, 26(15):1647–1659, Nov 2005. doi: 10.1002/jcc.20312. URL http://dx.doi.org/10.1002/jcc.20312.

[197] M. A. Johnston and J. Villà-Freixa. Enabling data sharing and collaboration in complex systems applications. *Lect. Notes Bioinform.*, 4360:124–140, 2007. URL http://cbbl.imim.es:8080/cbbl/Members/jvilla/cbbl-folder/PDF/johnston07.pdf.

[198] Johnathan Cooper, Frederic Cervenansky, Gianni de Fabritiis, John Fenner, Denis Friboulet, Toni Giorgino, Steven Manos, Yves Martelli, Jordi Villà-Freixa, Stefan Zasada, Sharon Lloyd, Keith McCormack, and Peter V. Coveney. The Virtual Physiological Human Toolkit. *Phil. Trans. R. Soc. A*, 368:3925–3936, 2010. URL http://www.ncbi.nlm.nih.gov/pubmed/20643685.

[199] IF Thorpe, J Zhou, and GA Voth. Peptide folding using multiscale coarse-grained models. *The Journal of Physical Chemistry B*, 112(41):13079–13090, 2008.

[200] Wendy D. Cornell, Piotr Cieplak, Christopher I. Bayly, Ian R. Gould, Kenneth M. Merz, David M. Ferguson, David C. Spellmeyer, Thomas Fox, James W. Caldwell, and Peter A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995. doi: 10.1021/ja00124a002. URL `http://pubs.acs.org/doi/abs/10.1021/ja00124a002`.

[201] M Scott Shell, Ryan Ritterson, and Ken A Dill. A test on peptide stability of amber force fields with implicit solvation. *The Journal of Physical Chemistry B*, 112(22):6878–6886, 2008.

[202] Tim Meyer, Marco D'Abramo, Manuel Rueda, Carles Ferrer-Costa, Alberto Pérez, Oliver Carrillo, Jordi Camps, Carles Fenollosa, Dmitry Repchevsky, Josep Lluis Gelpí, et al. Model (molecular dynamics extended library): a database of atomistic molecular dynamics trajectories. *Structure*, 18(11):1399–1409, 2010.

[203] Manuel Rueda, Carles Ferrer-Costa, Tim Meyer, Alberto Pérez, Jordi Camps, Josep Lluis Gelpí, Modesto Orozco, et al. A consensus view of protein dynamics. *Proceedings of the National Academy of Sciences*, 104(3):796–801, 2007.

[204] Tim Meyer, Carles Ferrer-Costa, Alberto Pérez, Manuel Rueda, Axel Bidon-Chanal, F Javier Luque, Charles A Laughton, and Modesto Orozco. Essential dynamics: a tool for efficient trajectory compression and management. *Journal of Chemical Theory and Computation*, 2(2):251–258, 2006.

[205] J Michael Word, Simon C Lovell, Jane S Richardson, David C Richardson, et al. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of molecular biology*, 285(4):1735–1747, 1999.

[206] Michael A Johnston, Ignacio Fdez Galván, and Jordi Villà-Freixa. Framework-based design of a new all-purpose molecular simulation application: The adun simulator. *Journal of computational chemistry*, 26(15):1647–1659, 2005.

[207] Giacomo MS De Mori, Cristian Micheletti, and Giorgio Colombo. All-atom folding simulations of the villin headpiece from stochastically selected coarse-grained structures. *The Journal of Physical Chemistry B*, 108(33):12267–12270, 2004.

[208] Giacomo De Mori, Giorgio Colombo, and Cristian Micheletti. Study of the villin headpiece folding dynamics by combining coarse-grained monte carlo evolution and all-atom molecular dynamics. *Proteins: Structure, Function, and Bioinformatics*, 58(2):459–471, 2005.

[209] Ariel Fernández, Min-yi Shen, Andrés Colubri, Tobin R Sosnick, R Stephen Berry, and Karl F Freed. Large-scale context in protein folding: villin headpiece. *Biochemistry*, 42(3):664–671, 2003.

[210] Min-yi Shen and Karl F Freed. All-atom fast protein folding simulations: The villin headpiece. *Proteins: Structure, Function, and Bioinformatics*, 49(4):439–445, 2002.

[211] Thomas Herges and Wolfgang Wenzel. Free-energy landscape of the villin headpiece in an all-atom force field. *Structure*, 13(4):661–668, 2005.

[212] Hongxing Lei, Chun Wu, Haiguang Liu, and Yong Duan. Folding free-energy landscape of villin headpiece subdomain from molecular dynamics simulations. *Proceedings of the National Academy of Sciences*, 104(12):4925–4930, 2007.

[213] Weiguo Liu, Bertil Schmidt, Gerrit Voss, and Wolfgang Müller-Wittig. Molecular dynamics simulations on commodity gpus with cuda. In *High Performance Computing–HiPC 2007*, pages 185–196. Springer, 2007.