DISS. ETH NO. 28165

Free Energy Methods: Drug-Like Molecules and Macrocycles

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Benjamin Joachim Ries

MSc. in Bioinformatics

born on 15.03.1991
citizen of Germany

accepted on the recommendation of

Prof. Dr. Sereina Riniker, examiner

Prof. Dr. Philippe H. Hünenberger, co-examiner
Prof. Dr. Niels Hansen, co-examiner

2022

pour Anne

Acknowledgements

This thesis would not have been possible without many people who have supported me over the last few years. I will try to list many of them here.

I deeply want to thank my supervisor Prof. Sereina Riniker. Thank you for giving me the opportunity to work on fascinating projects and develop my ideas. I admire your disciplined and efficient project management. Your patience and straightforward approach to science taught me a lot. I also want to thank you for listening when times were rough, and matters were not science-related. I would like to express my gratitude to Prof. Philippe H ”unenberger: Thank you for inspiring me with your endless creativity and authentic way of handling science. I would also like to thank my co-examiner, Prof. Niels Hansen, for the great lecture about free energy calculations and that he accepted to examine this thesis.

I want to thank the combined groups of computational chemistry (CCG) and Computer Simulation of Molecular Systems (CSMS) for the great atmosphere and the activities we could do together. It is a shame that the Covid pandemic pushed us to the home office and forced us to reduce the activities. Specifically, I would like to thank the following people (lexicographically sorted): Alzbeta Kubincova, for the great ice-skating events and the fantastic hackathon project we did together. Anick Rennevey, for the introduction into the computer system of IGC and for sharing the incredible Asian food. Anna Albertini, for the cool

shared teaching experience and for being the new organizer of the CCCCP. Candide Champion, for the discussions and the great hike to the grosser Mythen. Carmen Esposito, for the coffee, with discussions about PCAs and the great group opera visits. David Hahn, for the great discussions about free energy calculations and initial spark and work towards the Ensembler package. Dominik Sidler, for the fabulous group weekend at the Bettmeralp and teaching me the value of constructing theory. Emilia P. Barros, for the great discussions about EoffRebalancing and beautiful fairytale-like places in Switzerland. Felix Pultar for the discussions about PyGromosTools, and I hope you find it a convenient tool. Greg Landrum, for sharing his knowledge and experience in cheminformatics. M. M. Gregor Weiss, for sharing the office with me, his passion for coding, exploring the clustering space together, and our mountain bike tours. Gerhard König we had an exciting start that resulted in great collaboration, many beautiful coffee sessions, and discussions about high german. Jagna Witek, for giving me an excellent start to Zurich and introducing me to the theory of the Markov state modeling for macrocycles. Jessica Braun it was a pleasure meeting you, and I'm greatly cheered up to share the same liberal definition for APIs with you. Lennard Böselt, I enjoyed having discussions about a possible transition EDS with you, finding similarities in our families, and I am happy having convinced you of IDEs. Marc Lehner, for the introduction to the alpine world, seeing potential in the convenient PyGromosTools and sharing the swiss culture with me. Moritz Thürliman, for the great discussions about neuronal networks, politics, and reviving my passion for literature. Paul Katzberger, for the political status report of Austria and his fascination for Numba (which, of course, will lead to me learning more about it). Patrick Bleiziffer, for the help with the partial charge calcu-

lations for the 3D-PSAs. Sadra Kashef ol Gheta, for the great philosophical conversations. Marina Pereira Oliviera, for excellent soccer games and introducing her beautiful dog Lua to us. Salomé Rieder, thanks for the productive discussions, and I enjoyed the collaborative work together with you. Shuzhe Wang, for the good questions and discussions. Stephanie Linker, for the mood uplifting and motivating discussions, that were a great asset and always cleared up grey days. Thomas Stadelmann, for a great time with semi-peptidic macrocycles and the excellent sarcastic humor.

I want to thank Kay Shaller, Clemens Rhiner, Karl Normak, and Theo Smertnig, the students I supervised, for your work. You did contribute fundamentally to our research, and I'm proud to have been a part of your studies.

I want to especially thank our collaborators at the University de Sherbrooke Christian Comeau and Éric Marsault, who sadly passed away during the collaboration.

I also want to thank Prof. Chris Oostenbrink for the support with the Gromos Gitlab, and I hope that the GROMOS family will make it open source soon for the future of GROMOS.

Dear Claudia Hilty, thanks for your fast and always competent support. And the members of the group of Prof. Reiher for beautiful after-work activities.

I want to acknowledge Prof. van der Spoel, Nina Fischer, Philip Thiel, and Prof. Kohlbacher, who opened the door to Molecular dynamics simulations for me. I'm very thankful for their support and advice. Mohammad Mehdi Ghahremanpour, I want to thank you for your advice and support that encouraged me on my path.

I don't want to forget the societies where I could interact with many different people. Dear VAC, being your dragon counting

the coin or organizing hikes with you was a pleasure. I want to mention Danylo Matselyukh and Agathe Vanas especially. Dear YoungSCS and especially the Board of 2020, it was a wonderful experience to meet all of you and reboot YoungSCS together with you as president. In my first year with youngSCS, we started with five people at the general assembly. After two years, we achieved 20 active people with a general assembly of 80 people and many events organized by youngSCS, which I consider a great success of the Board. I especially want to mention Eva Vandale, Melanie Gut, Ahmed Elabd, Stephanie Linker, Patrick Fritz, Marie Dèsirèe Scheidt, Marie Perrin, and Lluc Farrera-Soler

Last but not least, I want to thank Dennis, Unn Beate & Edwin, Barbara, Enes, Viktoria, André and Alexander for their friendship, which I deeply value. I'm sincerely grateful for the support of my family and my wife.

"I wett das du nach de stärne griffsch
Gliich wie wiit entfärnt si si
Das du dänksch lachsch läbsch,
Wiu die zit goht schnäu verbi
Lueg das du nach de stärne griffsch
Gliich wie wiit entfärnt si si
..."

Manillio, Stärne, Jede Tag Superstar

Contents

Summary

Computational methods to calculate free-energy differences are of high interest in computer-aided drug discovery. Rigorous free-energy methods are based on molecular dynamics (MD) simulations. Chapter 1 provides a short overview of the fundamental concepts of MD. Chapters 2 - 4 describe developments in the area of free-energy calculation. In Chapter 2, the software package Ensembler for toy model simulations is introduced. Ensembler can be used to rapidly develop prototypes for method development or teach theoretical backgrounds of simulation techniques. An application example showcases how Ensembler can be used to execute and compare different free-energy methods with one-dimensional models. Chapter 3 provides a categorization of the existing approaches on how the end-states can be represented in relative free-energy calculations. Next, an algorithm is introduced for selecting (locally) optimal distance restraints to link molecules in a linked dual topology approach. The introduced algorithm can handle scaffold hopping-like transformations and is extended for multi-state methods. The performance of the approach is demonstrated in relative hydration free-energy calculations. In Chapter 4, the refined automated RE-EDS pipeline is presented, containing multiple methodological changes and tricks to ensure sufficient sampling of all end-states. The improved pipeline is applied to scaffold hopping-like transformations to showcase the capabilities of the methodology.

In Chapter 5, the Python API PyGromosTools is introduced.

The API provides functionality to modify GROMOS files, to set up and perform MD simulations, and to analyze the resulting data. Considerations regarding the code style and the code structure are discussed. Finally, usage examples are provided to show how simulations can be set up and executed with PyGromosTools. Chapter 6 explores the conformational behavior of pairs of semipeptidic macrocycles with a single stereo center change and its relationship with the observed passive membrane permeability. A particular interest is to rationalize a "permeability cliff" related to the stereocenter. We construct a hypothesis based on extensive MD simulations supported by NMR studies.

Finally, we will conclude the thesis with an outlook in Chapter 7 for the different topics covered in this work.

Zusammenfassung

Die computergestützte Berechnung von Freie Energiedifferenzen ist ein grundlegender Bestandteil der *in silico* Wirkstoffentwicklung. Rigorose Methoden zur Freie Energieberechnung basieren auf Molekulardynamik-Computersimulationen (MD-Simulationen). Kapitel 1 beinhaltet eine Zusammenfassung der grundlegenden Konzepte von MD-Simulationen. Die darauf folgenden Kapitel 2 - 4 beschreiben Entwicklungen im Bereich computergestützter Freie Energieberechnung. In Kapitel 2 wird das Softwarepaket Ensembler vorgestellt. Dieses kann verwendet werden, um schnell einfache Modelle für die Methodenentwicklung oder für Lehrzwecke zu generieren. In einem eindimensionalen Modellansatz werden verschiedene computergestützte Freie Energieberechnungsmethoden verwendet und miteinander verglichen. Kapitel 3 beinhaltet eine Einordnung für Systemrepräsentationen in relativen Freie Energieberechnungsmethoden. Zudem wird ein Algorithmus vorgestellt, der mehrere Moleküle mittels (lokal) optimaler Distanzdefinitionen für verknüpfte duale Topologien verbindet. Der entwickelte Algorithmus kann mit komplexen Ligandtransformationen, wie sie bei “Scaffold hopping” vorkommen, umgehen. Die Mächtigkeit des Algorithmus’ wird am Beispiel der Berechnung von Hydratisierungsenergie-Differenzen aufgezeigt. In Kapitel 4 werden Fortschritte in der RE-EDS Methodik präsentiert, welche auf Modifizierungen im automatischen Optimierungsprozess beruhen. Diese Modifizierungen stellen in erster Linie eine ausreichende Repräsentation jedes Endzustandes in der Simulation sicher. Die

verbesserte Pipeline wird auf ein komplexes Liganden-Protein System angewandt, um das Potential der Methode aufzuzeigen.

In Kapitel 5 wird die Python-Programmierschnittstelle (API) PyGromosTools vorgestellt. Die API ermöglicht den Zugriff auf GROMOS-Dateien, die Generierung von GROMOS-Systemen, die Durchführung von Simulationen sowie die Analyse der generierten Daten. Die Funktionalität der API wird anhand mehrerer Anwendungsbeispiele verdeutlicht. Kapitel 6 untersucht das konformationelle Verhalten von Paaren von semipeptidischen Makrozyklen, die sich in einem chiralen Zentrum unterschreiben, und verknüpft dieses mit der gemessenen passiven Membranpermeabilität. Die Änderung des chiralen Zentrums führte in einem Makrozyklenpaar zu einer starken Veränderung der Membrangängigkeit. Die Resultate der Computersimulationen werden mittels NMR-Messungen experimentell validiert. Das letzte Kapitel rekapituliert die Dissertation und gibt einen Ausblick auf mögliche Weiterentwicklung der Dissertationsthemen.

Publications

Articles in peer-reviewed journals:

1. B. Ries, S. M. Linker, D. F. Hahn, G. König, S. Riniker, Ensembler: A Simple Package for Fast Prototyping and Teaching Molecular Simulations, J. Chem. Inf. Model., 61 (2021) 560–564.

2. B. Ries[†], S. Rieder[†], C. Rhiner, Philippe H. Hünenberger, S. Riniker, *A Graph-Based Approach to the Restraint Problem in Dual Topology Approaches with RestraintMaker*, J. Comput.-Aided Mol. Des., submitted (2021).

3. B. Ries, K. Normak, R. G. Weiß, S. Rieder, C. Candide, G. König, S. Riniker, *Relative Free-Energy Calculations for Scaffold Hopping-Type Transformations with an Automated RE-EDS Sampling Procedure*, J. Comput.-Aided Mol. Des., in press (2021).

4. C. Comeau[†], B. Ries[†], T. Stadelmann, J. Tremblay, S. Poulet, U. Fröhlich, J. Côté, P. Boudreault, R. M. Derbali, P. Sarret, M. Grandbois, G. Leclair, S. Riniker, É. Marsault, *Modulation of the Passive Permeability of Semipeptidic Macrocycles: N- and C-Methylations Fine-Tune Conformation and Properties*, J. Med. Chem., 64 (2021) 5365–5383.

[†] These authors contributed equallly.

Related publications:

1. J. Witek, S. Wang, B. Schroeder, R. Lingwood, A. Dounas, H. Roth, M. Fouché, M. Blatter, O. Lemke, B. Keller, and S. Riniker, *Rationalization of the Membrane Permeability Differences in a Series of Analogue Cyclic Decapeptides*, J. Chem. Inf. Model., 59 (2019) 294–308 .

2. G. König, N. Glaser, B. Schroeder, A. Kubincová, P. H. Hünenberger, S. Riniker, *An Alternative to Conventional λ-Intermediate States in Alchemical Free Energy Calculations: λ-Enveloping Distribution Sampling*, J. Chem. Inf. Model., 60 (2020) 5407–5423.

3. R. G. Weiß, B. Ries, S. Wang, S. Riniker, *Volume-Scaled Common Nearest Neighbor Clustering Algorithm with Free-Energy Hierarchy*, J. Chem. Phys., 154 (2021) 084106.

4. S. M. Linker, S. Wang, B. Ries, T. Stadelmann, S. Riniker, *Passing the Barrier – How Computer Simulations Can Help to Understand and Improve the Passive Membrane Permeability of Cyclic Peptides*, CHIMIA, 75 (2021) 518–521.

5. G. König, B. Ries, P. H. Hünenberger, S. Riniker, *Efficient Alchemical Intermediate States in Free Energy Calculations Using λ-EDS*, J. Chem. Inf. Model., 17 (2021) 5805–5815.

6. E. P. Barros, B. Ries, L. Böselt, C. Champion, P. H. Hünenberger, S. Riniker, *Recent Developments in Multiscale Free Energy Simulations*, Curr. Opin. Struct. Biol. (2022) 72, 55–62.

7. S. M. Linker, C. Schellhaas, B. Ries, and S. Riniker, *Membrane-Like Interfaces Facilitate Conformational Closure for Pas-*

sive Permeability of Cyclic Peptides, RSC Adv., submitted (2021).

1 Introduction

„Dass ich erkenne, was die Welt
Im Innersten zusammenhält, “
“So that I may perceive whatever holds
The world together in its inmost folds.”

Dr. H. Faust, Faust
Johann W. von Goethe[1]

Over the past century, elucidating the three-dimensional (3D) structures of macromolecules has fundamentally enriched our understanding of biochemistry,[2,3] creating the field of structural biology. Achievements like the structure of deoxyribonucleic acid (DNA)[4] (with the unpublished work of Rosalind Franklin[5]) led to the central dogma of molecular biology,[6] while the first structure of an α-helix[7] shaped our understanding of the structural building blocks in protein structures. Over time, various methods have been developed to elucidate 3D structures, with the most prominent methods being X-ray crystallography,[3,8] nuclear magnetic resonance (NMR),[2,3,9,10] and the more recent single-particle cryo-electron microscopy.[11–14] The first complete 3D structure of a protein was of myoglobin,[15] published in 1958, opening the door for molecular modeling with biomolecules and rational drug design. Early perspectives on proteins considered their structure to be very rigid.[16] This belief was transformed to a much more

dynamic understanding of protein structures, influenced by theoretical methods such as molecular dynamics (MD) simulations[16,17] and by developments in experimental methodology, e.g., the B-factor analysis for X-ray crystallography[18] and NMR.[19–21] The first MD simulation contributing to the transition was performed by McCammon and co-workers on the pancreatic trypsin inhibitor (BPTI) for 9.2 *ps* under vacuum conditions.[22]

To date, a considerable amount of literature has been published using MD simulations to model the conformational behavior of biological systems containing proteins, nucleic acids, carbohydrates, and lipid membranes.[16,23–25] In these studies, simulation techniques are used to support structure determination, or to provide insights on thermodynamic or kinetic quantities.[16,26] Often, a combination of theoretical and experimental methods is used to exploit the complementary nature of the techniques and their synergies.[27] An example for such a combined approach is given in Chapter 6, where a rational for the difference in membrane permeability due to a stereocenter change in macrocyclic compounds is obtained based on MD simulation results, validated by additional experimental NMR.

With the first concept for binding free-energy calculations using MD simulations by Tembre,[28] an important step toward rational drug design was made.[29] However, this scheme for absolute free-energy calculations proved to be computationally very expensive and therefore was not feasible for a long time. Only relatively recently, advances in computer hardware and methodology have started to overcome these barriers.[30,31] Another important step toward rational drug design was made by Jorgensen and co-workers with the first relative free-energy calculation, opening the field to a more efficient methodology.[30,32,33] Today, free-energy calculations are (becoming) a standard tool in the drug discovery process and

support the design of modern therapeutics.[30,34–37]

Ligand binding free energies are not the only properties of interest for drug design. Another aspect that gained coverage recently in computational literature is the passive membrane permeability of drug molecules, as discussed in Chapter 6.[38–47] As simulations can nowadays routinely reach timescales in the order of μs, conformational dynamics can be determined to provide insight into the kinetic behavior of molecules.[38,39] In this context, a particular interest lies in compounds that are beyond the Lipinksi[48] rule-of-5 (bRO5).[38–41,46] The benefit of such bRO5 molecules lies in their complexity, allowing to target for example protein-protein interactions by mimicking substructure parts.[49–53] These more complex molecules such as the natural product cyclosporin A explore a larger conformational space. The good membrane permeability of this cyclic peptide is hypothesized to be due to its chameleonic nature, allowing the molecule to adopt closed (apolar) and open (polar) conformations depending on the polarity of the environment.[38,39]

A short introduction to simulation techniques and free-energy calculations is provided in the next section.

1.1 MOLECULAR DYNAMICS SIMULATIONS

MD simulations are powerful tools that enable the study of biomolecular systems under certain conditions. Four aspects of simulations will be shortly discussed: the model, force fields and interaction functions, integration schemes, and simulation conditions.

1.1.1 Model

In computational chemistry, simulations provide information about molecules and their properties. Three different levels of resolution are usually distinguished.[54] The highest resolution is based on quantum mechanics (QM) with an explicit description of the electrons.[55] Simulations on this level can for example provide insights into enzymatic reactions[56–59] or photon-induced electron excitation.[60–62]

The next lower level is molecular mechanics (MM), which is based on classical mechanics with atoms as single particles. Such atomistic simulations enable insights into the conformational behavior of molecules,[22,38,41,63] reaching much longer simulation times compared to QM calculations. Note that the large difference in computational cost led to the development of hybrid QM/MM methods, which are frequently used in modeling enzymatic reactions.[55] The third, most approximate, level is called coarse-grained (CG), where multiple atoms or even multiple molecules are described by CG beads.[64] Simulations on this level can access even longer timescales, enabling the study of polymer formation[65,66] or crowding effects in cells.[67,68]

In practical terms, the choice of a model defines the resolution of the coordinate and topology space of a system, which in turn specifies the phenomena that can be studied. In this thesis, the theory level of choice is MM with the united atoms approach[69] for aliphatic CH_X groups.

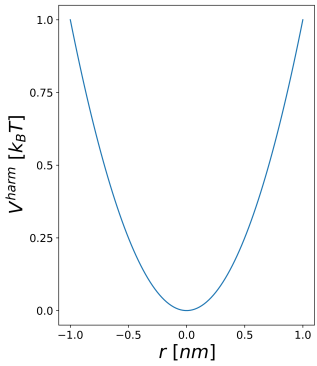
1.1.2 Force Fields and Interaction Functions

Empirical research has led to rules and concepts for atoms and their interactions with each other, for example covalent bond

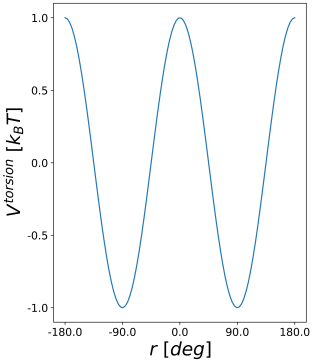
lengths and electrostatic interactions.[70–72] Interaction functions are used in modeling to represent these empirical rules and concepts. The required function parameters can be retrieved directly from the system coordinates or are provided via a predefined topology.[73–75] Topological parameters are often inferred from higher-level theoretical approaches such as QM, and/or fitted to reproduce experimental properties.[75–78]

The interaction functions can be split into two classes (Figure 1.1): (i) the intramolecular bonded terms, and (ii) the nonbonded terms (intramolecular and intermolecular).[78,79] The bonded terms consist of covalent bond-stretching, bond-angle bending, and dihedral-angle rotation. Covalent bonds of two atoms are an essential part of chemistry to build up molecules.[71] The length of a bond and its stretching impact was famously described by the Morse potential-energy function.[70,80] In simulations, this bond stretching term is, however, usually approximated with the simpler harmonic oscillator potential-energy function or alternative forms (Figure 1.1A).[81] Bond angles are calculated between three bonded atoms.[82,83] Again, a harmonic oscillator potential-energy function is typically used to model the bond-angle bending (Figure 1.1B). The torsion (or dihedral) angle describes the rotation around a covalent bond.[84] It is calculated from a set of four bonded atoms, and is an important quantity in chemistry (e.g., cis/trans isomerization[85] or ring conformations[86]). Dihedral-angle rotation is typically modeled with trigonometric functions (Figure 1.1C). Additionally, so-called improper dihedrals are applied to model rotation barriers, for example using harmonic oscillators (Figure 1.1A).[84]

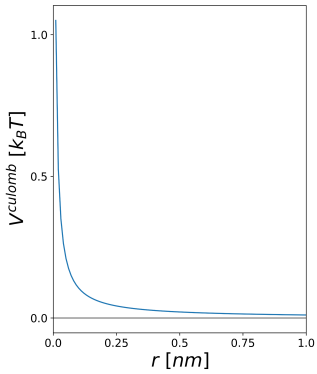
Nonbonded terms describe atom–atom interactions as a function of their spatial distance. Here, two fundamental terms are commonly distinguished. The electrostatic term describes the



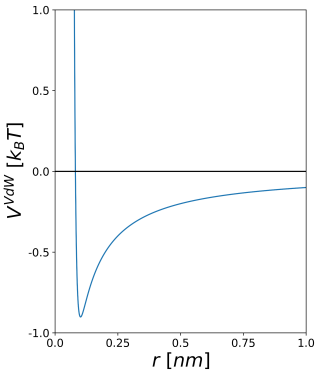
(A) Bond-stretching, bond-angle bending, improper dihedral



(B) Torsion (or dihedral) angle



(C) Electrostatic interaction



(D) van der Waals interaction

FIGURE 1.1: Potential-energy functions to represent atom–atom interactions. (A): Covalent bond-stretching, bond-angle bending, and improper dihedral. (B): Torsion (or dihedral) angle rotation, modeled by trigonometric potential-energy functions. (C): Electrostatic interaction, modeled by a Coulomb potential-energy function. (D): Van der Waals interactions, modeled by a Lennard-Jones (LJ) potential-energy function.

interaction of polar atoms with each other, and is modeled by a

Coulomb potential-energy function (Figure 1.1C).[72,87] In most MD simulations, a spatial cutoff is introduced to limit the directly calculated nonbonded terms for computational efficiency reasons. Cutoff inaccuracies are compensated by expanding the interaction function with additional terms that describe the contribution from the long-range interactions, like the reaction-field method (RF)[88] or the particle mesh Ewald (PME)[89] method. Van der Waals or dispersive interactions describe the electron fluctuations in atoms and their resulting interactions.[90,91] The van der Waals forces[91] are typically modeled with a Lennard-Jones (LJ)[92] 6-12 potential-energy function (Figure 1.1D). From a computational perspective, calculating the nonbonded terms is usually the most expensive operation. The complexity derives from the required distance calculation and the number of pairwise interactions in the system. To address this issue, much effort has been spent to parallelize the calculation of nonbonded interactions to significantly speed up the simulations.[93–96]

The different interaction terms are combined into a force-field function. The force field describes the total potential energy $V(\mathbf{r}^N)$ of the system with N particles and coordinates $\mathbf{r}^N$,[97]

$$\begin{aligned} V(\mathbf{r}^N) &= V^{\text{bonded}}(\mathbf{r}^N) + V^{\text{nonbonded}}(\mathbf{r}^N) \\ &= V^{\text{bond}}(\mathbf{r}^N) + V^{\text{angle}}(\mathbf{r}^N) \\ &\quad + V^{\text{torsion}}(\mathbf{r}^N) + V^{\text{improper}}(\mathbf{r}^N) \\ &\quad + V^{\text{electrostatics}}(\mathbf{r}^N) + V^{\text{vdW}}(\mathbf{r}^N). \end{aligned} \tag{1.1}$$

The total coordinate space gives rise to the potential-energy surface (PES) defined by a force field (Figure 1.2). During a simulation, minima and barriers of the PES are explored, providing insights into the conformational behavior of the system.

Many force-field functions and parameter sets were developed

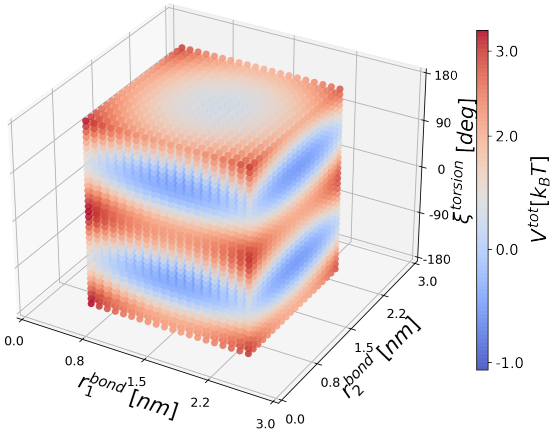


FIGURE 1.2: This PES was constructed with a grid from a force-field containing two bond stretch interactions and torsion angle interaction terms of an abstract system.

in the past, examples are AMBER,[74,98–100] GROMOS,[69,75,101–104] CHARMM,[105–107] OPLS[33,108] and OpenFF.[109] Additionally, automatic molecule parametrizing tools were developed for small organic molecules such as the automated topology builder (ATB),[103,104] the generalized Amber force field (GAFF)[110] or the CHARMM General force field (CGenFF).[111]

In practice, the bond-stretching terms are often replaced by Lagrange multiplier-based constraint algorithms such as the SHAKE,[112,113] SETTLE,[114] or LINCS[115] to omit the bond vibrations and enable larger time steps (up to 2 fs).[112]

1.1.3 Integration Schemes

Different multi-step integration methods can be used to integrate a force-field function, depending on the purpose. The first category consists of optimization algorithms such as the steepest descent,[116] or conjugated gradient.[117] These algorithms are used to find local minima in the PES, typically to relax the coordinates before starting a MD simulation.[118] The second category is represented by stochastic approaches like the Monte-Carlo approach or the Metropolis-Hastings integrator,[119] which depends on the Metropolis-Monte Carlo criterion.[120] These approaches can be used to sample the PES stochastically. The third and most popular category of integrations schemes is based on Newton's equation of motion.[121,122] Examples are second-order algorithms like the Verlet and leap-frog algorithms,[123] which perform well and provide sufficient accuracy.[26,124] The leap-frog algorithm employs the following two basic equations to predict the coordinates of

atom i at time $t + \Delta t$,

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t + \frac{1}{2}\Delta t)\Delta t \\ \mathbf{v}_i(t + \frac{1}{2}\Delta t) &= \mathbf{v}_i(t - \frac{1}{2}\Delta t) + \mathbf{a}_i(t)\Delta t. \end{aligned} \tag{1.2}$$

The acceleration $\mathbf{a}_i(t)$ is calculated from the force acting on atom i, $\mathbf{a}_i(t) = \mathbf{F}_i(\mathbf{r}(t))/m_i$, which in turn is derived from the potential energy, $\mathbf{F}_i(\mathbf{r}(t)) = \partial V(\mathbf{r}^N(t))/\partial \mathbf{r}_i(t)$.[26]

1.1.4 Simulation Conditions

The last aspect focuses on the tools to set specific boundary conditions in MD simulation. Boundary conditions are essential to retrieve correct thermodynamic properties from the simulations at certain physical conditions. Here, we shortly describe two of the most used ensembles in the context of Newtonian integrators.

In the canonical (NVT) ensemble, the number of particles (N), the system box volume (V), and temperature (T) are constant. Keeping N and V constant in a simulation is in most cases trivial. In contrast, thermostatting methods have to be introduced to keep T constant in a simulation. Various algorithms have been developed for this purpose. The first thermostat was introduced by Berendsen[125] with the weak-coupling approach. Because of the relation between T and the velocity of the particles in a system, the weak-coupling thermostat scales the velocities by a factor $\lambda(t; \tau_T, \Delta t, T_0)$ in order to maintain a constant temperature,

$$\lambda(t; \tau_T, \Delta t, T_0) = \sqrt{1 + \frac{\Delta t}{\tau_T}\left(\frac{T_0}{T(t)} - 1\right)}, \tag{1.3}$$

where τ_T is the coupling time, and T_0 the reference temperature.[125]

More sophisticated approaches are the Nosé-Hoover[126–128] and Nosé-Hoover chain[129] thermostats. Note that stochastic dynamics leads automatically to an NVT ensemble because of the temperature given in the Metropolis-Monte Carlo criterion.[119] The NVT ensemble can be used to calculate Helmholtz free energies.[130]

The second important ensemble is the isobaric-isothermal (NPT) ensemble with constant pressure (P). The pressure can be kept constant using barostat algorithms such as the weak-coupling barostat[125] that functions similar to the thermostat, the Parinello-Rahman barostat,[131] or the Nosé-Hoover barostat.[132] From an NPT ensemble, Gibbs free energies can be calculated.[133] This ensemble is often closest to experimental setups with a defined pressure and temperature.

1.2 FREE-ENERGY DIFFERENCES

Free energy is an fundamental quantity that describes the energy of states. In chemistry, the difference in free energy characterizes for example the spontaneity of reactions, the formation of polymers, or protein-ligand binding.[36,134–137]

From thermodynamics, the Helmholtz free energy[130] for a canonical ensemble can be obtained as,

$$A = -\frac{1}{\beta} \ln \Big(Q(N, V, T) \Big), \tag{1.4}$$

where Q(NVT) is the partition function of the system.[87]

The canonical ensemble is Boltzmann distributed[138] and thus,

the partition function can be formulated as,

$$Q = \sum_{i}^{N} e^{-\beta E_i}, \tag{1.5}$$

for an ensemble with discrete energy levels.[87]

A common principle in many free energy-based methods is to calculate the free-energy difference between two end-states.[139–141] Examples for such end-states are the bound and unbound states of a protein-ligand complex, or the assembled and disassembled states of polymers.[36,134–137] The free-energy difference between two end-states A and B is therefore defined as,[87]

$$\begin{aligned} \Delta A_{BA} &= -\frac{1}{\beta}\Big(\ln\big(Q_B(N,V,T)\big) - \ln(Q_A(N,V,T))\Big) \\ &= -\frac{1}{\beta}\ln\left(\frac{Q_B(N,V,T)}{Q_A(N,V,T)}\right). \end{aligned} \tag{1.6}$$

The ensembles generated by MD simulations approximate the partition functions of the end-states by sampling the Hamiltonian of the system $H(\mathbf{r},\mathbf{p}) = V(\mathbf{r}) + K(\mathbf{p})$ over the phase space as,[142]

$$Q = \frac{1}{h^{3N}N!}\int\int e^{-\beta H(\mathbf{p},\mathbf{r})} d\mathbf{p}d\mathbf{r}, \tag{1.7}$$

for indistinguishable particles, where h is the Planck constant and N the number of particles. From this, the Zwanzig equation[141] can be derived by employing the Boltzmann factors ($p = -\frac{e^{-\beta V_x}}{\sum^N e^{-\beta V_x}}$) of the two end-states A and B sampled as the ensemble average over state A,[141]

$$\Delta A_{BA} = -\frac{1}{\beta}\ln\left\langle \frac{e^{-\beta V_B}}{e^{-\beta V_A}} \right\rangle_A = -\frac{1}{\beta}\ln\left\langle e^{-\beta V_B - V_A} \right\rangle_A. \tag{1.8}$$

In Chapters 2-4, different free-energy methods will be discussed in detail, further developed, and applied to protein-ligand systems.

2

Ensembler: A Simple Package for Fast Prototyping and Teaching Molecular Simulations *

"It all works because Avogadro's number is closer to infinity than to 10."

R. Baierlein
Gromacs quote collection [143]

Ensembler is a Python package that allows for fast and easy access to the simulation of one and two-dimensional model systems. It enables method development using small test systems and to deepen the understanding of a broad spectrum of molecular dynamics (MD) methods, starting from basic techniques to enhanced sampling and free-energy calculations. The ease of installing and using the package increases shareability, comparability, and reproducibility of scientific code developments. Here, we

* This Chapter is reproduced in part from Benjamin Ries, Stephanie M. Linker, David F. Hahn, Gerhard König and Sereina Riniker, J. Chem. Inf. Model. 61 (2021) 560–564 , with permission from the American Chemical Society.

provide a description of the implementation and usage of the package as well as an application example for free-energy calculation. The code of Ensembler is freely available on GitHub `https://github.com/rinikerlab/Ensembler` and can be directly explored with Binder[144] or Colab.[145]

2.1 INTRODUCTION

Newly developed advanced simulation methods are routinely tested on simple one- and two-dimensional model systems. They provide valuable insights into the theory, conceptual advantages and limitations (for examples see e.g. Refs. 146–153). While the results of new methods are published, the implementation details may not always be available or difficult to use with different computer infrastructure. As a result, sharing, reproducing, understanding, and comparing simulation methodologies is often cumbersome.[154] To address this issue, we have developed the Ensembler package, an easy-to-use, yet powerful platform that enables fast prototyping of new methods and comparison against existing techniques using one or two-dimensional test systems.

Ensembler is designed following the recommendations of Stodden *et al.*[155] for the enhanced reproducibility of computational methods, which includes making code publicly accessible, providing documentation, and using open licensing.[155] Furthermore, Ensembler uses state-of-the-art software engineering tools (i.e. git,[156] MolSSI cookie-cutter,[157] and Binder[144]/Colab[145]) to fulfill these recommendations and enable features like continuous integration and the transparent versioning of the code.

2.1.1 Method Development

The lean Python3 code[158] of Ensembler allows for easy prototyping of new methods and comparison against a wide range of already implemented techniques. In contrast, the C/C++[159] code of traditional high-performance molecular dynamics (MD) packages (e.g. Refs. 93,160–163) is more efficient but also much more complex. The methods currently available in Ensembler are:

- *Model systems*: Harmonic oscillators as well as dihedral-angle, double-well, and Lennard-Jones potential-energy functions[92]

- *Sampling algorithms*: Conjugated gradient[117] for energy minimization, Metropolis Monte Carlo (MC),[119] leap-frog integration[164] for MD, and Langevin integration[165] for stochastic dynamics (SD)

- *Enhanced sampling techniques*: Umbrella sampling,[166] simulated tempering/temperature replica-exchange simulations,[167] local elevation/metadynamics[146,147]

- *Free-energy methods*: Free-energy perturbation (FEP),[141] Bennett's acceptance ratio (BAR),[168] thermodynamic integration (TI),[140] enveloping distribution sampling (EDS),[136,148,169] λ-EDS,[150] replica-exchange EDS (RE-EDS),[170] and conveyor-belt TI[171]

2.1.2 Teaching

Simple model systems can also be used for teaching MD concepts to students, as they allow to intuitively understand fundamental concepts.[172] Ensembler is well suited for didactic purposes because it is not only easy to use, but supports also a range of visualizations, i.e. interactive widgets, animations, and plots, which can be embedded in Jupyter notebooks.[173] Example Jupyter notebooks[173] are provided in the Ensembler GitHub repository.

2.2 IMPLEMENTATION

Ensembler is implemented in Python3[158] and available on GitHub[174] (*rinikerlab/Ensembler*). The repository is based on the template of the MolSSI cookie-cutter[157] and comprises a code folder, an example folder for tutorials, example models contained in the provided Jupyter notebooks,[173] an automatic pytest suite,[175] and the automatically generated sphinx[176] documentation. The code is continuously integrated via GitHub Actions,[177] providing information about code quality, test correctness, test coverage, and generation of an up-to-date documentation. Ensembler uses only open-source packages like the SciPy library[178–182] and Jupyter notebooks.[173] In the following, a user and a developer perspective are provided for the code structure.

2.2.1 User level

A simulation model in Ensembler consists of a potential class, a sampler class, and a system class wrapping the potential and the sampler (Figure 2.1), and provides control over the simulation approach. Additionally, multiple condition classes can be added that directly influence the simulation (e.g. periodic boundary condition[23,183] or thermostating[186]). After the construction of the system, the simulation can be started directly with the *simulate* function. The resulting trajectory is in the form of a Pandas data frame.[181] The trajectory is thus easily compatible with other packages like NumPy[179] or scikit-learn[187] and can be stored in different formats, e.g. as .csv or .hf5 file. The system itself can be stored directly via the save function using serialization of the object with the Python package pickle. In most cases, only a few

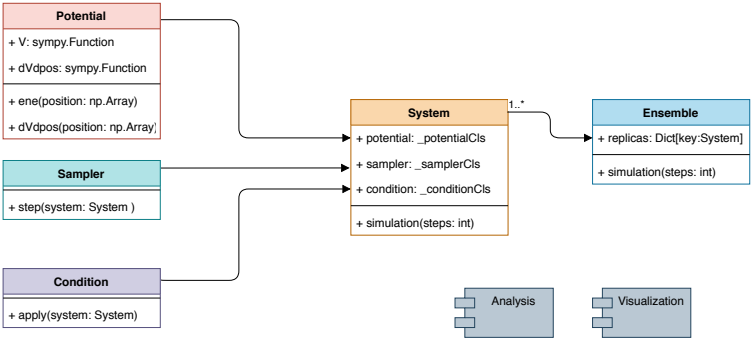


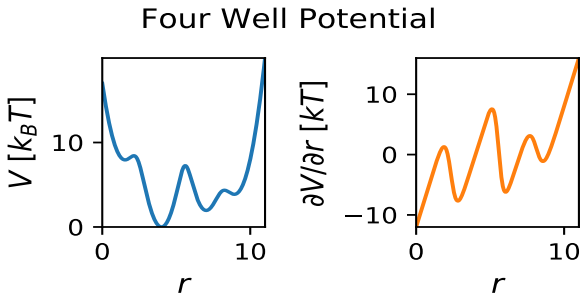
FIGURE 2.1: Unified modeling language (UML) class diagram of the five Ensembler base classes. The *potential class* defines the potential-energy functions to be explored and generates the required derivatives automatically (Figure 2.2). The implementation of a *potential class* (red) is based on the symbolic mathematical language of SymPy.[180] The *sampler classes* (cyan) are used for the sampling of potential-energy functions. *Condition classes* (purple) can have different functions, e.g. application of periodic boundary conditions,[23,183] thermostats, or restraints. The *system classes* (orange) serve as the scaffold for the potential, sampler, and condition classes. In this structure, all components, parameters, and the results of a simulation are stored. The *Ensemble class* (blue) can be used to perform advanced simulation techniques, e.g. using multiple walkers that explore the energy landscapes of the same or different systems as in replica-exchange approaches.[167,184,185] The *analysis package* includes free-energy methods such as the Zwanzig equation[141] or BAR.[168] A set of visualization functions is provided in the *visualization package* to enable an intuitive way of inspecting simulations or exploring potentials-energy functions. This includes plots, animations, and interactive widgets.

additional lines are needed to go from simple simulation technique to more advanced one, as shown below.

2.2.2 DEVELOPER LEVEL

The code of Ensembler is built on five interface-like base classes that allow extensive use of the inheritance concept and polymor-

phism[159] throughout the package. These fundamental classes are *potential*, *sampler*, *condition*, *system*, and *ensemble* (Figure 2.1), which can be grouped into three layers. *Potential*, *sampler*, and *condition classes* form the primary layer, providing different techniques to be used as components in a simulation. *Potential classes* provide the potential-energy functions in a symbolic form using SymPy,[180] enabling automatic on-the-fly derivation and simplification of the potential-energy function. *Sampler classes* are used to explore the potential-energy function (e.g. conjugate gradient,[117] Metropolis MC,[119] or leap-frog[164] integration). A new method can easily be implemented by inheriting from the *sampler class* and overwriting a single function called *step*. Finally, *condition classes* provide additional functionalities such as thermostatting[186] and periodic boundary conditions[23,183]). New techniques can be implemented by inheriting the base *condition class* and overwriting the function *apply*. In the second layer, the first-layer components are wrapped into one *system class* that executes the simulation(s) and manages the input and output. An optional higher-order layer is available in form of the *ensemble class*, which allows the user to perform simulations with replica exchange.[167,170,184,185] If additional parameters are needed in a newly designed class, the constructor of the new child class can be adapted but must call the parent constructor.



(A)

```
#Four Well Potential
##Imports
import numpy as np
from ensembler.potentials.OneD import fourWellPotential
from ensembler.visualisation.plotPotentials import plot_potential

#build potential
V = fourWellPotential(Vmax=4, a=1.5, b=4.0, c=7.0, d=9.0,
                      ah=2., bh=0., ch=0.5, dh=1.)

#visualize
positions = np.linspace(start=0, stop=11, num=1000) #phase space to be visualized
fig, outpath = plot_potential(potential=V, positions=positions,
                              title="Four Well Potential")
```

(B)

FIGURE 2.2: A four-well potential-energy surface visualized by the standard visualization function of Ensembler. (**A**) Potential-energy function (blue) and the automatically generated spatial gradient (orange) over a given coordinate range. (**B**) Source code to define the potential-energy function and the coordinate range to be visualized. These parameters are passed to the built-in plotting function in the *potential classes* of Ensembler.

2.3 APPLICATIONS AND EXAMPLES

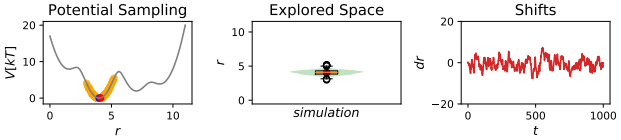
2.3.1 Simple Simulations

In the following, simple code examples are shown to introduce the usage of Ensembler. In addition, an application example is provided to illustrate the use of Ensembler for teaching about free-energy methods. The code for these examples can be found in the GitHub repository *https://github.com/rinikerlab/Ensembler/examples*.

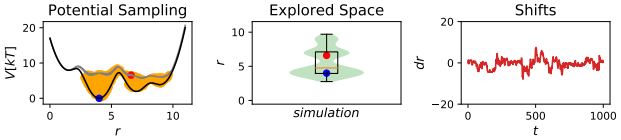
In typical applications of Ensembler, the user selects a potential-energy function from the available ones. In the following example, a potential-energy function with four wells is selected and initialized with chosen parameters. To sample this four-well potential-energy function with stochastic dynamics (SD),[165] the sampling method is instantiated and passed to the *system class*, which controls the execution of the simulation. The simulation is performed by calling the function *simulate* with the desired number of simulation steps passed as parameter. Subsequently, the results can be visualized using the built-in visualization functions that are compatible with the *simulation class* of Ensembler. As can be seen in Figure 2.3A, the energy barriers between the different minima were not crossed during the chosen simulation length. To overcome the sampling issue, enhanced sampling techniques can be employed.[172] In this example, local elevation[146]/metadynamics[147] is used to overcome the energy barriers (Figure 2.3B). The method adds a time-dependent biasing potential to the system, i.e. it adds a Gaussian biasing potential to positions that were already visited such that they become energetically less favorable. This decreases

the likelihood of visiting known positions again. The enhanced sampling technique can be applied by adding a single line of code compared to the previous simulation (Figure 2.3C).

(A) Standard Langevin Simulation



(B) Langevin Simulation with Local Elevation/Metadynamics



(C) Example Source Code

```
#Local elevation/metadynamics simulation:
##Imports
from ensembler.potentials.OneD import fourWellPotential, metadynamicsPotential
from ensembler.samplers.stochastic import langevinIntegrator
from ensembler.system import system
from ensembler.visualisation.plotSimulations import simulation_analysis_plot

##Simulation Setup
origpot = fourWellPotential(Vmax=4, a=1.5, b=4.0, c=7.0, d=9.0,  ah=2., bh=0., ch=0.5, dh=1.)
### Difference between Langevin and local elevation simulation
V = metadynamicsPotential(origpot, amplitude=0.35, sigma=0.5, n_trigger=15)
sampler = langevinIntegrator(dt=0.1, gamma=10, old_position=3)

system2=system(potential=V, sampler=sampler, start_position=4, temperature=1)

##Simulate
cur_state = system2.simulate(steps=1000,withdraw_traj=True, init_system=True)

##Visualize
positions = np.linspace(start=0, stop=10, num=1000) #phase space to be visualized
fig, out_path = simulation_analysis_plot(system=system2, title="Local Elevation/Metadynamics Simulation",
                                         limits_coordinate_space=positions)
```

FIGURE 2.3: Langevin simulation of a four-well potential energy-function. Results when sampling (1000 steps) with the standard SD integrator (**A**) or with local elevation [146]/metadynamics [147] (**B**). The left panel shows the potential-energy surface (black), the sampled range (orange), as well as the start point (blue) and end point (red). The middle panel shows the sampled space as a violin/box plot with the start point (blue) and end point (red). The right panel shows the shift $\Delta r_t = r_{t+1}$ - r_t as a function of simulation time t. (**C**) Source code to perform the simulations. First, the four-well *potential class* and the Langevin *sampler class* are initiated. Next, they are wrapped by a *system class*, which executes the simulation. Visualizations are generated with a built-in functions. Note that only one line has to be added to use the enhanced sampling technique (marked in bold).

2.3.2 Free-Energy Calculation

Free-energy calculation is an important field in computational chemistry because free-energy differences govern the outcome of processes in nature, e.g. protein-ligand binding or polymer formation.[36,135–137] The calculation of alchemical free-energy differences with Ensembler is exemplified with a mutation of the equilibrium position of a one-dimensional harmonic oscillator (Figure 2.4A). This mutation corresponds to a change of a covalent bond type at the terminus of a linear molecule and can be calculated analytically (Table 2.1). In practical applications, however, it is usually not possible to calculate the free-energy difference analytically. In these cases, MD-based simulation techniques can be employed. In the following, the sampling of the two end states of the model system and the results of the free-energy calculation with different methods are discussed. For more details, we refer to the Jupyter notebook in the Ensembler GitHub repository.

A simple free-energy method is to simulate one end state and estimate the free-energy difference with the Zwanzig equation.[141] The quality of the result depends on a sufficient phase-space overlap between the two end states.[188] Alternatively, one can simulate both end states separately and use BAR[168] (Figure 2.4A), yielding more converged results.[188] If the phase-space overlap between the two end states is not sufficient, more advanced sampling methods are necessary to obtain converged free-energy differences. One possibility to increase the phase-space overlap is to generate intermediate states as a linear combination of the two end states A and B with the coupling parameter λ, i.e. $H(\lambda) = (1 - \lambda)H_A + \lambda H_B$, such that $H(\lambda = 0) = H_A$ and $H(\lambda = 1) = H_B$. The intermediate states are positioned at discrete λ-points between 0 and 1 (Figure 2.4B).[189,190] The free-energy difference can be estimated using

FEP[141] or BAR[168] as the path over all intermediates, or with TI[140] as the integral along λ.

Another elegant free-energy method is EDS,[148,169] where a reference-state Hamiltonian H_r is sampled. H_r is constructed as a log-sum of the Hamiltonians of the two (or more) end states, guaranteeing the phase-space overlap of the reference state with all end states,

$$H_R = -\frac{1}{\beta s}\ln(e^{(-\beta s(H_A - E_A^R)} + e^{(-\beta s(H_B - E_B^R)}), \tag{2.1}$$

where $1/\beta = k_B T$, k_B being the Boltzmann constant and T the absolute temperature. H_R can be optimized for sampling using two kinds of parameters: The smoothing parameter s lowers the energy barriers between the end states, whereas the energy offsets E^R ensure equal weighting of all end states. In our example, both end states are sampled sufficiently during the EDS simulation with $s = 0.3$ and the energy offsets $E^R = [0, 0]$ (Figure 2.5A). Subsequently, the Zwanzig equation[141] is used to obtain the free-energy difference between the end states.[148,169] Recently, a hybrid form of EDS and λ-coupling was introduced, termed λ-EDS.[150] At $\lambda = 0$ or 1, the H_R is equal to the Hamiltonians of the respective end states, while conventional EDS is recovered with λ=0.5 (except for an offset).[150] λ-EDS allows for a λ-weighting of the exponential terms in the EDS equation. In the example in Figure 2.5B, the same reference-state parameters were used as before.

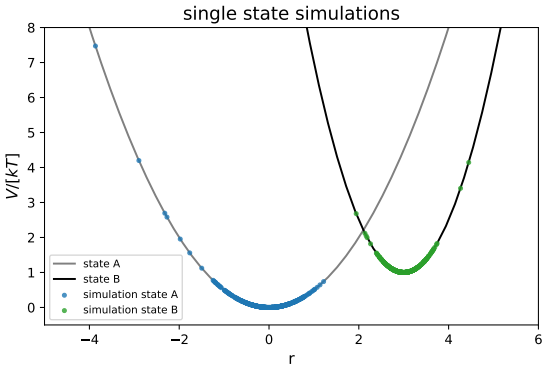
All free-energy calculations discussed above were performed with Ensembler for a total of 10'000 Monte Carlo (MC)[119] steps, and each simulation was repeated five times. The simulation results listed in Table 2.1 show that larger errors are obtained without intermediate states due to insufficient phase-space overlap. Using ten λ–intermediate states together with TI gave the best

result, however, this approach is also the computationally most expensive one (i.e. ten separate simulations). EDS and λ-EDS, on the other hand, yielded also good results, while requiring only one simulation (given a set of suitable reference-state parameters). We refer to the Jupyter notebook in the Ensembler GitHub repository for the source code, more detailed information on these methods as well as additional methods like conveyor-belt TI[171] and RE-EDS,[170,191] which combine enhanced sampling and free-energy methods.

TABLE 2.1: Estimated free-energy difference for the model system shown in Figure 2.4 and Figure 2.5. Sampling was performed with Monte Carlo method (MC)[119] for 10'000 steps in each simulation. Each calculation was replicated five times and the averaged result is shown together with the standard deviation. The duration of the computations (without visualizations) was estimated directly in the Jupyter notebook and is given relative to the FEP simulation (absolute duration = 2.0 seconds). The performance was tested on a Lenovo Thinkpad T420s with an Intel i5-2520 (2.5 GHz) CPU and 8 GB RAM. The RAM usage for the full Jupyter notebook execution was in total 578 MB.

Method	Average ΔF $[k_BT]$	Deviation from analytical result $[k_BT]$	Speed (rel. to FEP) Simulation	Analysis
analytical	1.275	-	-	-
FEP[141]	6.579 ± 1.009	5.305	1.0	0.1
BAR[168]	2.437 ± 0.500	2.437	3.0	2.1
FEP 10-λ-points	1.406 ± 0.431	0.131	14.0	0.7
TI[140] 10-λ-points	1.242 ± 0.015	0.033	14.0	0.04
EDS[136,148,169]	0.958 ± 0.110	0.317	2.4	0.2
λ-EDS[150] $\lambda = 0.5$	0.987 ± 0.111	0.287	3.1	0.2

(A)



(B)

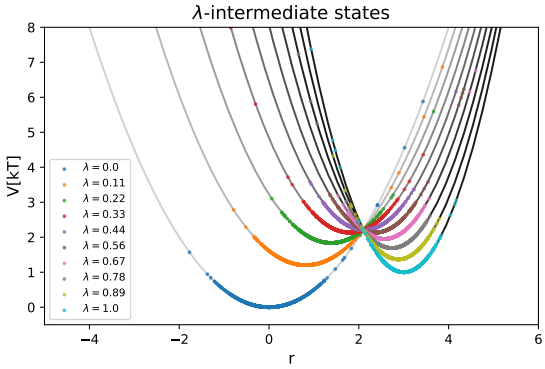
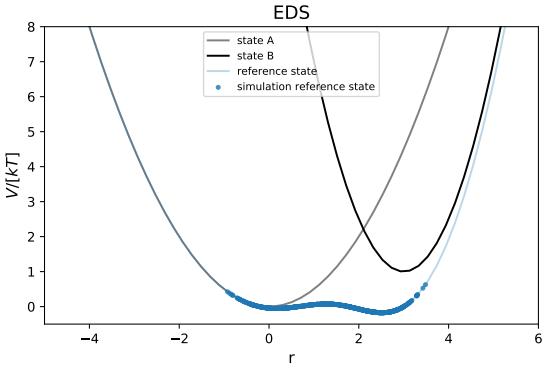


FIGURE 2.4: Illustration of different free-energy methods implemented in Ensembler (part I). (a) For FEP[141] and BAR,[168] the two end states (grey and black) are sampled separately (green and blue). (b) To increase the phase-space overlap, the two end states can be coupled as a linear combination of their Hamiltonians using a coupling parameter λ. This allows the generation of intermediate states (grey to black) and sampling of those (colored points).

(A)



(B)

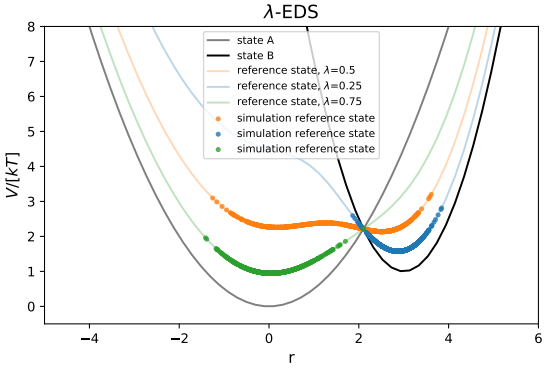


FIGURE 2.5: Illustration of different free-energy methods implemented in Ensembler (part II). (**A**) An alternative method is EDS,[136,148,169] where a reference-state Hamiltonian (blue line) is sampled (blue points), which envelopes the end states. By setting the reference-state parameters to $s = 0.3$ and energy offsets = [0,0], all relevant phase-space regions can be sampled. (B) A recently developed approach called λ-EDS[150] introduces a λ-dependence in the EDS method (blue, orange and green line). Colored points indicate sampling. The reference-state parameters were set to $s = 0.3$ and energy offsets = [0,0], and three different lambda values 0.25,0.5, 0.75 were chosen.

2.4 CONCLUSION

In this work, we introduced the Ensembler package as a tool to support teaching of MD simulations and free-energy techniques, and to enable rapid prototyping of new methods using 1D or 2D model systems. The package provides a large set of implemented methods for comparison. The open-source basis, the lean structure, and the simplicity of Python3 form a convenient and efficient framework. The code examples and application example for free-energy calculation highlight the ease of using Ensembler. With this, Ensembler can contribute to improving the shareability, comparability, and reproducibility for method development in our field.

3

RestraintMaker: A Graph-Based Approach to Select Distance Restraints in Free-Energy Calculations with Dual Topology *

"Allzu straff gespannt, zerspringt der Bogen."
"And much too tightly stretched the bow will split."

Rudenz in Wilhelm Tell
Friedrich Schiller[192]

The calculation of relative binding free energies (RBFE) involves the choice of the end-state/system representation, of a sampling approach, and of a free-energy estimator. System representations are usually termed "single topology" or "dual topology". As the terminology is often used ambiguously in the literature, a

* This Chapter is reproduced in part from Benjamin Ries†, Salomé Rieder†, Clemens Rhiner, Philippe H. Hünenberger, and Sereina Riniker, J. Comput.-Aided Mol. Des., submitted (2021). † These authors contributed equally.

systematic categorization of the system representations is suggested here. In the dual topology approach, the molecules are simulated as separate molecules. Such an approach is relatively easy to automate for high-throughput RBFE calculations compared to the "single topology" approach. Distance restraints are commonly applied to prevent the molecules from drifting apart, thereby improving the sampling efficiency. In this chapter, we introduce the program RestraintMaker, which relies on a greedy algorithm to find (locally) optimal distance restraints between pairs of atoms based on geometric measures. The algorithm is further extended for multi-state methods such as enveloping distribution sampling (EDS) or multi-site λ-dynamics. The performance of RestraintMaker is demonstrated for toy models and for the calculation of relative hydration free energies. The Python program can be used in script form or through an interactive GUI within PyMol. The selected distance restraints can be written out in the GROMOS or GROMACS file formats. Additionally, the program provides a human-readable JSON format that can be easily parsed and processed further. The code of RestraintMaker is freely available on GitHub `https://github.com/rinikerlab/restraintmaker`.

3.1 INTRODUCTION

Recent methodological developments have improved the statistical robustness and the degree of automation of relative binding free-energy (RBFE) calculations, which are now routinely applied in drug discovery projects in industry.[34–37,54,135,193–199] A free-energy calculation provides information about the relative populations of multiple end-states in equilibrium. Examples are drug design, where the end-states represent the different ligands that bind to a protein,[31,136,170,195,200–205] or protein engineering, where the end-states correspond to the different amino acids considered for one position in the protein.[206–208] Each free-energy calculation involves the choice of a sampling approach, a free-energy estimator (e.g. TI,[140] the Zwanzig equation,[141] or BAR[168]), and a representation of the end-states (i.e., molecules or substructures of molecules) during the simulation.

Several possible representations have been proposed in the past to build a coordinate and topology space of the end-states. Historically, two approaches emerged, which were termed "single topology"[209,210] and "dual topology".[209,211] Unfortunately, the terminology is not always clear in the literature and these terms are used ambiguously.[212–214] To distinguish the different representation options, we propose here a terminology based on the difference in the respective coordinate space (Figure 3.1). These definitions may differ from the historical ones. The single topology approach contains a single set of coordinates for both end states. In contrast, the dual topology approach involves a separate set of coordinates for each end state. The two approaches can be seen as opposite extremes. Three sub-variants of the dual topology approach can be found in the literature: linked, separated and

unconstrained. In addition, a "hybrid topology" approach was recently described,[204] which presents an intermediate between the single and dual approaches (Figure 3.1). This scheme has been used in many studies for RBFE calculations before but not called hybrid topology. In protein engineering, Shobana *et al.*[206] called a similar approach hybrid topology.[206] The different representations vary with respect to sampling efficiency and the capability of handling complex transformations.

With the high-throughput application of RBFE calculations comes the need for automation.[34] While there exist tools such as FESetup,[215] ProtoCaller,[216] SMArt,[217] or LOMAP,[218] to automatically set up single-topology RBFE calculations, the dual topology approaches are in principle the easiest to automate as any alchemical molecule transformation can be realized without requiring atom mapping.[213] For the unconstrained dual topology variant, an automatic set-up procedure is available in the packages pyAutoFEP[219] and FEW.[220] When representing the end-states with a linked dual topology approach, the set-up is more difficult than in the unconstrained case as the distance restraints between the molecules need to be chosen. For example, the QligFEP pipeline[195] provides an automatic system generation for the linked dual topology approach, where the distance restraints are placed in the perturbed common substructure of the end-states. These distance restraints only become active if the restrained atoms surpass a distance of 0.02 nm. However, for more complex transformations (e.g. in scaffold hopping), a more flexible approach is needed to select the optimal distance restraints between molecules.

In this chapter, we present a greedy algorithm to select optimal distance restraints for RBFE calculations with the linked dual topology approach, which is also applicable to molecule pairs

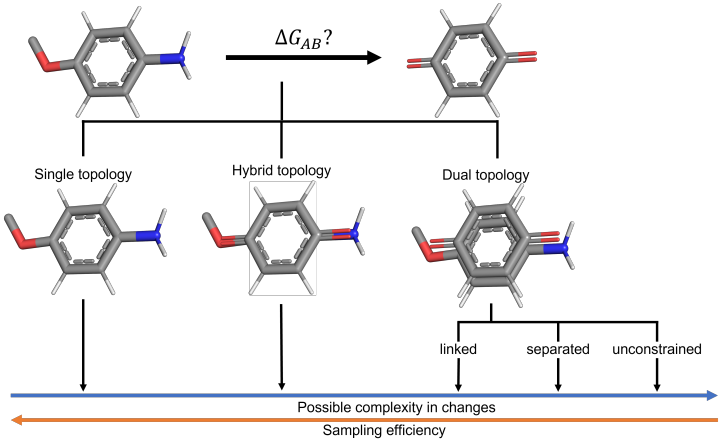


FIGURE 3.1: Three end-state representations can be distinguished based on the coordinate space. The "single topology" approach (left) contains a single set of coordinates for all end-states. The "dual topology" approach contains separate sets of coordinates for each end state (right). The "hybrid topology" approach (middle) combines atoms of common substructures into one coordinate set, while atoms that differ between the end-states are represented separately. It is therefore an intermediate between the two other representations. The dual topology approach can be further subdivided into three sub-variants: linked, separated, and unconstrained. The linked dual topology approach is closest to the single topology approach, as the coordinate overlap between the end-states is ensured with direct spatial restraints (e.g. distance restraints). The separated variant is connecting the molecules indirectly by restraining them spatially to the same area, whereas the unconstrained variant does not restrain the molecules at all and is therefore also the most difficult to sample.

without a common core. In addition, the algorithm is extended to solve the same problem for multi-state RBFE methods such as enveloping distribution sampling (EDS)[148,169] and multi-site λ-dynamics,[221] resulting in a linked multi-topology approach. Finally, we analyze the sampling behavior and performance of the approach for the calculation of relative hydration free energies.

The algorithm is implemented in a Python package (`https://github.com/rinikerlab/restraintmaker`), which can be used as a scripting library or with a GUI inside PyMOL.[222]

3.2 THEORY

3.2.1 End-State Representations

In the following, we provide the categorization of the different system representation approaches (Figure 3.2).

Single Topology

The single topology approach was first used by Jorgensen *et al.*[223] to calculate the relative hydration free energies of methanol and ethane. The approach was later termed "single topology" by Pearlman *et al.*[209] The end-states are represented by the union of the coordinates of the molecules, limiting the possible transformations that can be handled by this method. Usually, perturbations for a single topology approach include both atom types (i.e. van der Waals parameters and/or partial charges) and bonded terms.[151,201,202,209,210,212,223–227] A single topology approach in this definition is constructed as follows for two end-states A and B with the two molecules a and b (Figure 3.2),

$$dim(\mathbf{r}^{AB}_{single}) = dim(\mathbf{r}^{(a \cup b)})$$
$$\mathbf{r}^{AB}_{\text{single}} = \{\mathbf{r}_1{}^{AB}, \mathbf{r}_2{}^{AB}, ..., \mathbf{r}^{AB}_{dim(\mathbf{r}^{AB}_{single})}\}$$

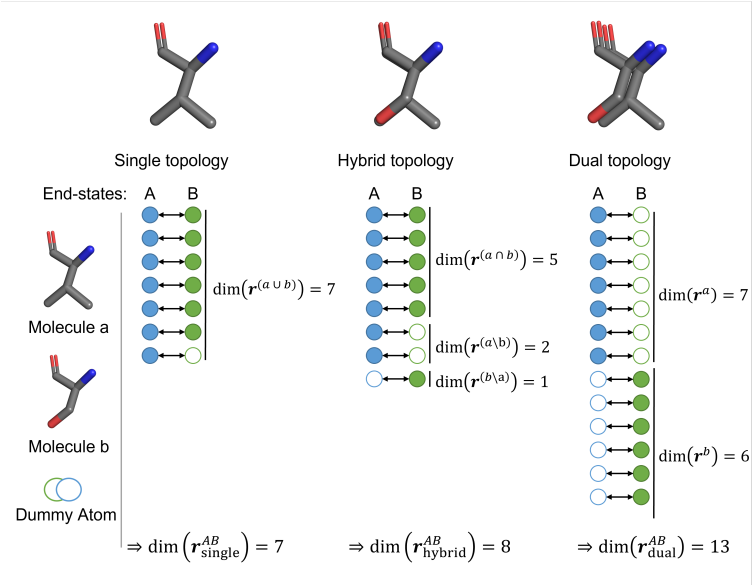


FIGURE 3.2: The three end-state representations can be illustrated using the coordinate mapping of molecules a and b in the end-states A and B, respectively. The smallest number of coordinates is required for the single topology coordinate space ($dim(\mathbf{r}^{AB}_{\text{single}})$) as the coordinate set is formed from the union of all coordinates (left). If a coordinate is only used in one end-state, it becomes a non-interacting dummy atom in the other end-state. The hybrid topology approach (middle) requires more coordinates for its coordinate space ($\mathbf{r}^{AB}_{\text{hybrid}}$), as the coordinates of differing atoms are represented separately. The largest coordinate space is required for the dual topology approach. Here, the coordinate space ($dim(\mathbf{r}^{AB}_{\text{dual}})$) is the sum of both molecules.

where $dim(\mathbf{r}^{AB}_{single})$ is the total number of atom coordinates of the end-states A and B, and $\mathbf{r}$ the coordinate space vector. Note that the unperturbed atoms of the environment (i.e., solvent and/or protein atoms), were excluded for simplicity.

The single topology approach has in principle the best sampling efficiency compared to other representations as it is con-

structed with the least amount of coordinates and therefore the fewest degrees of freedom are perturbed.[203,210,214,225] However, an issue arises when the molecules do not only differ in the type of atoms but also in their number. To address this, a non-interacting "dummy" state can be assigned to the vanishing atom(s).[203,210,214,225] Different variants of dummy states are possible. Typically, only the non-bonded interactions are removed. However, it has been shown that also the bonded terms of "dummy" states can influence the free-energy calculations.[214] The construction of a single topology becomes increasingly challenging with more complex molecule transformations. For example, to realize complex transformations such as ring size changes or ring opening/closing, special soft-bond terms had to be implemented.[202]

Hybrid Topology

The term hybrid single-dual topology was used by Jiang *et al.*[204] in 2019 to describe the combination of a single-topology core (common among the molecules) with dual-topology substituents (differing among the molecules). However, similar schemes were already used in many previous studies (although called either single or dual topology).[200,206,217,228–230] A hybrid topology approach in this definition can be constructed as follows for two end-states A and B with the two molecules a and b (Figure 3.2),

$$dim(\mathbf{r}^{AB}_{hybrid}) = dim(\mathbf{r}^{\,(a\,\cap\,b)}) + dim(\mathbf{r}^{\,(a\setminus b)}) + dim(\mathbf{r}^{\,(b\setminus a)})$$

$$\begin{aligned}\mathbf{r}^{AB}_{\text{hybrid}} = \{&\mathbf{r}_1^{(a\,\cap\,b)}, \mathbf{r}_2^{(a\,\cap\,b)}, ..., \mathbf{r}^{(a\,\cap\,b)}_{dim(\mathbf{r}^{\,(a\,\cap\,b)})},\\ &\mathbf{r}_1^{(a\setminus b)}, \mathbf{r}_2^{(a\setminus b)}, ..., \mathbf{r}^{(a\setminus b)}_{dim(\mathbf{r}^{\,(a\setminus b)})},\\ &\mathbf{r}_1^{(b\setminus a)}, \mathbf{r}_2^{(b\setminus a)}, ..., \mathbf{r}^{(b\setminus a)}_{dim(\mathbf{r}^{\,(b\setminus a)})},\}\end{aligned}$$

Hybrid topology approaches aim a combining the advantages of

single and dual topology, i.e., to keep the number of perturbed degrees of freedom minimal for sampling efficiency while facilitating more complex transformations.

Dual Topology

In a dual topology, two fully separate sets of coordinates are used for the molecules. This approach was first introduced by Gao *et al.*,[211] and termed later on "dual topology" by Pearlman *et al.*[209] The atoms of molecule a, which are fully interacting in end-state A, are transformed to the dummy state in end-state B, and vice versa. Importantly, the atoms of the molecules do not interact with each other and only share the same environment.[200,213] In such dual topology approaches, only the non-bonded terms are usually perturbed.[170,200,212,231] A dual topology approach in this definition can be constructed as follows for the end-states A and B with the two molecules a and b (Figure 3.2),

$$dim(\mathbf{r}_{\text{dual}}^{AB}) = dim(\mathbf{r}^{\,a}) + dim(\mathbf{r}^{\,b})$$
$$\mathbf{r}_{\text{dual}}^{AB} = \{\mathbf{r}_1^{\,a}, \mathbf{r}_2^{\,a}, ..., \mathbf{r}_{dim(\mathbf{r}^{\,a})}^{\,a}, \mathbf{r}_1^{\,b}, \mathbf{r}_2^{\,b}, ..., \mathbf{r}_{dim(\mathbf{r}^{\,b})}^{\,b}\}$$

The separated coordinates lead to a larger number of atoms in the system and thus, more degrees of freedom are perturbed, lowering the sampling efficiency compared to single topology approaches. Three sub-variants of the dual topology approach can be distinguished depending on how this issue is addressed in practice: (i) the linked variant with direct spatial restraints between the molecules to prevent them from drifting apart during the simulation,[170,195,200,232] (ii) the separated variant with restraining to the environment,[213,233] and (iii) the unconstrained variant.[219,234] The linked dual topology is in principle the most efficient variant

if the transformation is relatively simple (no changes in binding modes induced by reorientation or large conformational differences). The separated dual topology approach is expected to be less efficient than the linked variant, but can handle these very challenging transformations.[233] A significant advantage of the dual topology approach is the straightforward set-up of a system compared to the single and hybrid topologies, especially for more complex transformations or multiple end-states.

3.2.2 Automated Placement of Distance Restraints

To facilitate the set-up of RBFE calculations with the linked dual topology approach, the selection of optimal distance restraints between the molecules needs to be automated. The proposed algorithm is based on classical graph algorithms. Its goal is to identify suitable placements for the distance restraints between two molecules m_i and m_j. The following conditions are applied:

1. m_i and m_j are pre-aligned to each other
2. Optimal placement of distance restraints requires that
   (a) the restrained atom pairs are maximally distant over the two molecules
   (b) the restrained atoms have a small distance to each other in the aligned structures
   (c) the restraints do not influence the conformational sampling of the molecules
3. For a user-defined number of required restraints n_{res}, it holds that $n_{\text{res}} \ll n_{m_i}$ and $n_{res} \ll n_{m_j}$, where n_i and n_j are the numbers of atoms of molecules m_i and m_j, respectively

From these conditions follows that only atoms in relatively rigid regions of the molecules such as rings should be selected for the restraint search space. While restraining non-ring atoms might be favorable for the maximally distant distribution of the restrained atoms over the molecules, it is more likely to distort the conformational sampling of the molecules. The steps of the algorithm are shown schematically in Figure 3.3 and explained in the following subsections.

Assigning Distance Restraints for a Pair of Molecules

Translation into a graph problem: The developed algorithm is based on a graph representation of the restraint search space. To solve the problem of selecting an optimal set of atom pairs to define the distance restraints, it needs to be translated into a graph G fulfilling,

$$G(N, E, \omega), E \subseteq \{\{x, y\} \mid x, y \in N \text{ and } x \neq y\}, \tag{3.1}$$

where N is a set of nodes, E a set of edges, and ω a set of weights.

We consider a molecule pair consisting of the two molecules m_i and m_j, with their sets of atoms A_i and A_j, respectively. In a first step, existing algorithms such as e.g. implemented in the RDKit[235] or PyMol[222] can be used to align the two molecules to each other. The alignment can for example be based on the maximum common substructure (MCS), or in the case of scaffold hopping the maximal overlap of the van der Waals volumes of the molecules. Next, the sets of possible atoms A_i and A_j for the distance restraints are reduced to the ring atoms of the respective molecules, A_i^{ring} and A_j^{ring}. In addition, a user-defined cutoff distance d_{res} between the atom pairs is introduced (here, $d_{\text{res}} = 0.1$ nm). Therefore, a possible distance restraint is a pair of atoms $(a_i^{\text{ring}}, a_j^{\text{ring}})$ that

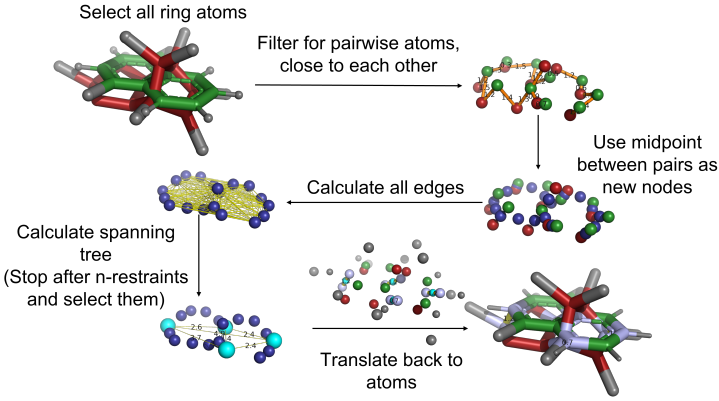


FIGURE 3.3: Schematic illustration of the algorithm steps to identify optimal placements for the distance restraints between a pair of molecules. The described algorithm uses a set of possible atoms (here these are the ring atoms). Next, possible restraints are filtered by the user-defined atom distance cutoff d_{res}. After this filtering step, the midpoints of the remaining possible restraints are calculated and used further as nodes of a graph. These nodes are connected by edges that have assigned weights, corresponding to the Euclidean distance of the midpoints. From this, a spanning tree is built with a min-max decision scheme. The construction of the spanning tree is stopped after n_{res} iterations or if all nodes were connected. The result is a set of optimal restraints, $C_{\text{res,opt}}$, which will be translated back to an atom selection for further use in MD packages.

fulfills the distance criterion $d(a_i^{\text{ring}}, a_j^{\text{ring}}) \leq d_{\text{res}}$.

The possible distance restraints C_{res} are used as nodes N to construct a fully connected graph G. Each individual restraint c_{res} is represented by the midpoint of the two involved atoms. The undirected edges of G have as weight $\omega_{ji}^{\text{dist}}$ the Euclidean distance between the midpoints of the two atom pairs $\omega_{ji}^{dist} = d(c_{\text{res}_j}, c_{\text{res}_i})$.

Solving the graph problem: From the generated graph, only a subset of restraints fulfills the conditions 1-3 listed above. To

obtain a relevant subset of restraints, we decided to use a min-max decision scheme inspired by the minimax theorem[236] to build a spanning tree (i.e. a subset of restraints, $C_{\text{res,opt}}$) within a greedy Prim-like approach.[237]

The algorithm starts by picking the edge in the graph G with the largest weight $\omega_{ij}^{\text{dist}}$ (distance), i.e. the two restraints whose midpoints are the furthest apart. After this initial selection of two restraints for $C_{\text{res,opt}}$, the weights of the edges in G are updated with the minimal distance of all c_{res} in $C_{\text{res,opt}}$ to a respective node n_i. Subsequently, all edges and nodes are removed, which contain atoms that are already selected in $C_{\text{res,opt}}$. After the update of the edge weights, the restraint c_{res} with maximal $\omega_{ji}^{\text{dist}}$ is added to $C_{\text{res,opt}}$. This procedure is repeated until either $|C_{\text{res,opt}}| = n_{\text{res}}$ (in practice we expect a rather small number for n_{res}, typically $4 < n_{\text{res}} < 10$) or all remaining nodes are connected.

Back-mapping: The selected subset of n_{res} restraints, $C_{\text{res,opt}}$ is mapped back to the atoms in the molecules, such that the distance restraints can be written in a format readable by MD packages such as GROMOS[94] or GROMACS.[143] Additionally, a JSON[238] format is provided, allowing to import the results with any standardized JSON-Parser.

Tie-breaking: Due to non-perfect alignment and finite numerical precision, a tie between multiple restraints can occur, i.e. they have a very similar distance to the already selected restraints. This practical problem was solved by adding a tie-breaker that detects whether multiple high-priority restraints are within a range of 0.02 nm in a given iteration step and refines the decision result by applying a second criterion. For each candidate restraint in an iteration step, the distance to the center of geometry (COG) of

all already selected restraints is calculated, and the restraint is chosen for which this distance is maximal.

Extension to Multiple End-states

For multi-state methods such as EDS,[148,169] replica-exchange EDS (RE-EDS),[170,191,239] multi-site λ-dynamics,[221] or multi-state λ-LEUS,[240] more than two molecules need to be restrained to each other. Based on our experience with RE-EDS, it is best to apply the distance restraints between multiple molecules in form of a ring, i.e. each molecule is restrained to two neighboring molecules.[239] This scheme is used in the following.

In a first step, the pairwise greedy algorithm is used to calculate an optimal set of distance restraints between all possible molecule pairs, building up a fully connected graph (Figure 3.4). The possible sets of restraints are subsequently compared to each other by calculating the convex hull around the coordinates of all the restraint midpoints. The convex hull volume (CHV) is then assigned to the edges of the fully connected graph as weight ω^{CHV}. The optimal chain of connected molecules is determined by applying another greedy algorithm, inspired by the Kruskal algorithm,[241] which picks the edges with the largest CHVs without forming cycles or branches (Figure 3.4). The chain is closed to a ring by tying the loose ends together. This last molecule pair may have a less good set of restraints.

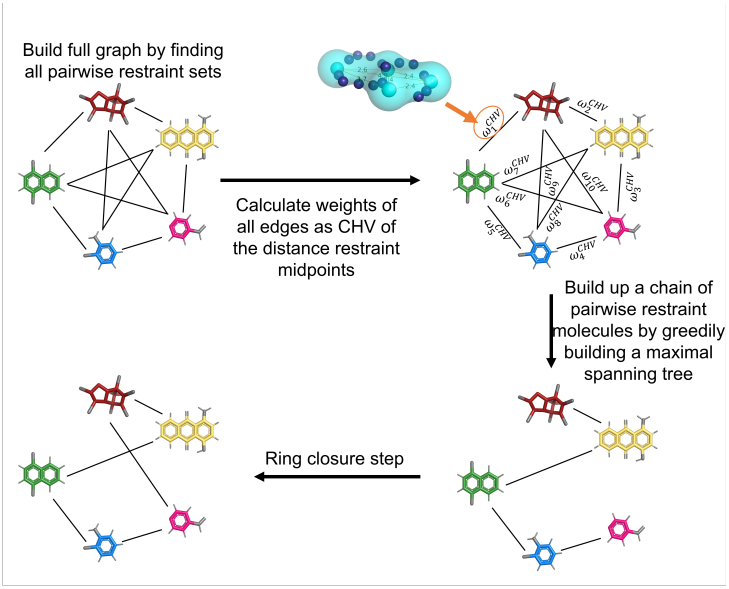


FIGURE 3.4: Schematic illustration of the algorithm steps to link multiple end-states by distance restraints for a multi-state RBFE calculation. The selection is carried out in four steps. (i) Optimal restraints are calculated for all possible molecule pairs, building up a fully connected graph. (ii) The weights ω_i^{CHV} of the edges are calculated as the convex hull volume (CHV) formed by the selected restraints. (iii) A maximum spanning tree without branching is greedily constructed by selecting the edges with maximal weights. (iv) The ring is closed by connecting the ends of the chain.

3.2.3 FREE-ENERGY METHODS

Two free-energy methods were tested with the linked dual topology approach.

THERMODYNAMIC INTEGRATION

TI is a standard method to estimate free-energy differences,[140] where a λ-dependent path between the two end-states A and B is

sampled. The potential energy of the system is constructed as,

$$V(\mathbf{r};\lambda) = (1-\lambda)\ V_A(\mathbf{r}) + \lambda\ V_B(\mathbf{r}) \tag{3.2}$$

End-state A is obtained when $\lambda = 0$, and end-state B when $\lambda = 1$. In practice, simulations at discrete λ-points between 0 and 1 are performed, and the free-energy difference is obtained by numerical integration,

$$\Delta G_{BA}^{rel} = \int_0^1 \left\langle \frac{\partial V(\lambda)}{\partial \lambda} \right\rangle_\lambda d\lambda \tag{3.3}$$

Replica-Exchange Enveloping Distribution Sampling

RE-EDS[170,191,239] is a combination of Hamiltonian replica exchange[184,242] and EDS.[148,169] In EDS, a reference state Hamiltonian V_R is sampled, which combines N end-states as,

$$V_R\left(\mathbf{r}; s, \mathbf{E}^R\right) = -\frac{1}{\beta s} \ln\left[\sum_{i=1}^{N} e^{-\beta s\left(V_i(\mathbf{r}) - E_i^R\right)}\right] \tag{3.4}$$

where s is the smoothness parameter, E_i^R a set of energy offsets, and $\beta = 1/(k_B\ T)$, with k_B being the Boltzmann constant and T the absolute temperature. The force on a particle k can be calculated as,[148,169]

$$\mathbf{f}_k(t) = -\frac{\partial V_R(\mathbf{r}; s, \mathbf{E}^R)}{\partial \mathbf{r}_k} = \sum_{i=1}^{N} \frac{e^{-\beta s(V_i(\mathbf{r}) - E_i^R)}}{\sum_{j=1}^{N} e^{-\beta s(V_j(\mathbf{r}) - E_j^R)}} \left(-\frac{\partial V_i(\mathbf{r})}{\partial \mathbf{r}_k}\right). \tag{3.5}$$

For high s-values (close to 1.0), the sampling of the reference state is dominated by the end-state with the lowest value of $V_i(\mathbf{r}) - E_i^R$, whereas for small s values (close to zero), all end-states contribute

to the forces, resulting in the so-called "undersampling".[200]

The free-energy difference between a pair of end-states in the system is then calculated as,

$$\Delta G_{BA}^{rel} = -\frac{1}{\beta} \ln \frac{\left\langle e^{-\beta(V_B - V_R)} \right\rangle_R}{\left\langle e^{-\beta(V_A - V_R} \right\rangle_R} . \tag{3.6}$$

In practice, an optimal choice of s and E_i^R is essential to sufficiently sample all end-states in an EDS simulation. RE-EDS overcomes the difficulty of choosing an optimal s-value by simulating multiple replicas with different s-values and performing replica exchanges between them.[170,191]

3.3 COMPUTATIONAL DETAILS

3.3.1 Validation of the Restraint Selection Algorithm

To assess the performance of the greedy algorithm for selecting optimal distance restraints between two molecules, it was first tested on toy models. These contained 12 to 30 particles that were randomly distributed in space. The particles were randomly assigned to two entities representing two molecules. A selection of four restraints was performed with no pre-processing steps. Different algorithmic approaches were compared: the developed greedy algorithm, random selection (100 repetitions), and two brute-force approaches. One of the brute-force approaches maximizes the sum of the restraint midpoint distances between the selected restraints by considering all possible quadruples of restraints explicitly (BF-maxD). The other one maximizes the CHV

around the selected restraints (BF-maxCHV), as done for chaining in multi-state systems. Each number of particles was sampled 20 times (using different particle coordinates each time) to provide an uncertainty estimate. The scripts for this validation are available in the example folder of the GitHub repository (*examples/publication/a_benchmark_algorithms*).

3.3.2 Molecules with Hydration Free Energies

The algorithm was applied for the calculation of relative hydration free energies $\Delta\Delta G_{\mathrm{hyd}}$ (Figure 3.5). A set of 16 molecules with experimentally available hydration free energies[243–248] were selected (Table 3.1). The topologies for these molecules were taken from the ATB server.[104] The selected molecules are small and possess a ring core. The corresponding pairwise transformations are nevertheless relatively complex, and involve R-group and ring-size changes as well as scaffold hopping-type transformations (e.g. benzene to cyclohexane).

Table 3.1: Identifier of the ATB server,[104] IUPAC name, and canonical SMILES for the 16 molecules with experimental hydration free energies.

Ligand	Identifier	IUPAC name
1	_O6T	1,2-dimethoxybenzene
2	_O70	(2R,5R)-2-methyl-5-prop-1-en-2-ylcyclohexan-1-one
3	_O71	(1S,5R)-2-methyl-5-prop-1-en-2-ylcyclohex-2-en-1-ol
4	_P8I	cyclopentanone
5	6J29	1-amino-4-hydroxyanthracene-9,10-dione
6	6KET	3-methoxyphenol
7	8018	(1R,2S,3R,4R,6S,7S)-1,3,4,7,8,9,10,10-octachlorotricyclo[5.2.1.02,6]dec-8-ene
8	E1VB	[1,2,2-Trifluoroethoxy]benzene
9	F313	4-methoxyaniline
10	G078	1,4-dimethylnaphthalene
11	G277	cyclohexa-2,5-diene-1,4-dione
12	M030	1,3,5-trimethylbenzene
13	M097	2-chloroaniline
14	M218	N-methylaniline
15	S002	bromomethylbenzene
16	TVVS	pyridine-4-carbaldehyde

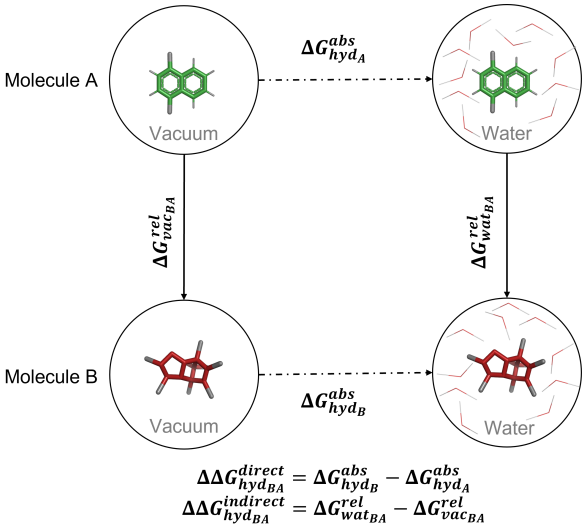


FIGURE 3.5: Thermodynamic cycle for the calculation of relative hydration free energies $\Delta\Delta G_{\mathrm{hyd}_{AB}}$. The direct way to obtain $\Delta\Delta G_{\mathrm{hyd}_{AB}}$ employs two absolute free-energy calculations giving $\Delta G^{\mathrm{abs}}_{\mathrm{hyd}_A}$ and $\Delta G^{\mathrm{abs}}_{\mathrm{hyd}_B}$. The indirect way uses two alchemical or relative free-energy calculations giving $\Delta G^{\mathrm{rel}}_{\mathrm{vac}_{AB}}$ and $\Delta G^{\mathrm{rel}}_{\mathrm{wat}_{AB}}$.

Pairwise TI calculations were carried out with a linked dual topology approach for the 16 molecules in a star-like scheme with molecule **12** as center, resulting in 15 relative hydration free energies (Figure 3.6).

3.3.3 Simulation Details

All simulations were carried out using the MD software package GROMOS[94] version 1.5.0 (freely available on `http://www.gromos.net`),[239] the Python RE-EDS pipeline (`https://github.com/rinikerlab/reeds`) and PyGromosTools[249] (`https://github.com/rinikerlab/PyGromosTools`).

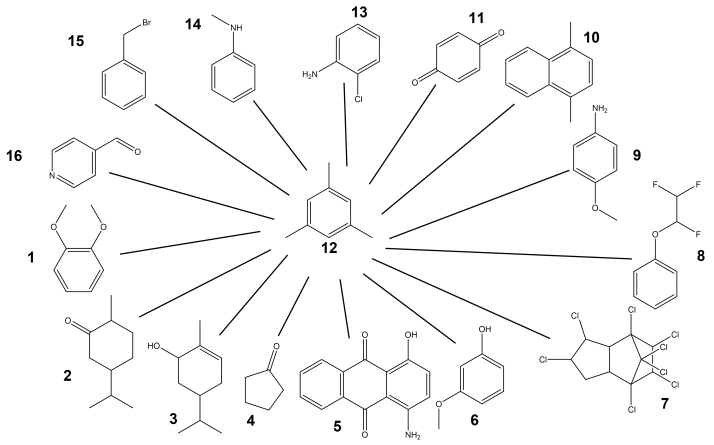


FIGURE 3.6: Set of 16 molecules with experimental hydration free energies available.[104,243–248] The black lines indicate the pairs of molecules for which TI calculations were performed. RestraintMaker was used to select pairwise distance restraints between the central molecule and all others.

In order to compare our results with the absolute hydration free energies reported in the ATB server,[104] the same simulation setup was used. The simple point-charge (SPC) model[250] was employed for water. A single cutoff radius of 1.2 nm was used for the calculation of the non-bonded interactions. The integration time step was set to 2 fs and the pairlist was updated every five steps. Long-range nonbonded interactions were calculated using a reaction-field correction[88] with $\varepsilon_{\mathrm{rf}} = 1$ for the simulations in vacuum and $\varepsilon_{\mathrm{rf}} = 61$ for the simulations in water.[251] The force constant for the distance restraints was set to 5000 kJ/(mol·nm^2).

THERMODYNAMIC INTEGRATION

The topologies and coordinate files of the single states were obtained from the ATB server[104] and were merged to pairs us-

ing PyGromosTools.[249] The coordinates of the different single molecules were aligned to each other using the common molecular skeleton of the molecules (only rings), with the *align* function in RDKit.[235] After the alignment, RestraintMaker was used to place four restraints with $d_{\text{res}} = 0.1$ nm (Figures 3.7.

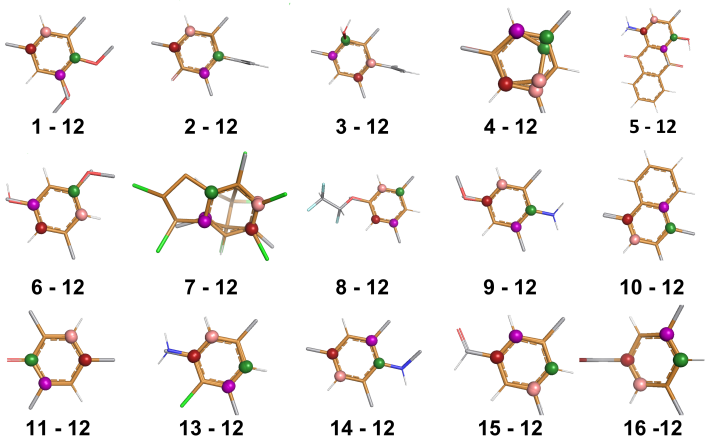


FIGURE 3.7: Selected distance restraints (colored spheres) for the 15 pairwise TI calculations with molecule **12** (i.e. 1,3,5-timethylbenzene) as the central molecule (Figure 3.6). For each pair, four distance restraints were determined with the greedy algorithm.

The computational boxes for the simulations in water were generated with the GROMOS++[252] program *simbox* using a minimal solute-to-wall distance of 0.8 nm, and relaxed by energy minimization. The scripts can be found in the example folder on Github (*https://github.com/rinikerlab/restraintmaker/tree/main/examples/publication/b_ATB_solvationFreeEnergies*). The TI calculations were carried out with 21 evenly spaced λ-points between 0 and 1, both for the molecules in water and in vacuum. Each λ-point was equilibrated for 1 ns, followed by a production run of 5 ns.

The free-energy differences were calculated using the Simpson integration implemented in the SciPy library.[178]

3.3.4 Analysis

The analysis of the simulations was carried using GROMOS++[252] and PyGromosTools.[249] In addition, the following Python packages were employed for the statistical analysis and plotting: Pandas,[181] Matplotlib,[182] NumPy,[179] SciPy,[178] mpmath,[253] and Jupyter notebooks.[173]

3.4 Results and Discussion

3.4.1 Validation of the Restraint Selection Algorithm

As a greedy algorithm, the approach in RestraintMaker might lead to sub-optimal solutions. To test this, the performance of the algorithm was compared with that of two brute-force approaches. One brute-force approach, BF-maxD, maximizes the distance of all selected restraint midpoints to each other, whereas the second brute-force approach, BF-maxCHV, uses the CHV spanned by the selected restraint midpoints as a criterion. Toy systems consisting of two strongly overlapping particle clouds were constructed, for which four restraints should be selected. The systems varied in the number of randomly placed particles. Each particle mimics an atom of a hypothetical molecule that might be selected to be restrained.

The advantage of the greedy algorithm is evident when considering the time complexity as the brute-force approaches scale with $\mathcal{O}(N^4)$ (where N is the number of atoms), making them unusable for larger molecules (Figure 3.8A). For selecting four restraints in the 20 particles toy system, the BF-maxCHV requires 75 s (single core), and 3325 s for 30 particles. In comparison, the greedy algorithm requires only 0.031 s for the 30 particles.

Comparison of the sum of all distances between restraint midpoints shows that BF-maxD (optimizing for the maximal distance between all midpoints of the selected restraints) yields the best results with the largest distances (blue line in Figure 3.8B). The second brute-force algorithm BF-maxCHV (optimizing for the CHV defined by the restraint midpoints) and our greedy algorithm give comparable results to BF-maxD. All three approaches are very good at maximizing the distance between the selected restraints. Random selection with 100 trials (negative control), on the other hand, performs significantly worse. Second, we compared the approaches based on the CHV generated by the selected restraints (Figure 3.8C). Here, the BF-maxCHV approach yields the best results as expected, while BF-maxD and the greedy algorithm perform similar to each other. The difference in CHV between the latter two approaches and BF-maxCHV increases with increasing number of particles in the toy system. This may be due to the growing number of possible choices, or due to the fact that the distance metric used in BF-maxD and the greedy algorithms is suboptimal. All approaches clearly outperform random selection. The greedy algorithm in RestraintMaker can thus be seen as a trade-off between optimizing a metric (distance or CHV) and limiting the required computing time. It is the fastest algorithm among the ones tested and yields comparable results to the brute-force approaches.

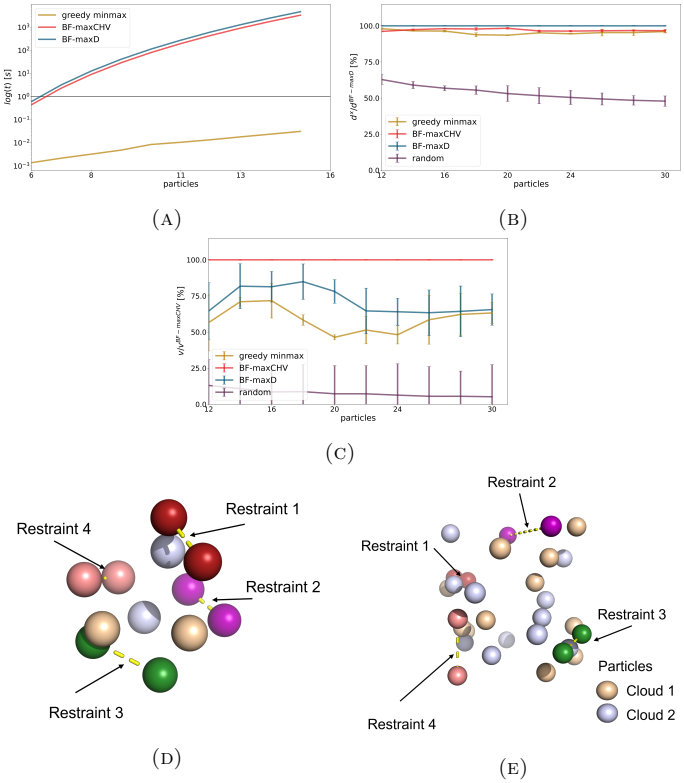


FIGURE 3.8: Comparison of algorithms to select distance restraints on toy systems. (**A**): Time complexity as a function of the number of particles in the system for the brute-force approaches BF-maxCHV (red) and BF-maxD (blue), and the greedy algorithm (yellow). (**B**): Distance metric as a function of the number of particles in the system for the brute-force approaches (red and blue), the greedy algorithm (yellow), and random selection with 100 trials (purple). (**C**): CHV as a function of the number of particles in the system. (**D** and **E**): Final distance restraints selected by the greedy algorithm for 12 (**D**) and 30 (**e**) particles. The restrained atoms are colored in green, red, pink and rose, and connected by yellow dashed lines. The two particle clouds are colored in wheat and light blue.

3.4.2 Pairwise Calculation of Relative Hydration Free Energies

To assess the quality of the selected distance restraints, the greedy algorithm in RestraintMaker was tested with a set of 16 small molecules with experimentally available hydration free energies. First, the relative hydration free energies were calculated between molecule **12** and the 15 other molecules using TI (Figure 3.6).

Free Energy Calculation

The resulting $\Delta\Delta G_{\text{hyd}}^{\text{TI,indirect}}$ agree very well with the experimental values,[243–248] with a root-mean-square error (RMSE) of 4.1 kJ/mol, a mean absolute error (MAE) of 3.1 kJ/mol (Figure 3.11, left) and a Spearman correlation coefficient r_{Spearman} of 0.87. The numerical values are reported in Table 3.2, and the corresponding $\langle \partial V(\lambda)/\partial\lambda \rangle$ curves in water and vacuum in Figures 3.9 and 3.10.

TABLE 3.2: $\Delta\Delta G_{\text{hyd}}$ for the 16 small molecules from experiment, the absolute free-energy calculations with TI taken from the ATB server [104] (TI, abs), and the pairwise relative free-energy calculations with TI and linked dual topology (TI, rel). The uncertainty estimate was calculated via Gaussian error propagation of the provided errors. The experimental uncertainty for molecule 11 was set to a default value of 2.5 kJ/mol, [248] as the uncertainty was not reported in the original source. [243] The RMSE and its uncertainty were estimated with a 100 fold bootstrap approach. The accumulated simulation time is split into preparation (pre-processing, equilibration) and production run. The data is displayed graphically in Figure 3.12.

Ligands		Experiment	$\Delta\Delta G_{\text{hyd}}^{\text{TI,direct}}$	$\Delta\Delta G_{\text{hyd}}^{\text{TI,indirect}}$
i	j	[kJ/mol]	[kJ/mol]	[kJ/mol]
1	12	18.5 ± 2.5 [244,247]	26.2 ± 0.8	17.5 ± 0.7
2	12	11.9 ± 2.7 [244,247]	14.4 ± 0.7	8.6 ± 0.6
3	12	14.8 ± 3.1 [244,247]	22.4 ± 0.7	15.2 ± 0.9
4	12	15.9 ± 3.5 [244]	15.3 ± 0.6	9.4 ± 0.4
5	12	36.1 ± 2.8 [244,247]	48.2 ± 0.4	46.9 ± 0.8
6	12	28.1 ± 3.5 [244]	36.1 ± 0.6	30.5 ± 0.5
7	12	10.6 ± 2.5 [244,246]	22.8 ± 0.5	9.5 ± 0.9
8	12	1.6 ± 3.5 [244,248]	5.2 ± 0.7	3.0 ± 1.2
9	12	27.5 ± 3.5 [244]	30.2 ± 0.6	25.7 ± 0.5
10	12	8.0 ± 3.5 [244]	6.5 ± 0.6	5.9 ± 0.5
11	12	20.4 ± 3.5 [243,244]	17.0 ± 0.6	14.8 ± 0.39
12	13	16.8 ± 3.5 [244]	18.6 ± 0.6	16.6 ± 0.5
12	14	15.9 ± 3.5 [244]	24.9 ± 0.6	20.5 ± 0.5
12	15	6.2 ± 2.6 [244,245]	14.9 ± 0.7	10.3 ± 0.4
12	16	25.5 ± 3.5 [244]	26.1 ± 0.6	24.0 ± 0.4
RMSE			6.7 ± 0.3	4.1 ± 0.3
MAE			5.5 ± 3.9	3.1 ± 2.7
r_{Spearman}			0.84	0.87
$t_{preparation}$				630 ns
$t_{production}$			$112 - 272\ ns$	3150 ns

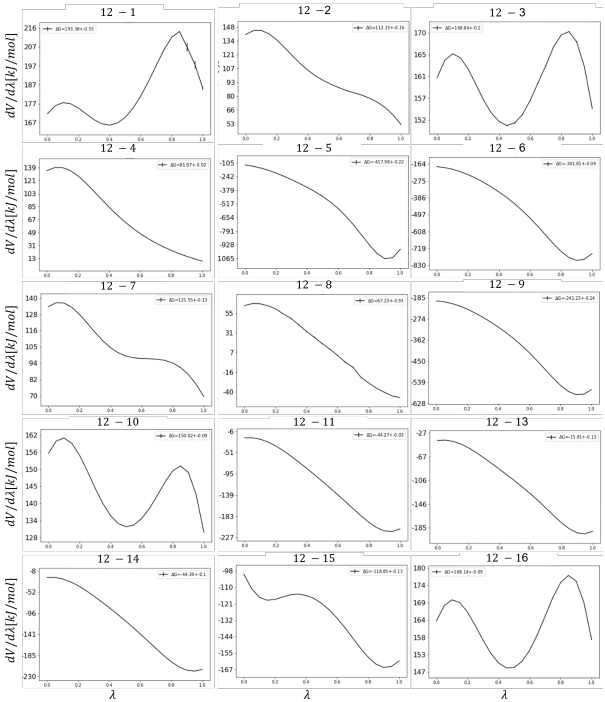


FIGURE 3.9: $\left\langle \frac{\partial V(\lambda)}{\partial \lambda} \right\rangle_{\lambda}$ as a function of λ for the 15 pairwise TI calculations in vacuum. The production run was 5 ns per λ-point.

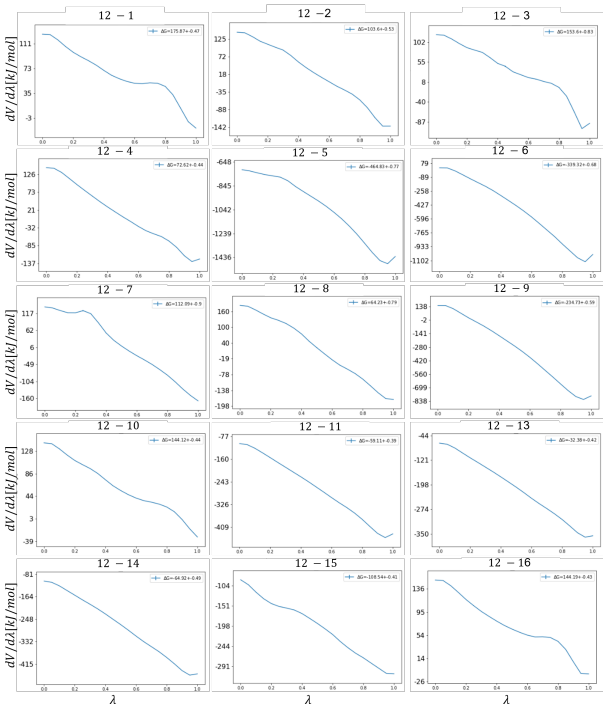


FIGURE 3.10: $\left\langle \frac{\partial V(\lambda)}{\partial \lambda} \right\rangle_{\lambda}$ as a function of λ for the 15 pairwise TI calculations in water. The production run was 5 ns per λ-point.

For comparison, $\Delta\Delta G_{\mathrm{hyd}}^{\mathrm{TI,direct}}$ values were derived from the calculated absolute hydration free energies reported on the ATB server,[104] which were carried out with TI using the same topologies. The $\Delta\Delta G_{\mathrm{hyd}}^{\mathrm{TI,direct}}$ values deviate a bit more from experiment with an RMSE of 6.7 kJ/mol, a MAE of 5.5 kJ/mol and a r_{Spearman} of 0.84 (Figure 3.11, center). Generally, the results of the direct[104] and indirect TI calculations agree well with each other (Figure 3.11, right). Note that for the molecule pair **5** - **12**, a similarly large deviation from experiment is observed in both types of TI calculations (10.7 kJ/mol and 12.1 kJ/mol, respectively), suggesting either a force-field deficiency or a problematic experimental value.

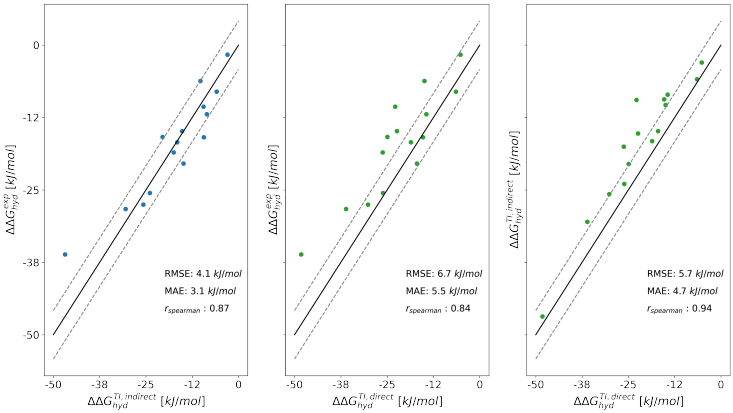


FIGURE 3.11: Comparison of the relative hydration free energies $\Delta\Delta G_{\mathrm{hyd}}$ (with molecule **12** as reference) for the 16 small molecules between experiment (exp), the pairwise relative free-energy calculations with TI and linked dual topology (TI, indirect), and the absolute free-energy calculations with TI taken from the ATB server[104] (TI, direct). The numerical values are given in Table 3.2.

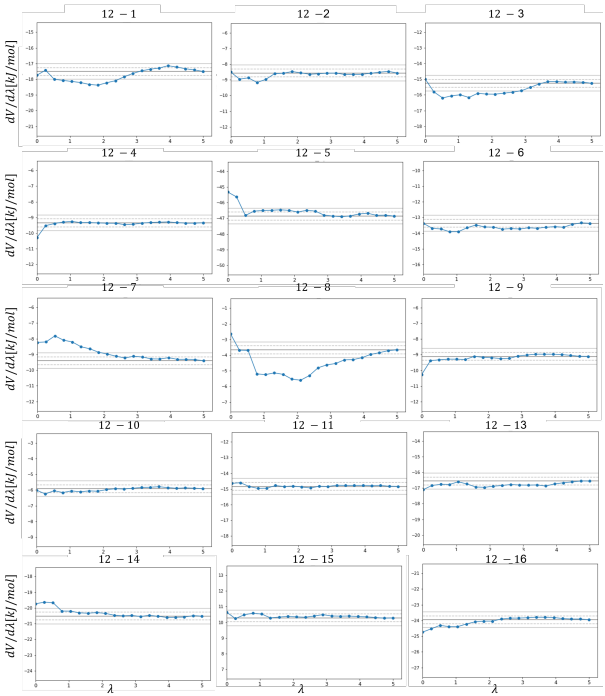


FIGURE 3.12: Convergence of $\Delta\Delta G_{\text{hyd}}$ as a function of the simulation time per λ-point for the 15 pairwise TI calculations.

SAMPLING

It is crucial for the linked dual topology approach that the applied distance restraints do not distort the conformational sampling of the molecules. For this, the relative translational motion of the two aligned molecules was analyzed for each λ-window and molecule pair by calculating the fluctuation of the distance between the COGs of the restrained atoms in the central rings of the two molecules. Figure 3.13 shows both the standard deviation and the maximum observed distance between the two COGs.

The standard deviation is close to zero for all pairs, indicating

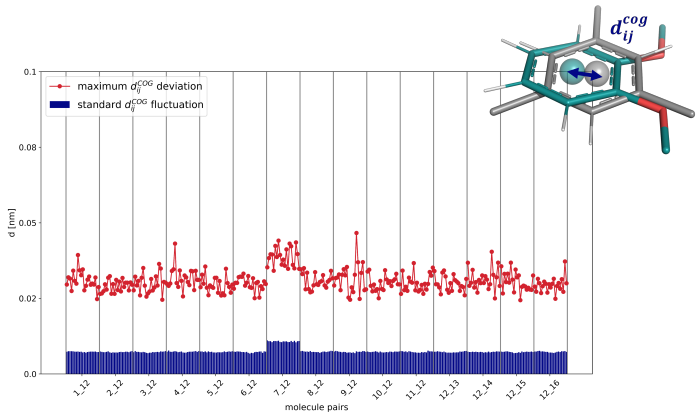


FIGURE 3.13: Standard deviation of the distance distribution (blue) and maximum distance (red) between the COGs of the central rings of the molecule pairs in the TI simulations in water. The COG was calculated for the restrained atoms in the rings. The horizontal axis shows for each molecule pair the different λ-windows between 0 and 1.

that the two cores overlap well given the chosen restraints. Maximum distances are around 0.03 nm. For the pair **7** - **12**, the distances are slightly higher, which results from the fact that molecule **7** is a bridged bicycle. The force constant of 5000 kJ/(mol·nm^2) for the distance restraints was found to be a good compromise to ensure a tight overlap of the molecules without significantly perturbing their conformations. Note that the range of reasonable force constants is rather large and only for extremely high values (i.e. 50'000 kJ/(mol·nm^2) or larger), does the restraining affect the free-energy results.

A similar analysis was carried out for the relative rotational motions of the molecules, considering the restrained atoms in the molecule pairs (Figure 3.14). In terms of the three Euler angles, a maximum relative rotation of 6.3° was observed, which

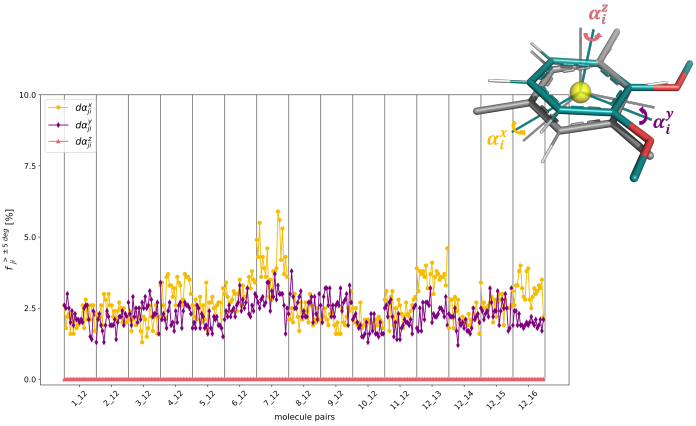


FIGURE 3.14: Fraction f of frames in the TI simulations in water, in which the relative rotation around the x-axis (yellow), y-axis (purple), and z-axis (red) of the central rings of the molecule pair exceeds 5°. The horizontal axis shows for each molecule pair the different λ-windows between 0 and 1.

is reasonably small for one dimension. The largest fluctuation was again observed for the pair **7** - **12**. The rotation around the z-axis shows significantly smaller deviations compared to the other dimensions, because the two molecules need to rotate against each other in plane. In contrast, the rotations around the x and y-axis correspond to a relative tilt of the molecules, which is easier to realize.

While most of the molecule pairs in Figure 3.6 have the same central benzene core, the transformations from molecule **12** to molecules **2** and **3** involve the change from benzene to cyclohexane or cyclohexene, respectively. To assess whether the applied distance restraints affect the conformational sampling of the aliphatic ring, the distributions of the three pseudo torsional angles (Pickett and Strauss coordinates[86]) were monitored in the simulation at

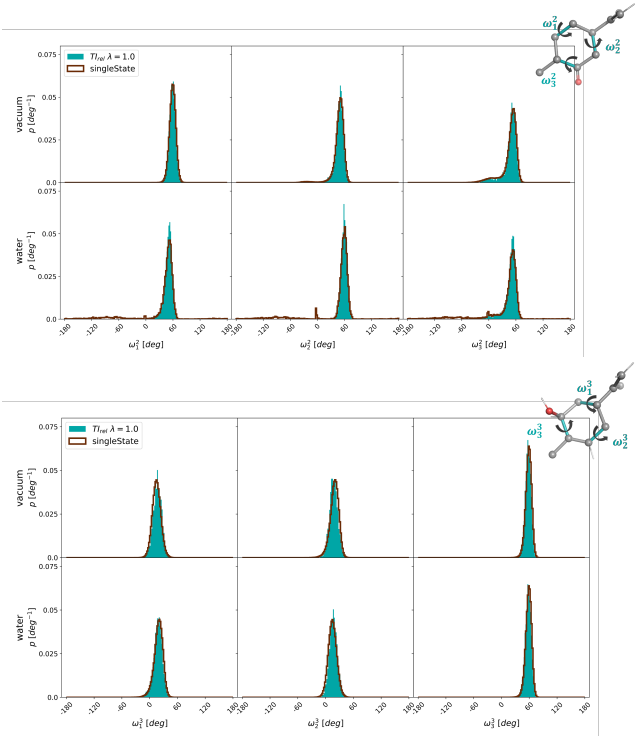


FIGURE 3.15: Comparison of the normalized torsional angle distributions of the three pseudo torsional angles of the aliphatic ring of molecules **2** (top) and **3** (bottom) in the TI calculation at $\lambda = 1.0$ (filled) and in plain MD simulations (dark red line) in vacuum (top) and in water (bottom).

$\lambda = 1.0$, and compared to plain MD simulations of molecules **2** and **3** in vacuum and in water (Figure 3.15). In both cases, the distributions showed nearly perfect overlap, indicating that the sampling is not affected by the distance restraints in the linked dual topology.

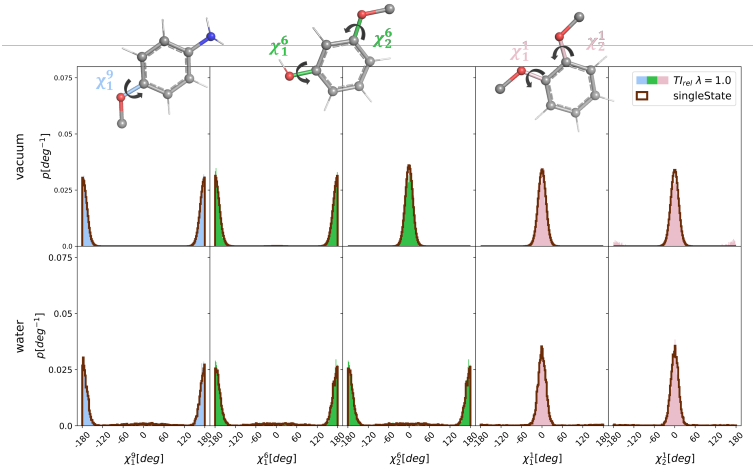


FIGURE 3.16: Comparison of the normalized torsional angle distributions of the substituents of molecules 9 (blue), 6 (green), and 1 (pink) in the simulation at $\lambda = 1.0$ (filled) and in plain MD simulations (dark red line) in vacuum (top) and in water (bottom).

A similar analysis of the torsional angle distributions was also carried out for the substituents of molecules **1**, **6** and **9** (Figure 3.16). Again, no major differences are observed between $\lambda = 1.0$ and the plain simulations.

3.5 CONCLUSION

In this chapter, we presented an efficient algorithm for the appropriate placement of distance restraints in free-energy calculations performed with the linked dual topology approach. Linked dual topologies have the advantage that larger transformations can be simulated in a straightforward manner (e.g. no soft bonds are required), while reducing the sampling complexity. With the developed RestraintMaker Python package, distance restraint sets can be created from a script or at GUI level, and written out in the GROMOS and GROMACS formats or in JSON format. The greedy algorithm is a graph-based approach and can be straightforwardly applied to molecules with (semi)rigid cores (typically aromatic or aliphatic rings). The only required user inputs are the number of restraints n_{res} to be selected and the maximum distance between the restrained atoms d_{res} .

The performance of the algorithm was evaluated using toy systems (particle clouds) and compared to two brute-force approaches. In view of the results, the greedy algorithm represents a good trade-off between computing time and accuracy.

RestraintMaker was used to select optimal distance restraints for the calculation of relative hydration free energies with both TI (pairwise) and RE-EDS (multi-state). The results of the RE-EDS approach were obtained by Salomé Rieder and are provided in the related publication. In all cases, good agreement between the different free-energy methods and with experiment was observed. Detailed analysis of the conformational sampling also indicated that the effect of the possible distortions induced by the distance restraints on the conformations is negligible. Even when restraining the benzene core and the cyclohexane core of two molecules

together, accurate free-energy differences were obtained and the distributions of the pseudo torsional angles of the cyclohexane ring were nearly identical with those from plain MD simulations. The results with RE-EDS highlighted the superior sampling efficiency of the method, which will be further discussed in Chapter 4.

4 Relative Free-Energy Calculations for Scaffold Hopping-Type Transformations with an Automated RE-EDS Sampling Procedure *

"During my undergraduate work I concluded that electrostatics is unlikely to be important [for enzymes]"

Arieh Warshel, Nobel lecture 2013

The calculation of relative free-energy differences between different compounds plays an important role in drug design to identify potent binders for a given protein target. Most rigorous methods based on molecular dynamics (MD) simulations estimate the free-energy difference between pairs of ligands. Thus, the comparison of multiple ligands requires the construction of a

* This Chapter is reproduced in part from Benjamin Ries, Karl Normak, R. Gregor Weiß , Salomé Rieder, Emília P. de Barros, Candide Champion, Gerhard König, Sereina Riniker, J. Comput.-Aided Mol. Des., in press (2021), licensed by Creative Commons CC BY.

"state graph", in which the compounds are connected by alchemical transformations. The computational cost can be optimized by reducing the state graph to a minimal set of transformations. However, this may require individual adaptation of the sampling strategy if a transformation process does not converge in a given simulation time. In contrast, path-free methods like replica-exchange enveloping distribution sampling (RE-EDS) allow the sampling of multiple states within a single simulation without the pre-definition of alchemical transition paths. To optimize sampling and convergence, a set of RE-EDS parameters needs to be estimated in a pre-processing step. Here, we present an automated procedure for this step that determines all required parameters, improving the robustness and ease of use of the methodology. To illustrate the performance, the relative binding free energies are calculated for a series of checkpoint kinase 1 (CHK1) inhibitors containing challenging transformations in ring size, opening/closing, and extension, which reflect changes observed in scaffold hopping. The simulation of such transformations with RE-EDS can be conducted with conventional force fields and, in particular, omit the need for soft bond-stretching terms.

4.1 INTRODUCTION

Rigorous free-energy calculations using MD simulations have become an important tool to estimate binding free energies of novel compounds for lead optimization in drug discovery.[35,36,135] Although computationally relatively expensive, these methods are needed to properly account for entropic contributions introduced by protein/ligand conformational changes, entropy-enthalpy compensation, and the desolvation of a ligand.[254]

Computational free energy calculations typically make use of thermodynamic cycles, i.e., the transitive difference relations of idealized states of the system of interest that are representable by a graph. For instance, to estimate the binding free energy of five compounds, a "state graph" can be constructed (Figure 4.1), where the nodes represent the end states and the edges the free-energy differences between them. Although not impossible,[31] the direct calculation of (absolute) binding free-energies (ΔG_i^{bind}) is generally very challenging to achieve computationally.[35] A simpler alternative is to calculate the alchemical free-energy differences between two compounds i and j in a given environment (ΔG_{ji}^{env}) and then compare the relative binding free energy $\Delta\Delta G_{ji}^{\text{bind}}$ with the difference of the ΔG_i^{bind} obtained from experiments,[255,256]

$$\Delta\Delta G_{ji}^{\text{bind}} = \Delta G_{ji}^{\text{protein}} - \Delta G_{ji}^{\text{water}} = \Delta G_j^{\text{bind}} - \Delta G_i^{\text{bind}} \quad (4.1)$$

Conventional free-energy methods such as TI[140] and FEP[141] introduce a coupling parameter λ to define a pathway from end state i ($\lambda = 0$) to end state j ($\lambda = 1$). In practice, simulations at discrete intermediate λ-points are performed to obtain converged free-energy differences.

If a (large) series of N compounds is investigated, the free-

energy difference for all $(N(N-1))/2$ pairs of ligands would in principle have to be calculated. To reduce the computational cost, automatic schemes have been developed to identify the edges in the state graph (Figure 4.1) with the smallest perturbations such that all nodes (for a given environment) are connected.[201,218,257] It is thereby important to include some cycles as cycle closure is a frequently used measure to assess convergence. Nevertheless, manual optimizations may sometimes be required to determine the best sampling strategy.[195] Furthermore, calculating only a subset of the edges leads to a larger uncertainty in the estimated free-energy difference for pairs that are no longer directly connected. As $\Delta\Delta G_{ji}^{\text{bind}}$ values are often relatively small, the increased uncertainty may negatively impact the usefulness of such calculations in practical applications.

An attractive and more efficient alternative to path-dependent methods is to simulate a reference state, which includes all N end states simultaneously, without the specification of pathways (green rings in Figure 4.1). Such a reference state is provided by the EDS[136,148,169,200] method. The EDS reference state can be further tuned for optimal sampling with parameters. Note that cycle closure is guaranteed by definition in this approach. In order to enhance sampling further, combinations of EDS with enhanced sampling methods were developed such as replica-exchange EDS (RE-EDS)[170,191,258] and accelerated EDS.[259,260]

In this chapter, we present an improved automated workflow for RE-EDS simulations that was restructured into two phases. The first phase aims to automatically estimate method parameters that otherwise had to be provided by the user. The second phase automatically optimizes the estimates from the first phase to retrieve a robust parameter set. The final production phase calculates the relative binding free energies of multiple ligands

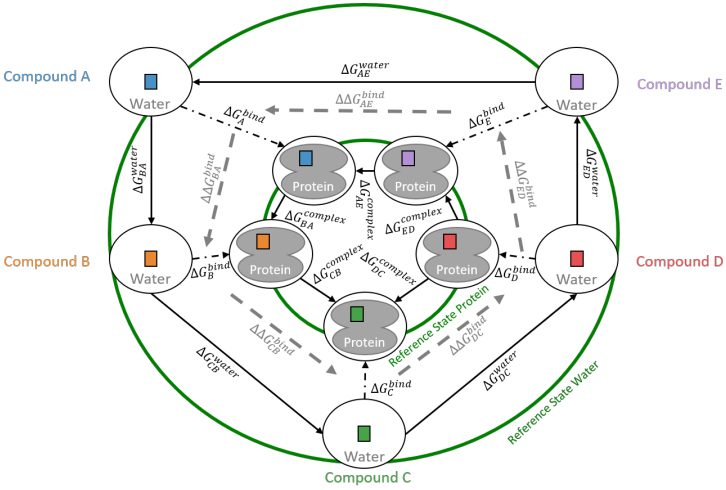


FIGURE 4.1: State graph to calculate relative binding free energies, where the nodes represent specific compounds A - E in a particular environment (water/protein). The connecting (directed) edges describe the transformations from one end state to another. The dashed-dotted arrows denote the direct calculation of the (absolute) binding free energy of compound i to the protein, ΔG_i^{bind}, whereas solid arrows indicate alchemical transformations between compound i to compound j in a given environment. From the resulting $\Delta G_{ji}^{\text{env}}$, $\Delta\Delta G_{ji}^{\text{bind}}$ can be calculated and compared with the value obtained from the difference of the experimentally determined ΔG_i^{bind} (gray dashed arrows). In pathway-dependent methods, each edge between two end states is calculated separately. With (RE-)EDS, all end states in a given environment can be considered simultaneously in a single simulation of a reference state (green circles).

from a single simulation per environment. The robustness and versatility of the RE-EDS workflow are demonstrated on a series of five inhibitors of human checkpoint kinase 1 (CHK1).[261] These ligands were selected by Wang *et al.*[202] as a challenging benchmarking set for FEP calculations since the changes between these ligands exemplify different types of core-hopping transformations

(i.e. ring size change, ring opening/closing, and ring extension). Special soft bond-stretching terms were developed to be able to handle these transformations.[202] In contrast to many other methods, no such special soft bonds are required with RE-EDS as we can use the linked dual topology approach[200] in a straightforward manner.

4.2 THEORY

4.2.1 Enveloping Distribution Sampling (EDS)

In EDS, free-energy differences between multiple end states are obtained by sampling a reference-state Hamiltonian, i.e. without the definition of specific alchemical paths.[148,169,200] Given N end states, the potential energy function V of the EDS reference state R is defined as,

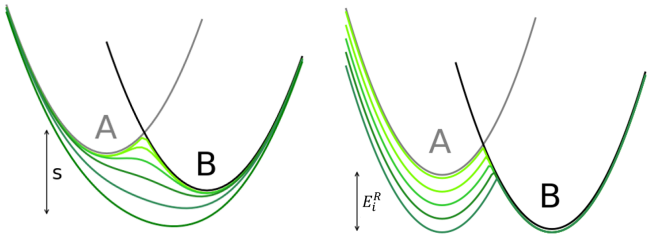
$$V_R(\mathbf{r}; s, \mathbf{E}^R) = -\frac{1}{\beta s} \ln \left[\sum_{i=1}^{N} e^{-\beta s \left(V_i(\mathbf{r}) - E_i^R \right)} \right], \tag{4.2}$$

where $\beta = (k_\mathrm{B} T)^{-1}$ with k_B being the Boltzmann constant and T the absolute temperature. The smoothing parameter s and the energy offsets $\mathbf{E}^R$ were introduced to enable tuning of the reference state for optimal sampling of all end states.[148,169]

A smoothness parameter set to $s = 1.0$ gives a reference potential-energy landscape that contains all the relevant minima of the end states. However, these might be separated by high barriers. For $s < 1$, the energy barriers between different end states V_i are smoothed in the reference potential V_R, increasing the transition rates between the different minima (Figure 4.2A).[169]

However, if s is chosen too small, V_R consists of a global unphysical minimum, which does not correspond to any of the end states. In the limit of $s \to 0$, all end states contribute equally to the potential-energy function of the reference state,[150] which can lead to unphysical deformations. The situation with a too small s has been termed "undersampling".[200]

The energy offsets $\mathbf{E}^R$ are used to ensure equal weighting of all end states V_i in V_R (Figure 4.2B). Note that the optimal values of s and $\mathbf{E}^R$ are not independent of each other (as can be seen in Eq. (4.2)).[169] Different schemes have been proposed to determine optimal reference-state parameters,[136,200,262] however, these are only applicable to systems with two end states.



(A) Effect of s on V_R (B) Effect of E_i^R on V_R

FIGURE 4.2: Schematic illustration of the effect of the two types of EDS reference-state parameters. (**A**): The smoothing parameter s decreases the barriers between the end states. If s is too small, an "undersampling" situation occurs with a global unphysical minimum. (**B**): The energy offsets $\mathbf{E}^R$ provide equal weighting to all end states in the EDS reference state. The figure was generated with Ensembler[263] (Chapter 2).

The force on a particle k in the EDS reference state is calculated

as,[169]

$$\mathbf{f}_k(t) = -\frac{\partial V_R(\mathbf{r}; s, \mathbf{E}^R)}{\partial \mathbf{r}_k} = \sum_{i=1}^{N} \frac{e^{-\beta s(V_i(\mathbf{r})-E_i^R)}}{\sum_{j=1}^{N} e^{-\beta s(V_j(\mathbf{r})-E_j^R)}} \left(-\frac{\partial V_i(\mathbf{r})}{\partial \mathbf{r}_k}\right). \tag{4.3}$$

For s values close to one, the reference-state forces are dominated by the one end state, for which the current coordinates are most favourable, while the other end states give high energies and therefore contribute little (i.e. "dummy states"). For small s values (undersampling situation), all end states contribute effectively to the forces, resulting in the global unphysical minimum.

The free-energy difference between two end states A and B can be calculated by employing the Zwanzig equation twice forming a path via the reference state R,[141,148,169]

$$\begin{aligned}\Delta G_{\mathrm{BA}} &= \Delta G_{\mathrm{BR}} + \Delta G_{\mathrm{RA}} \\ &= -\frac{1}{\beta}\left(\ln\langle e^{-\beta(V_B-V_R)}\rangle_R - \ln\langle e^{-\beta(V_A-V_R)}\rangle_R\right) \qquad (4.4) \\ &= -\frac{1}{\beta}\ln\frac{\langle e^{-\beta(V_B-V_R)}\rangle_R}{\langle e^{-\beta(V_A-V_R)}\rangle_R}. \qquad (4.5)\end{aligned}$$

4.2.2 Replica-Exchange EDS (RE-EDS)

The recently introduced RE-EDS method[170,191] is a type of Hamiltonian replica exchange[184,242] with the smoothness parameter s as the exchange dimension ($1 \geq s > 0$), which was inspired from constant pH simulations by Lee *et al.*[258,264] The approach is shown schematically in Figure 4.3. RE-EDS does not require a single (optimal) s-value. Instead enhanced sampling is achieved by exchanging between the replicas with different smoothness

levels. This simplifies the parameter choice problem and thus, the method can be applied to systems with more than two end states.[170,191]

For the pairwise exchanges between neighboring replicas k and l, a Metropolis-Hastings criterion[119] is used,[170,184]

$$p_{k,l} = min\Bigg(1, \exp\Big[-\beta\big((H_R(\mathbf{r}_k; s_l) + H_R(\mathbf{r}_l; s_k)) - (H_R(\mathbf{r}_l; s_l) + H_R(\mathbf{r}_k; s_k))\big)\Big]\Bigg), \tag{4.6}$$

where H_{R_k} and H_{R_l} are the reference-state Hamiltonians of the respective replicas, $\mathbf{r}_k$ and $\mathbf{r}_l$ are the current coordinates of the replicas.

Replicas are placed between $s = 1.0$ and a lower bound of s, where the reference state is in undersampling. The replicas with low s values facilitate the transitions between the low-energy regions of the different end states. Especially for systems with slowly adapting environments (e.g. protein binding pockets), regions in s-space with very low acceptance probability can occur. Thus, to ensure sufficient exchanges between all pairs of replicas, a local variant of the round-trip time optimization algorithm[265,266] was developed to optimally place the replicas in s-space.[191] It was found that a single set of energy offsets can be used for all replicas.[170] However, it is important that these energy offsets are chosen well to avoid "leakage" effects, resulting in one or more end states not being properly sampled.[170] The final free-energy differences are estimated from the replica at $s = 1.0$, which represents the physical minima of the end states.

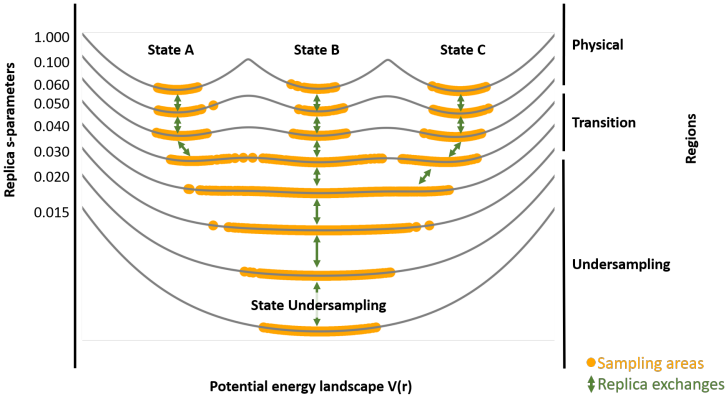


FIGURE 4.3: Schematic illustration of RE-EDS with three harmonic oscillators as end states (A, B, and C). Each replica differs by the s-parameter, generating reference states with a different degree of smoothness. Sampling of each replica is denoted with orange dots. Exchanges between the replicas are indicated with green arrows. The replica graph shows three regions: a "physical" region where s is close to 1, a transition region, and the "undersampling" region when s approaches zero. The figure was generated with Ensembler[263] (Chapter 2).

4.2.3 AUTOMATIC PARAMETER OPTIMIZATION

To facilitate the determination of the energy offsets and s-parameter distribution, we have extended and further automatized the previous[191] RE-EDS workflow (Figure 4.4).

The initial input for a system with N end states consists of a prepared EDS system (i.e. topology, perturbation topology, initial coordinates, and distance restraints), a list of energy offsets of length N with $E_i^R = 0;\ \forall\ i \in [1, ..., N]$, and a list of s-parameters, which are logarithmically distributed in the range $s_i \in [1, 10^{-5}]$. Typically, we use 21 initial s values.

The parameter exploration consists of three substeps: (i) deter-

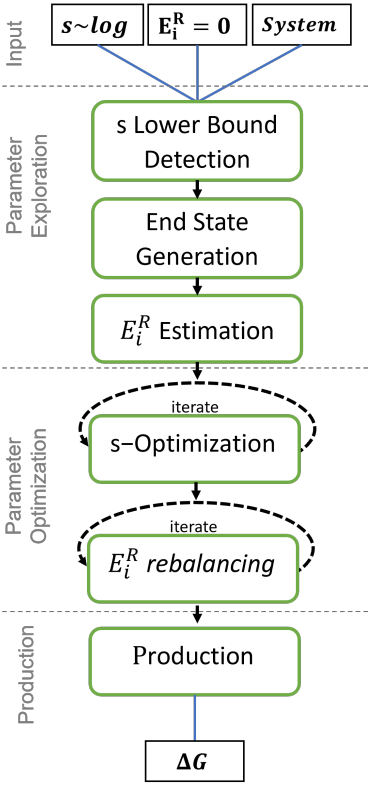


FIGURE 4.4: The RE-EDS workflow can be split into four steps: (1) Input stage with energy offsets set to $E_i^R = 0$ and a set of s-parameters logarithmically distributed between 1 and 10^{-5}; (2) Parameter exploration to determine the lower bound for s, to obtain equilibrated coordinates for each end state, and to estimate initial energy offsets with the PEOE scheme;[170] (3) Parameter optimization to improve the s-distribution with the N-LRTO algorithm[191] and the state sampling with energy offset rebalancing; (4) Production run and calculation of the free-energy differences.

mining the lower bound for the s-distribution (newly introduced), (ii) obtaining optimized coordinates within the EDS set-up for each end state (newly introduced), and (iii) estimation of an initial set of energy offsets (as done previously in Ref. 170).

To enable sampling of all end states at $s = 1.0$, some replicas have to be in undersampling to facilitate transitions. However, for efficiency reasons (and numerical stability) the number of replicas M in undersampling should be small and the lowest s-value should be as high as possible. From a short simulation with the initial s-distribution between $[1, 10^{-5}]$, the highest smoothing parameter $s_{M_{\mathrm{us}}}$ at which undersampling still occurs is determined and used in the following as a lower bound for the s-distribution. The s-distribution for the next step is then defined by logarithmically distributed replicas between $s = 1.0$ and the automatically determined lower bound.

Optimized coordinates for each end state in the EDS setup can be automatically obtained by short parallel simulations, where one end state in turn is favoured by setting an arbitrarily large energy offset for this state. The optimized coordinates allow the user to start RE-EDS simulations from different end states and are needed for the subsequent parameter optimization.

In the last substep, $E_{\mathrm{i}}^{\mathrm{R}}$ estimation, the previously developed parallel energy offset estimation (PEOE)[170] scheme is used to estimate the initial set of energy offsets. This is done based on a short simulation with the initial parameters. For each replica k in the undersampling region, the energy offsets are extracted using,[170]

$$E_i^R(new) = -\frac{1}{\beta} \ln \left\langle e^{-\beta\left(V_i(\mathbf{r}) - V_R(\mathbf{r};s_k,\mathbf{E}^R(old))\right)} \right\rangle_{R(s_k,\mathbf{E}^R(old))}. \tag{4.7}$$

The energy offsets that were extracted in parallel for the k replicas are subsequently averaged and used as initial set of energy offsets. These energy offsets should provide a first solution that is close to the optimal choice of energy offsets, which leads to an optimal state sampling of all end states in the RE-EDS simulation. As the initial energy offsets are obtained from the replicas in undersampling, they may not be exactly optimal and require fine-tuning in the next phase.

In the second step of the RE-EDS workflow, first the s-distribution is optimized and subsequently the energy offsets are fine tuned. The s-distribution is improved by minimizing the round-trip time τ and increasing the number of round-trips, using the multistate local round-trip time optimization (N-LRTO) algorithm.[191] The optimization is performed in an iterative manner with short simulations. This step is required as exchange bottlenecks between two replicas might occur leading to a very slow round trip time or to no round trips at all. In the N-LRTO algorithm, new replicas are inserted in each iteration by linear interpolation in the s-regions with exchange bottlenecks, while the replica positions of the previous iteration are retained. Adding replicas theoretically increases the round-trip time because of a longer path between the top and bottom replicas. However, the addition of intermediate replicas also increases the exchange probability between neighboring replicas, thus reducing the round-trip time. With the optimization algorithm, we aim to determine the balance between the length of the replica path and the likelihood of exchange between replicas for minimal round-trip time. The exchange bottlenecks are identified for each end state separately (i.e. multistate). The number of replicas added can be chosen by the user. The iteration is stopped when the average round-trip time $\overline{\tau}$ converges. The N-LRTO variant is needed for systems for

which severe bottlenecks are observed with the initial logarithmic s-distribution (e.g. protein binding pockets). For systems with smaller perturbations, the global multistate variant (N-GRTO)[191] can be more efficient as this algorithm re-distributes the replicas in s-space according to the exchange statistics. In this study, we started with the same number of replicas as used for the PEOE scheme above and added four replica positions per iteration in the N-LRTO algorithm.

After optimizing the number of round trips and τ, the distribution of the state sampling is improved. To reach the ideal situation that each end state is sampled to an equal amount, the initial energy offsets need to be fine tuned, while keeping the round trips approximately constant. For this, we introduce here the energy offset rebalancing scheme. To avoid overshooting, a correction factor is calculated and applied iteratively,

$$\Delta E_i^{corr} = -\frac{1}{\beta} \ln \left(\frac{f_i^{\mathrm{mc}} + c}{f_i^{\mathrm{mc,ideal}} + c} \right), \tag{4.8}$$

where f_i^{mc} is the current sampling fraction (or estimated probability) of an end state contributing to V_R, and $f^{\mathrm{mc,ideal}}$ is the ideal sampling fraction (see Section 4.2.3). To make the approach more robust, a pseudo count c is introduced to avoid singularities with zero sampling, which is defined as,

$$c = \frac{f^{\mathrm{mc,ideal}}}{x}, \tag{4.9}$$

with the intensity factor x. The default of the pseudo count was chosen to result in a maximal correction of $\Delta E_i^{corr} = 8.43$ kJ/mol, corresponding to a minimum 30-fold reduced sampling compared to the expected optimal sampling.

After optimizing the RE-EDS parameters, the production run

is performed for a chosen length. The free-energy differences are subsequently calculated using the replica at $s = 1.0$ with Eq. (4.5).

Starting State Mixing

The sampling in RE-EDS simulations can be further improved by using starting coordinates for the replicas corresponding to the different end states (i.e. replica 1 starts in a low-energy configuration for end state 1, replica 2 in a low-energy configuration for end state 2, etc.). This technical approach is called "starting state mixing" (SSM) in the following and is also used for Hamiltonian replica-exchange TI calculations (see e.g. 267,268). The optimized coordinates obtained in the parameter exploration step can be used for SSM. We compare RE-EDS simulations with SSM and with a single set of starting coordinates (abbreviated as 1SS).

Analysis

Three types of metrics were used to quantify the sampling in RE-EDS simulations. The first metric determines for each end state i the sampling fraction where it is maximally contributing to the reference state, i.e. f_i^{mc}. A maximally contributing state is defined as the end state with the lowest potential energy minus its energy offset in a frame. As can be seen in Eq. (4.3), maximally contributing end states have the largest impact on the reference-state sampling at a given time point.

Optimal sampling in a RE-EDS system is achieved when all end states are sampled as maximally contributing states to an equal extent at $s = 1.0$, i.e.

$$f_i^{\mathrm{mc,ideal}} = \frac{1}{N} \ , \forall \ i \ \in \ \{1, ..., N\} \tag{4.10}$$

The second metric is the estimated sampling fraction of "phys-

ical occurrence" of an end state i, i.e. f_i^{occur}. As a result of phase-space overlap with the current maximal contributing end state, other end states in the EDS system might be sampled simultaneously. An end state is counted as "occurred" when its potential energy is below the threshold $V_i \leq T_i^{\text{phys}}$ at a time point t. These thresholds are estimated during the second substep of the parameter exploration phase. If end states show no phase-space overlap, f_i^{occur} will be (nearly) the same as f_i^{mc}.

Undersampling is detected with a third metric using the thresholds T_i^{us}. These thresholds are determined in the first substep of the parameter exploration phase from the simulation with the lowest s-value. If all end states have a potential energy below their respective $V_i - E_i^R \leq T_i^{\text{us}}$, the current frame is characterized as undersampling.[170]

4.3 COMPUTATIONAL DETAILS

4.3.1 Model System

To showcase the performance of RE-EDS, a system of five inhibitors (L1, L17, L19, L20 and L21) of checkpoint kinase 1 (CHK1) taken from Ref. 261 was chosen (Figure 4.5). The numbering of the compounds is according to Ref. 261. The same system was studied in Ref. 202 as part of a series of scaffold hopping systems. Although the five ligands share a common substructure, they were considered to exemplify different types of core-hopping transformations (i.e. ring size change, ring opening/closing, ring extension) and R-group modifications.[202]

For the protein, the GROMOS 54A7 force field[102] was used.

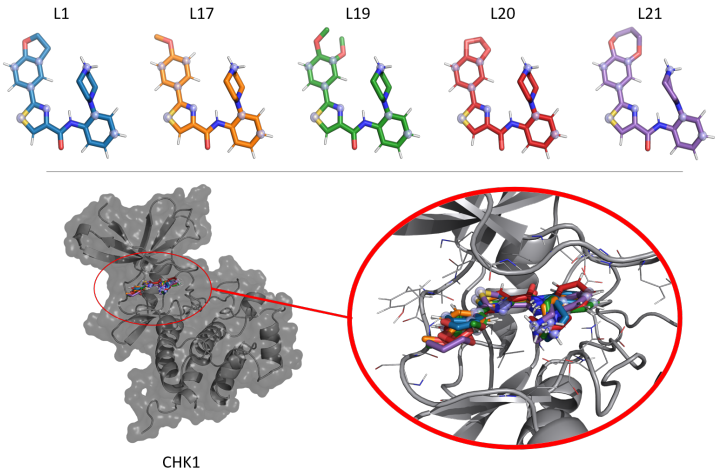


FIGURE 4.5: (Top): 3D depiction of the five CHK1 inhibitors L1, L17, L19, L20, and L21 (numbering according to Ref. 261). The selected locations of the distance restraints are indicated by the silver spheres. (Bottom): CHK1 protein in complex with the ligand bundle (PDB ID:3U9N).

For the ligands, topologies were generated using the parametrization by the ATB server[269] as an initial guess. The bonded terms were manually harmonized and adjusted to match the parameterization of similar functional groups in the GROMOS 54A7 force field. Partial charges were generated with our previous machine learning approach[270] ($\epsilon = 4$) and manually arranged into charge groups. The input files can be retrieved from: `https://github.com/rinikerlab/reeds/tree/main/examples/systems`.

4.3.2 System Preparation

The crystal structure of CHK1 in complex with ligand L1 (PDB ID:3U9N) was used as starting structure. The initial coordi-

nates for ligands L17, L19, L20, L21 were generated with the `ConstrainedEmbed()` functionality in the RDKit,[235] where the common part was kept fixed in the crystal conformation. The coordinates of each ligand and those of the protein were subsequently energy minimized in vacuum using the steepest descent[271] approach implemented in the GROMOS software package.[94]

The linked dual topology approach was used for the RE-EDS simulations, i.e. each ligand is present in the system separately.[200] Thus, each end state comprises of one active ligand and $N-1$ inactive (dummy) ligands. To avoid spatial drifting of the dummy ligands, eight distance restraints per ligand pair were defined within the common substructure (Figure 4.5) to connect all ligands in a ring with the help of the RestraintMaker program (`https://github.com/rinikerlab/restraintmaker`) (order: -L1-L17-L19-L20-L21-). The reference distance was set to 0.0 nm and the force constant to 1000 kJ mol^{-1} nm^{-2}. The combined topology file was generated with the program `prep_eds` in the GROMOS++[252] package. The EDS system was solvated in a cubic box of SPC[250] water (resulting in 1'848 solvent molecules for the ligands in water and 15'639 solvent molecules for the protein-ligands complex). An energy minimization was carried out with the steepest descent algorithm,[271] where all solute atoms were position restrained with a force constant of 25$'$000 kJ mol^{-1} nm^{-2}.

4.3.3 Simulation Details

All simulations were performed with the GROMOS software package[94] (freely available on `http://www.gromos.net`). The equilibrations and production runs were carried out under isothermal-isobaric (NPT) conditions using the leap-frog integration algorithm[123] and a time step of 2 fs. Bond lengths were constrained

with SHAKE[112] using a tolerance of 10^{-4} nm. The nonbonded contributions were calculated with a twin-range scheme using a short-range cutoff of 0.8 nm and a long-range cutoff of 1.4 nm. The electrostatic nonbonded contributions beyond the long-range cutoff were calculated with the reaction-field[88] approach and a dielectric permittivity of 66.7[272] for water.

The temperature was kept constant at 300 K using the weak coupling scheme[125] and a coupling time of 0.1 ps^{-1}. The pressure was kept at 1.031 bar (1 atm) with the same type of algorithm and a coupling time of 0.5 ps and an isothermal compressibility of $4.575 \cdot 10^{-4}$ $(\text{kJ mol}^{-1} \text{ nm}^{-3})^{-1}$. Rotation and translation of the center of mass of the simulation box were removed every 2 ps. Energies were written to file every 20 steps and coordinates every 5′000 steps. In the RE-EDS simulations, replica exchanges was attempted every 20 steps.

4.3.4 RE-EDS Workflow

The new Python code to manage the RE-EDS workflow, including the analysis steps, can be retrieved from: `https://github.com/rinikerlab/reeds`. The workflow starts with the energy-minimized coordinates of the EDS system (all N ligands plus environment, maximally contributing end state is L20) into the parameter exploration step, which is used as equilibration phase. A RE-EDS simulation of 0.2 ns length was performed with 21 logarithmically distributed replicas between $s = 1.0$ and 10^{-5} and all energy offsets set to zero. The thresholds T_i^{us} were estimated from replicas with very low s-values. Undersampling was observed when each end state occurred with a fraction $f_i^{\text{occur,us}} \geq 0.75$ during the simulation period. To be conservative, the lower bound of the s-parameters for the following steps was set to the s-value

two levels below the highest replica with undersampling.

To optimize the coordinates of the system for each end state, an EDS simulation of 2 ns length was performed for each end state i with $s = 1.0$ and $E_i^R = 500$ kJ mol^{-1} while the energy offsets of all other end states were set to -500 kJ mol^{-1}. L20 was the initial maximally contributing end state in the starting configuration. The coordinates were considered to be optimized when the desired end state was constantly sampled as the maximally contributing state in the last 30 % of the simulation.

To determine the energy offsets, a 1.5 ns RE-EDS simulation was carried out with 12 logarithmically distributed replicas for the ligands in water and 17 for the protein-ligands complex between $s = 1.0$ and the lower bound (determined above). The first 0.4 ns of the simulation were discarded as equilibration. This simulation was performed in two manners: (i) using the final coordinates from the lower-bound determination as starting configuration for all replicas (1SS approach), or (ii) using the different optimized coordinates from the previous substep for the replicas in an alternating way (SSM approach). For the PEOE[170] scheme, the following parameters were used: fraction $f_i^{\mathrm{us}} \geq 0.9$ and the potential thresholds determined in the lower bound exploration T_i^{us}.

The iterative optimization of the s-distribution with the N-LRTO[191] algorithm was started with the energy offsets and the final coordinates of the previous substep. Four replicas were added per iteration. The simulation length of the first iteration was 0.5 ns, and subsequently increased by 0.5 ns at each iteration until a maximum length of 1.5 ns was reached.

The iterative optimization of the f_i^{mc} distribution was carried out with the described scheme. The scheme used short 0.5 ns simulations, and adjusted in each step the energy offsets E^R with

a pseudo-count intensity factor $x = 30$.

The optimization was considered converged here when all end states were sampled as maximally contributing states at $s = 1.0$, the number of round trips per ns was above zero, and the improvement of the round-trip time was below $\overline{\tau}/nRT < 0.5$ ns.

The production run with constant reference-state parameters was performed for 3.5 ns.

4.3.5 Simulation of Single States

The input coordinates for the simulations of the individual end states were extracted from the RE-EDS starting coordinates and subsequently energy minimized. Next, a production run of 4 ns was performed.

4.3.6 Analysis

Free-energy differences were calculated with the program `dfmult` from the *GROMOS++*[252] package. Statistical analysis and handling of the workflow steps are based on the Python packages pandas,[181] Matplotlib,[182] NumPy,[179] SciPy,[178] and PyGromosTools.[263]

4.4 RESULTS AND DISCUSSION

The chosen model system of five inhibitors of CHK1 kinase exemplifies different core-hopping transformations (i.e. ring size change, ring opening/closing, ring extension) and R-group modifications,[202] increasing the complexity compared to the systems previously studied with RE-EDS. Furthermore, the performance

can be directly compared to the results obtained with FEP+ and OPLS3 in Ref. 202 as well as with QligFEP results in Ref. 195.

Parameter Exploration

The RE-EDS workflow was started by estimating the lower bound for the s-distribution. Using the above mentioned undersampling criterion (see Methods section), a lower bound of $s = 0.01$ was determined for the protein-ligands complex and $s = 0.0056$ for the ligands in water.

A fast transition of the initial maximally contributing end state to the desired maximally contributing end state was observed in the End State Generation. This process was monitored by the maximally contributing end state metric over time. The transition occurred latest after 1.3 ns, and the system remained in the biased end state for the rest of the simulation time. In both water and complex simulations, the desired end state was sampled about 99% of the simulation time with the exception of L19 in water (Table 4.1). Optimized coordinates were obtained for all five ligands, as verified by comparing the potential-energy distribution from the EDS simulation with the one extracted from a standard MD simulation of the respective ligand (Figure 4.8). From these same steps, the potential-energy thresholds for the occurrence sampling (T_i^{phys}) and undersampling (T_i^{us}) were estimated (Table 4.2).

TABLE 4.1: Fraction of the simulation time f_i^{mc} (in %) that the desired end state was sampled as the maximally contributing state during the EDS simulation to optimize the coordinates for a desired end state.

Ligand	Water	Complex
L1	99.84	99.97
L17	99.99	99.97
L19	36.07	99.98
L20	99.99	100
L21	100	99.97

To inspect if the optimized state simulations' results sufficiently represent the target states, a comparison between the target state obtained potential-energy distributions in the EDS simulations with MD simulations consisting of only the target state was conducted (Figure 4.6).

TABLE 4.2: Potential thresholds for occurrence sampling (T_i^{phys}) and undersampling (T_i^{us}) determined during the parameter exploration (in kJ mol^{-1}).

Ligand	Water		Complex	
	T^{phys}	T^{us}	T^{phys}	T^{us}
L1	-582.96	-436.05	-737.37	-516.41
L17	-572.41	-419.16	-717.95	-492.83
L19	-579.13	-415.91	-738.95	-483.78
L20	-636.00	-492.75	-759.01	-549.35
L21	-656.22	-488.43	-805.30	-539.78

The energy offsets $\mathbf{E}^R$ were estimated from a short RE-EDS simulation with the PEOE[170] scheme and are listed in Table 4.3. For $s = 1.0$, the energy offsets should ideally be equal to the free energy of the corresponding state (i.e. $\Delta E_{ji}^R = \Delta G_{ji}$) such that the partition function of the reference state is the sum of the partition functions of the end states.[169] Therefore, the comparison between the relative estimated energy offsets in water

and in complex ($\Delta\Delta E^R_{ji} = \Delta E^R_{ji,\text{complex}} - \Delta E^R_{ji,\text{water}}$) and the relative binding free energy $\Delta\Delta G^{\text{bind}}_{ji}$ can be used to (roughly) assess the quality of the estimated energy offsets. As shown in Figure 4.6, the energy offsets estimated from the SSM simulations are in better agreement with the experimental relative binding free energies than those estimated from the 1SS simulations. The relative energy offsets $\Delta\Delta E^R_{ji}$ are compared with the experimental relative binding free energies $\Delta\Delta G^{\text{bind}}_{ji}$ in Figure 4.7. The RMSE between $\Delta\Delta E^R_{ji}$ obtained with RE-EDS 1SS and $\Delta\Delta G^{\text{bind}}_{ji}$ is 12.6 kJ mol^{-1}. Outliers are mainly related to L19. With the RE-EDS SSM approach, the RMSE was reduced to 7.0 kJ mol^{-1}. No clear outliers were observed in this case. Thus, the use of the SSM approach is recommended for RE-EDS simulations.

TABLE 4.3: Energy offsets $\mathbf{E^R}$ estimated from a short RE-EDS simulation using the PEOE[170] scheme. The errors indicate the standard deviation over the different replicas in undersampling. All energy offsets were calculated relative to ligand L1. The starting coordinates were selected following the 1SS or the SSM approach (see Theory 4.2 and Methods 4.3.1 sections).

Ligand	Water		Complex	
	RE-EDS 1SS [kJ mol^{-1}]	RE-EDS SSM [kJ mol^{-1}]	RE-EDS 1SS [kJ mol^{-1}]	RE-EDS SSM [kJ mol^{-1}]
L1	0.0	0.0	0.0	0.0
L17	11.07 ± 7.61	17.81 ± 0.69	20.03 ± 5.04	18.19 ± 3.43
L19	-9.38 ± 6.85	-12.37 ± 5.23	-2.09 ± 1.56	2.4 ± 1.56
L20	-53.15 ± 2.95	-56.01 ± 13.67	-58.73 ± 4.87	-52.2 ± 2.6
L21	-76.75 ± 5.79	-69.15 ± 3.74	-77.29 ± 3.12	-77.9 ± 3.4

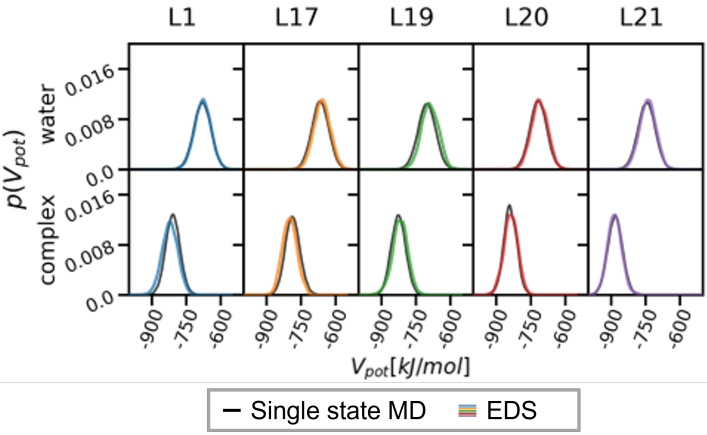


FIGURE 4.6: Comparison of the potential-energy distribution obtained from a standard MD simulation of a given end state (black) and from an EDS simulation with the given end state favoured (colored) from the first step of the RE-EDS workflow.

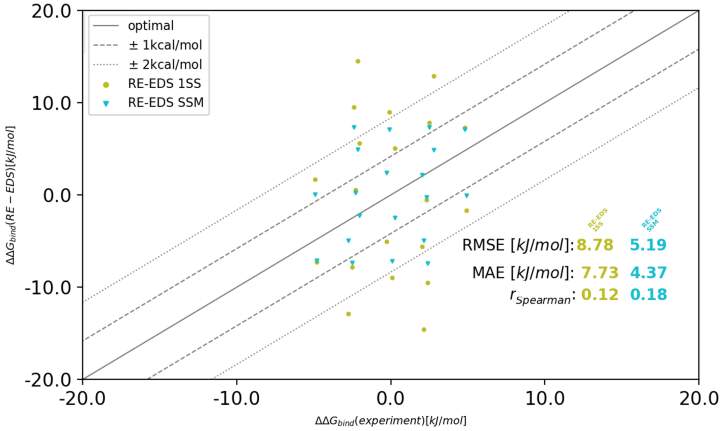


FIGURE 4.7: Comparison of the relative energy offsets $\Delta\Delta E_{ji}^{R}$ in water and complex with the experimental relative binding free energies $\Delta\Delta G_{ji}^{\text{bind}}$. The energy offsets were estimated from RE-EDS simulations using the 1SS (green) or SSM (blue) approach to select the starting configurations of the replicas.

4.4.1 Parameter Optimization

The optimization of the s-distribution was performed with the N-LRTO[191] algorithm, thereby minimizing the average round-trip time $\overline{\tau}$ in the replica graph. For the 1SS complex system, four optimization iterations were used. For the other systems, three iterations were used.

In the first iteration, the total number of observed round trips was very low or zero for all approaches. In the following iterations, this quantity increased, and the average round-trip time decreased for all simulations (Figure 4.9). The number of round trips was generally smaller in the complex than in water due to a more pronounced gap region.[191] Already after the second iteration, the round-trip time was reduced in all approaches. The improvement of the $\overline{\tau}$ over the iterations can also be seen in Figure 4.8.

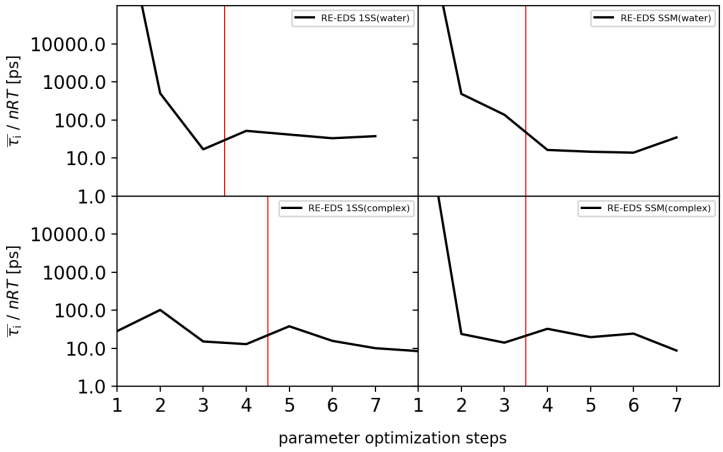


Figure 4.8: Average round-trip time as a function of the optimization steps i ($\overline{\tau}_i$) on a logarithmic scale. The red line indicates the switch from s-optimization to energy offset rebalancing.

As can be seen in the third row of Figure 4.9, the optimization algorithm increases the density of the replicas around $s = 0.041$, where the major gap region lies.

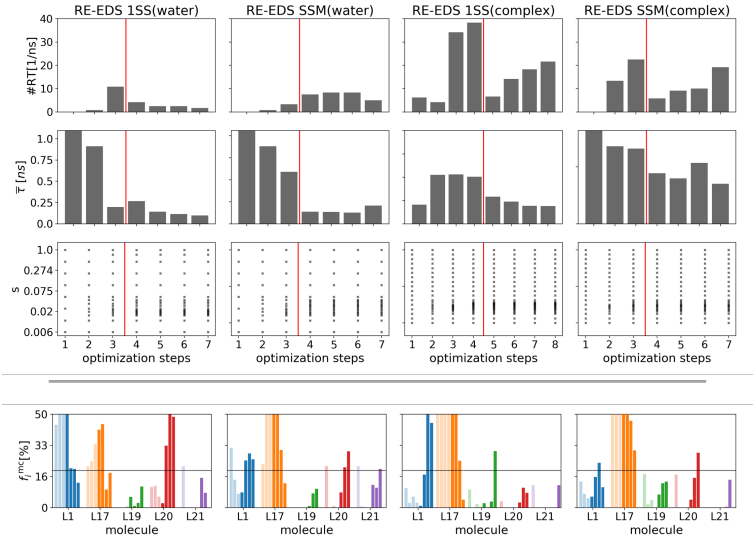


FIGURE 4.9: Optimization steps of the s-distribution with the N-LRTO[191] algorithm followed by the energy offset rebalancing scheme (start indicated by the red horizontal line). The measured quality criteria were the number of round trips (1. row), the average round-trip time $\overline{\tau}$ (2. row), the placement of the replicas in s-space (3. row), and the sampling fractions of maximally contributing states f_i^{mc} (4. row). The light colored bars of f_i^{mc} indicate s-optimization iterations, whereas the fully colored bars indicate energy offset rebalancing steps.

The s-optimization was stopped after a sufficiently high number of round trips and low round-trip time was reached. This resulted in 20 replicas for the ligands in water after three s-optimization iterations. For the protein-ligands complex, the fourth s-optimization iteration was chosen for the 1SS approach, and the third iteration for the SSM approach, resulting in 29

and 25 replicas, respectively. The average round-trip time after convergence was $\overline{\tau} = 0.4 \pm 0.2$ ns for all simulations.

After the s-optimization, the energy offset rebalancing scheme was applied to improve the state sampling.

During the rebalancing steps, no further replicas were added to the s-distribution. It is essential for the success of the rebalancing scheme that round trips occur. Therefore, the number of round trips and average round-trip time were monitored. In all systems, the number of round trips and $\overline{\tau}$ remained relatively stable over the four rebalancing steps. For the RE-EDS 1SS approach in water, the number of round trips slightly decreased but never dropped to zero.

Across the optimization steps, also the sampling of the end states as maximally contributing states at $s = 1.0$ was monitored. During the s-optimization, some end states "vanish" and are no longer sampled as maximal contributing states. This leakage effect can occur when the initially estimated E^{R} are not exactly optimal.[170] With energy offset rebalancing, the sampling of each end state can be recovered, and the sampling distribution approaches the ideal case. After rebalancing, all end states showed a $f_i^{\mathrm{mc}} > 0$ and the mean absolute deviation of the sampling distribution from ideal decreased from $20 - 25\%$ to approximately $7 - 12\%$ (Figure 4.10).

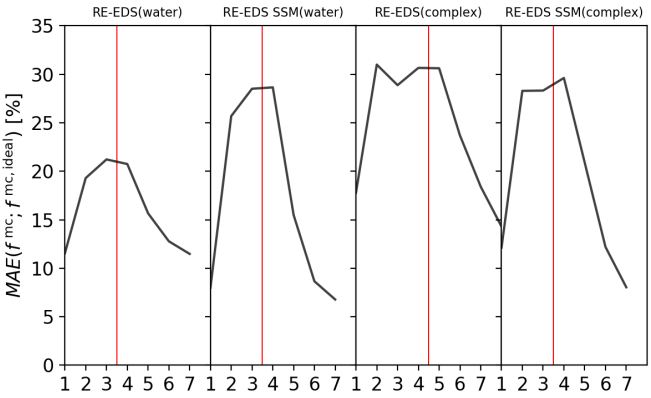


FIGURE 4.10: Mean absolute deviation (MAE, in percentage) of the observed state sampling f_i^{mc} from the ideal equal distribution $f_i^{\mathrm{mc,ideal}}$ during the short optimization simulations. The red line indicates the switch from s-optimization to energy offset rebalancing.

4.4.2 Free-Energy Calculation

After successfully optimizing the RE-EDS parameters, the production runs were performed for 3.5 ns.

The analysis of the maximally contributing end states at $s = 1.0$ shows that in water all end states were sampled close to the ideal equal distribution (Figure 4.11). Both in water and in complex, the potential-energy distributions of the end states generally match well the corresponding distributions from the standard MD simulations of the single end states (Figure 4.12). Only in the complex 1SS approach, a deviation can be seen for L17, with a slight shift to higher potential energies. This is due to insufficient sampling of L17 in this case (see below).

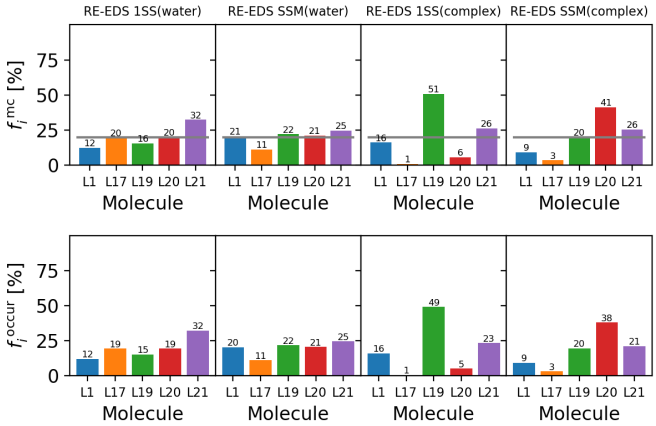


Figure 4.11: Sampling of the end states in the final production run at replica $s = 1.0$. Sampling was assessed by monitoring the maximally contributing end state (top panels) and by counting all end states a potential energy below T_i^{phys} (see Table 4.2) (bottom panels). Ideally, the sampling fraction as maximally contributing end state should be $1/N$ (Eq. (8) in the main text) for all end states, indicated as a black horizontal line.

In the simulation of the protein-ligands complex, there are still differences in sampling. Especially with the 1SS approach, L19 is generally sampled too much, while L17 is not sampled enough. The situation is improved with the SSM approach. Comparing f_i^{occur} and f_i^{mc} in Figure 4.11 indicates that the end states in the CHK1 system are clearly separated (i.e. no phase-space overlap).

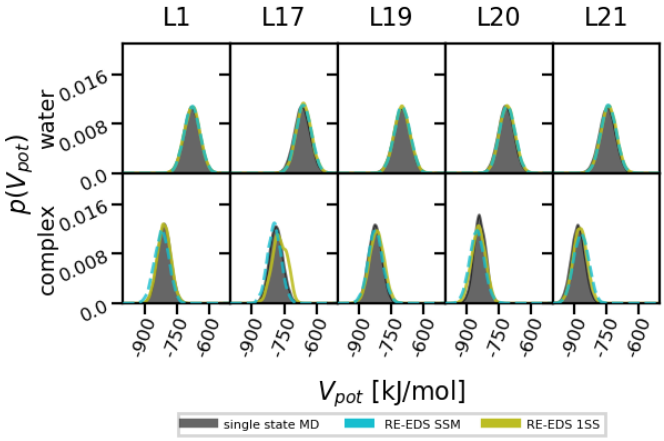


FIGURE 4.12: Comparison of the Boltzmann reweighted potential-energy distributions obtained from standard MD simulations of a given end state (black) and from the RE-EDS production runs of the 1SS (green) and SSM (turquoise, dashed) approaches.

From the replica at $s = 1.0$, the free-energy differences were calculated using Eq. (4.5) and the individual free-energy differences are given in Table 4.4. The resulting $\Delta\Delta G_{ji}^{\text{bind}}$ were compared with the experimental results taken from Ref. 261. The results are shown graphically in Figure 4.13 and numerically in Table 4.5. The RMSE with RE-EDS 1SS is 4.4 kJ mol^{-1} and the MAE is 3.9 ± 2.8 kJ mol^{-1}.

TABLE 4.4: Free-energy differences in water and in complex calculated from the production run of 3.5 ns of length with the RE-EDS 1SS and RE-EDS SSM approaches.

Ligand		RE-EDS 1SS		RE-EDS SSM	
J	I	water [kJ mol^{-1}]	complex [kJ mol^{-1}]	water [kJ mol^{-1}]	complex [kJ mol^{-1}]
L17	L1	11.9 ± 0.0	17.0 ± 0.8	12.4 ± 0.5	9.4 ± 1.9
L19	L1	2.7 ± 0.0	5.7 ± 1.0	3.1 ± 0.0	8.0 ± 0.0
L20	L1	-47.8 ± 0.0	-47.6 ± 0.9	-47.7 ± 0.0	-48.1 ± 0.0
L21	L1	-61.7 ± 0.06	-63.1 ± 0.8	-61.7 ± 0.0	-64.8 ± 0.0
L19	L17	-9.2 ± 0.0	-11.3 ± 0.6	-9.3 ± 0.5	-1.4 ± 1.9
L20	L17	-59.6 ± 0.0	-64.5 ± 0.1	-60.1 ± 0.5	-57.6 ± 1.9
L21	L17	-73.6 ± 0.0	-80.1 ± 0.1	-74.1 ± 0.5	-74.3 ± 1.9
L20	L19	-50.5 ± 0.0	-53.2 ± 0.6	-50.7 ± 0.0	-56.2 ± 0.0
L21	L19	-64.4 ± 0.0	-68.8 ± 0.6	-64.7 ± 0.0	-72.9 ± 0.0
L21	L20	-13.9 ± 0.0	-15.5 ± 0.2	-14.0 ± 0.08	-16.7 ± 0.0

The main deviations stem from ligand L17 in the RE-EDS 1SS approach, which can be explained by the insufficient sampling of L17 in the complex (see Figure 4.12 and Figure 4.11).

The performance was substantially improved using the SSM approach with RE-EDS, giving an RMSE of 3.3 kJ mol^{-1} and an MAE of 2.8 ± 1.7 kJ mol^{-1}. Only two values (L21-L11) and (L21-L19) deviate more than 4.184 kJ mol^{-1} (i.e. 1 kcal mol^{-1}) from experiment. The Spearman correlation coefficient for RE-EDS 1SS is $r_{\text{Spearman}} = 0.01$ and for RE-EDS SSM $r_{\text{Spearman}} = 0.69$.

Next, we assessed the convergence of the ΔG_{ji} values as a function of simulation time (Figure 4.14).For the RE-EDS 1SS approach, all free-energy differences appeared converged after 2.5 ns in water and after 2.7 ns in the complex. For the RE-EDS SSM approach, convergence was observed after 2.5 ns in water and after 2.9 ns in the complex.

By applying the RE-EDS methodology to the same system of five CHK1 inhibitors as studied by Wang *et. al.*[202] and later on also Jespers *et al.*,[195] a direct comparison with FEP+ and QligFEP is possible (Table 4.5). Note that the quality metrics were

calculated over all possible pairs of ligands and in both directions, not only those directly calculated by FEP+ and QligFEP. For FEP+, we obtained an RMSE of 2.4 kJ mol^{-1} and an MAE of 1.8± 1.2 kJ mol^{-1} with a Spearman correlation coefficient of $r_{\text{Spearman}} = 0.67$. Including cycle closure correction (CC)[202] reduced the RMSE to 2.1 kJ mol^{-1} and the MAE to 1.9 ± 1.0 kJ mol^{-1}. The Spearman correlation coefficient increased to $r_{\text{Spearman}} = 0.73$. Jespers *et al.*[195] reported free-energy differences with QligFEP as an average over ten independent replicas, each with significantly less simulation time per λ-window than in Ref. 202. For QligFEP, an RMSE of 2.3 kJ mol^{-1}, an MAE of 2.0 ± 1.2 kJ mol^{-1}, and a Spearman coefficient of $r_{\text{Spearman}} = 0.61$ was obtained.

Overall, the performance of RE-EDS SSM is comparable with the pairwise methods. The results with FEP+ CC and QligFEP showed a slightly higher accuracy compared to experiment, likely due to the different force fields used. The Spearman correlation coefficient is comparable with the other methods for the RE-EDS SSM approach.

In terms of computational cost, the RE-EDS approach (with 3.5 ns per replica) resulted in about a quarter of the total simulation time (in ns) than reported for the FEP+ calculations in Ref. 202 (Table 4.5). However the QligFEP approach is the approach with the lowest simulation time consumption. A major advantage of the simultaneous simulation of multiple ligands in a single RE-EDS simulation is that all $N(N-1)/2$ transformations are sampled directly, leading to low statistical errors and removing the need for a state graph. This advantage increases with increasing number of ligands. The current workflow of RE-EDS uses a relatively large amount of simulation time for parameter optimization. Future work will focus on further optimization of the workflow to reduce the pre-processing time.

TABLE 4.5: Relative binding free energies $\Delta\Delta G_{ji}^{\text{bind}}$ from experiment and calculated with the RE-EDS 1SS and RE-EDS SSM approaches. For comparison, the results for FEP+ with and without cycle closure (CC) correction taken from Ref. 202 and the results for QligFEP taken from Ref. 195 are listed. The free-energy differences of directly simulated paths were used to infer not directly simulated free-energy differences (marked in bold). If multiple indirect paths were possible, their average was used. The errors for QligFEP were determined in Ref. 195 by calculating the standard deviation over ten replicas. For FEP+, the error of the results was taken from the used BAR[168] method and the FEP+ CC errors were obtained from the cycle closure analysis. For the RE-EDS approaches, the reported error is based on the statistical uncertainties of the ΔG_{ji}^{env} values estimated using Gaussian error approximation.[169] The uncertainty estimate of the RMSE was obtained by a 100-fold bootstrapping approach.

Ligands		Exp.[261]	FEP+[202]	FEP+ CC[202]	QligFEP[195]	RE-EDS 1SS	RE-EDS SSM
i	j	[kJ mol^{-1}]	[kJ mol^{-1}]	[kJ mol^{-1}]	[kJ mol^{-1}]	[kJ mol^{-1}]	[kJ mol^{-1}]
L17	L1	0.1	-3.6 ± 0.4	-2.9 ± 1.0	-1.6 ± 1.7	5.1 ± 0.8	3.0 ± 2.0
L19	L1	-4.8	-3.9 ± 0.3	-4.0 ± 0.6	-1.7 ± 2.0	3.0 ± 1.0	-5.0 ± 0.1
L20	L1	-2.0	-2.5 ± 0.1	-3.1 ± 1.0	-1.3 ± 1.3	0.2 ± 0.9	0.5 ± 0.1
L21	L1	-2.3	**-3.4 ± 0.7**	**-3.2 ± 1.3**	**-0.1 ± 3.5**	-1.4 ± 0.8	3.2 ± 0.1
L19	L17	-4.9	-1.4 ± 0.3	-1.1 ± 1.0	**0.1 ± 2.6**	-2.1 ± 0.6	-7.9 ± 1.9
L20	L17	-2.1	0.3 ± 0.4	-0.1 ± 0.8	-1.3 ± 2.3	-4.9 ± 0.1	-2.5 ± 1.9
L21	L17	-2.4	-1.1 ± 0.4	-0.9 ± 0.9	**0.7 ± 2.6**	-6.5 ± 0.1	0.2 ± 1.9
L20	L19	2.8	**0.8 ± 0.6**	**0.1 ± 1.3**	**-0.4 ± 3.7**	-2.7 ± 0.6	5.4 ± 0.1
L21	L19	2.5	-0.1 ± 0.6	0.6 ± 0.1	0.6 ± 4.9	-4.4 ± 0.6	8.2 ± 0.1
L21	L20	-0.3	-0.3 ± 0.8	-0.6 ± 0.8	0.6 ± 1.1	-1.6 ± 0.1	-2.7 ± 0.1
RMSE			2.4 ± 0.3	2.1 ± 0.2	2.3 ± 0.38	4.8 ± 0.5	3.3 ± 0.3
MAE			1.8 ± 1.2	1.9 ± 1.0	2.0 ± 1.2	3.9 ±2.8	2.8 ± 1.7
r_{Spearman}			0.67	0.73	0.61	-0.01	0.69
$t_{simulation}[ns]$			640	640	51	171.5	157.5

From the calculated relative binding free energies, ΔG_i^{bind} can be obtained by using one experimental value as anchor point. This allows us to generate a ranking of the five ligands. To avoid any bias from the selected experimental anchor point, all possibilities were calculated and the resulting values averaged (Table 4.6). While the RMSE is generally low for all approaches ($<$ 1 kcal mol^{-1} = 4.184 kJ mol^{-1}), the ranking of the ligands as measured by r_{Spearman} is not very good. This observation is not uncommon for ligand series with small differences in binding free energy.[201,273] Note that the uncertainties of the individual values have increased

compared to the relative binding free energies due to the anchoring and averaging procedure.

TABLE 4.6: Absolute binding free energies ΔG_i^{bind} and ranking of the ligands derived from the relative binding free energies. The values were calculated from the relative binding free energies using an experimental binding free energy as anchor point, and then averaged over the five possibilities. The errors are standard deviations over the possible outcomes. For comparison, the results for FEP+ with and without cycle closure (CC) correction taken from Ref. 202 and the results for QligFEP taken from Ref. 195 are shown (calculated with the same procedure). The uncertainty estimate of the RMSE was obtained by a 100-fold bootstrapping approach.

Ligands Molecule	Exp.[261] [kJ mol^{-1}]	FEP+[202] [kJ mol^{-1}]	FEP+ CC[202] [kJ mol^{-1}]	QligFEP[195] [kJ mol^{-1}]	RE-EDS 1SS [kJ mol^{-1}]	RE-EDS SSM [kJ mol^{-1}]
L1	-40.7	-41.7 ± 1.7	-41.7 ± 0.9	-38.5 ± 1.5	-40.0 ± 3.4	-38.0 ± 2.0
L17	-40.8	-38.0 ± 1.0	-38.2 ± 1.1	-38.6 ± 1.3	-33.7 ± 1.3	-41.7 ± 2.3
L19	-35.9	-38.1 ± 0.9	-38.3 ± 1.8	-38.3 ± 1.0	-37.6 ± 3.3	-33.0 ± 2.0
L20	-38.6	-38.6 ± 1.6	-38.3 ± 1.4	-39.2 ± 1.7	-40.4 ± 3.3	-39.1 ± 2.3
L21	-38.4	-37.7 ± 1.4	-37.8 ± 1.3	-39.4 ± 1.9	-42.4 ± 2.9	-42.5 ± 1.4
RMSE		1.7 ± 0.4	1.7 ± 0.4	1.7 ± 0.4	3.8 ± 1.3	2.6 ± 0.6
MAE		1.3 ± 1.0	1.4 ± 0.9	1.4 ± 0.9	3.0 ± 2.3	2.2 ± 1.6
r_{Spearman}		0.20	0.10	-0.21	-0.40	0.30

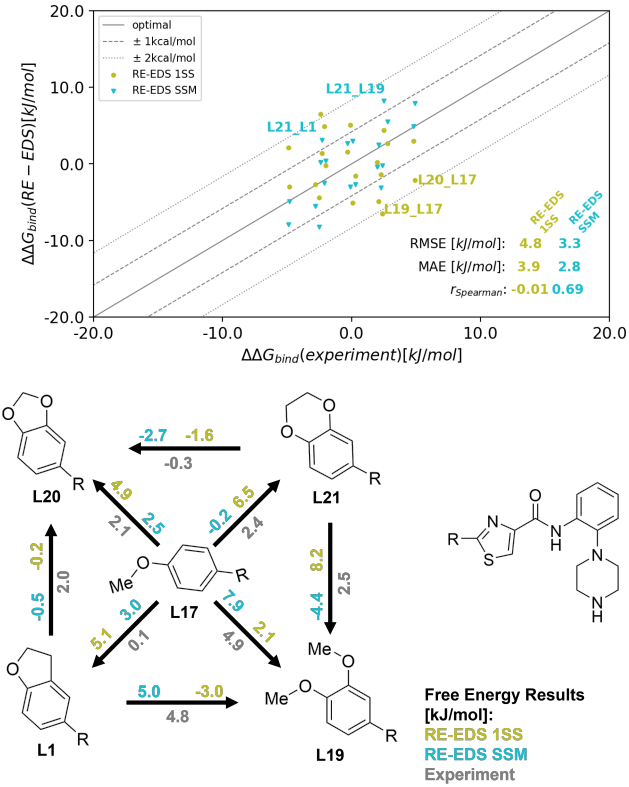


FIGURE 4.13: Free-energy differences estimated from the production run of 3.5 ns length. (Top): Comparison between the experimental and calculated $\Delta\Delta G_{ji}^{\text{bind}}$ using RE-EDS 1SS and RE-EDS SSM. The results were calculated with all possible pairwise transformations (forward and backward). (Bottom): Graphical representation of the $\Delta\Delta G_{ji}^{\text{bind}}$ results with structures, inspired by the one in Ref. 202.

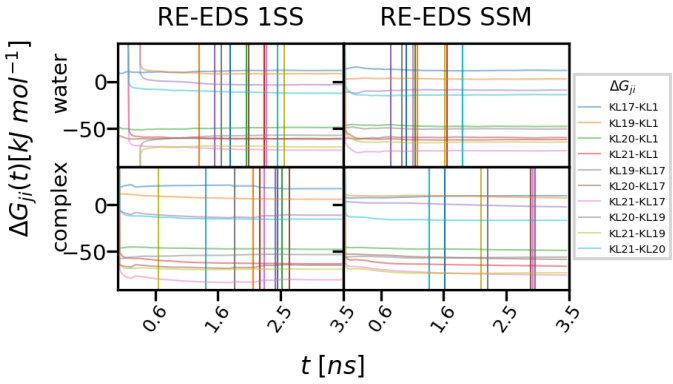


FIGURE 4.14: Convergence analysis of the RE-EDS production runs (total 3.5 ns): The free-energy results are plotted as a function of the simulation time. The vertical lines indicate when a particular ΔG_{ji} value was found to be converged (deviation below 1 kJ mol^{-1}).

4.5 CONCLUSION

This chapter reports the recent developments for the multistate free-energy method RE-EDS, which omits the definition of alchemical transition paths. The automatic workflow for RE-EDS was improved in robustness, and was applied to estimate the relative binding free energies of five CHK1 inhibitors containing typical core-hopping transformations. This system was investigated previously with FEP+ and QligFEP, allowing for a direct comparison of RE-EDS with state-of-the-art pairwise free-energy methods. Using different starting configurations representing all end states (SSM approach) in the parameter optimization of the RE-EDS workflow improved the sampling, convergence, and the accuracy of the resulting free-energy differences. The performance of RE-EDS SSM was found to be comparable with FEP+ and QligFEP, and shows that RE-EDS with a "dual topology" approach can be readily applied to challenging ligand transformations like ring size change, ring opening/closing, and ring extension.

In terms of computational efficiency, the total production run time with RE-EDS (3.5 ns per replica) was about a quarter of that reported for FEP+ with this system. As multiple ligands are simulated simultaneously in a single RE-EDS simulation, this sampling enhancement will increase with increasing number of ligands. However, the pre-processing phase in the RE-EDS workflow currently uses a relatively large amount of simulation time. Making these steps more efficient will be addressed in future work.

5

PyGromosTools: A Fast and Flexible API for the Molecular Dynamics Software Package GROMOS

"The determined Real Programmer can write FORTRAN programs in any language."

Ed Post [274]

Making data accessible and decreasing the complexity of process execution are key aspects of an application programming interface (API). Additionally, APIs reduce code duplication through code encapsulation and therefore increase code readability and re-usability. We have developed PyGromosTools to make the molecular dynamics (MD) software package GROMOS accessible in Python and facilitate its usage. PyGromosTools provides users with file classes and simulation functionality. In this chapter, we will discuss design aspects, code structure, and example usage of PyGromosTools. The code is open source and available on GitHub `https://github.com/rinikerlab/PyGromosTools`.

5.1 INTRODUCTION

Digitalization has been identified as one of the most promising tools to increase productivity in a vast range of disciplines.[275] Over the last four decades, computational techniques have also become increasingly relevant in chemistry, and many methods are now routinely used in academia and industry to e.g. predict the physicochemical properties of molecules, their 3D conformation and interactions with other molecules, or their chemical reactivity. A prominent application of computational chemistry is the highly interdisciplinary field of drug discovery, where computational approaches are employed in all stages of the development process.[30,35–37,137,276]

With the growing application of computational chemistry in industry, also the fraction of scientists that need to read or write software code increases. Typical tasks do not require writing highly optimized code in low-level languages such as Java,[277] Fortran,[278] or C++.[159] As a consequence, more advanced concepts such as memory management, parallel programming, or advanced programming paradigms are not always considered during code development, possibly leading to problems later on. In most projects, it is a balance between long-term sustainable code and so-called technical debt[279] to obtain results on the short term. This issue has led to the development of convenient-to-use scripting languages and APIs, which can be used to swiftly implement solutions to complex problems with minimal technical debt. By doing this, time is freed to focus on problems requiring specialized knowledge.[280] The scripting language with the highest impact in science nowadays is Python. Already in 2011, Python was declared the *de facto* standard language in natural sciences and

engineering, and from 2017 to today (2021), Python has remained the highest-ranked language in the *IEEE Spectrum* journal. This ranking reflects the user interests measured by the internet community Stack Overflow, data available on GitHub, and IEEE Articles.[179,281–286] Core features identified by Oliphant[287] that make Python attractive to a widespread audience are:

- Intuitive syntax that is easy to read and learn, and thus allows rapid prototyping.
- Straightforward integration with other programming languages (especially C/C++,[159] enabled by tools like pybinds,[288] Boost Python,[289] SWIG,[290] Cython,[291] or Numba[292]).
- Large community supporting high-quality packages (e.g., NumPy,[179] SciPy,[178] Matplotlib,[182] Pandas,[181] and Jupyter[173]).
- High-quality tools for environment and package management (e.g. pip[293] and conda[294]).
- Platform independence that enables development on different operating systems and computer architectures.

Despite the boom of Python, one downside of this programming language is its limited computational performance in a native form, which is related to the dynamic typing concept. To address the efficiency issues of native Python code, many different solutions have been developed. One solution is to translate the Python code during run-time (just-in-time-compiling) into C code[295] or directly into machine code.[296] Tools making this solution accessible are for example Cython[291] or Numba.[292] Prominent packages like SciPy,[179] NumPy,[178] or Pandas[181] follow a reverse approach and use Python as a wrapper to bind C or Fortran code. In this case, Python merely 'steers' the code execution and the user only interacts with the Python layer.[287] Tools like pybind11[297] or Boost

Python[289] make this approach easy to implement and allow rapid construction of Python APIs. Many packages in computational chemistry make use of this concept. Examples are RDKit,[235] PyMol,[222] PySCF,[298] pyOpenMS,[299] BioPython,[300] and Pybel.[301] The most popular MD packages follow this trend and offer Python APIs. GROMACS[93,143,160] and AMBER[98,99,302] provide for this purpose the packages gmxapi[303] and ParmEd,[304] respectively. OpenMM[95,162,305] and LAMMPS,[306,307] on the other hand, include Python APIs natively.[308] Here, we introduce a Python API for the GROMOS software package[94] called PyGromosTools.[249] This API is a starting point for further development that may ultimately allow access to the entire functionality of GROMOS as well as GROMOS++[252] from Python. Currently, PyGromosTools already provides access to the simulation trajectories and to a selection of GROMOS features, together with job queueing for high perfomance computing (HPC) clusters, and data analysis functionality. This chapter presents our rationale behind the design of PyGromosTools and shows examples of how the API can be used.

5.2 IMPLEMENTATION

PyGromosTools is a software package that builds on the long history of GROMOS. After initial development phases of the package, its structure and associated usage patterns were established with long-term stability in mind. Consequently, the implementation of PyGromosTools was preceded by identification of several design objectives that are in line with state-of-the-art API development:[309–311]

- An API is easy to learn and memorize such that solving problems with it comes naturally.
- The usage of an API should result in code that is easy to read and maintain.
- A well-designed API is hard to misuse and easy to extend.
- An API is complete and simple.

With these considerations in mind, the resulting API should enable fast, reproducible, and expandable simulation setup and execution of MD simulations. Additionally, the API should align with fundamental scientific principles to make data and code easily accessible, shareable, and reproducible. The importance of open and reliable data was already introduced and highlighted in Chapter 2.

5.2.1 Coding Style

PyGromosTools follows several coding styles. These are not enforced on the users but on the developers who would like to contribute to the API.

Code Visibility/Information Hiding

Encapsulation is an essential concept in coding of larger projects. Only those layers of the software should be presented to a user, which are required to make the API still understandable or usable. Presenting the entire code basis of larger packages to end users may be overwhelming and hamper usage. Therefore, encapsulating code into functions and classes, or alternatively managing the accessibility of certain variables/functions is vital in code

hiding.[312,313] Visible code should be easy to integrate and conveniently to use for writing more complex solutions.

In PyGromosTools, accessibility is managed with methods provided by Python. For example, private variables in the *Gromos_System* class are assigned with a prefix "_" like the attribute *_gromosPP*. Note that this way of declaring a variable "private" still leaves it easily accessible, in contrast to other languages like C++. If a variable of function should never be used externally, name mangling is employed with the prefix "__" (like *__ionDecorator*) as defined in the Python enhancement proposals (PEP) 8 (`https://www.python.org/dev/peps/pep-0008/`). The second aspect of encapsulating code into functions and classes is achieved through modularization of the code. This concept allows quick construction of more complex structures, and therefore speeds up the development process and readability at the same time.[313] One example in PyGromosTools is the file structure that is encoded by several classes, which represent the blocks, tables and fields of a GROMOS file (fields → blocks → file).

Variables, Signatures, and Classes

PyGromosTools in general follows the principle of using descriptive variables. Each name in the package should give a comprehensive description of the function of a given code section, and abbreviations are therefore discouraged. Following this style increases readability of the code, leads to improved understanding of code, and consequently reduces the number of errors. Along these lines, the second style decision for PyGromosTools is annotation of types for attributes in classes, function parameters, and function return types. This style is in accordance with PEP 526 (`https://www.python.org/dev/peps/pep-0526/`), PEP 484 (`https://www.python.org/dev/peps/pep-0484//`), and PEP 3107 (`https:`

//www.python.org/dev/peps/pep-3107/), in which a type annotation system was introduced to Python3. This decision was made due to the more complex data structures defined by PyGromosTools. Annotations provide guidance to users and developers on which data type is expected for a given function or attribute. The type annotations can be quickly accessed in the source code and also visualized in most modern IDEs or interactive Python sessions. A third style choice is the usage of keyword arguments for passing function parameters. Keyword argument passing was introduced with PEP 468 (https://www.python.org/dev/peps/pep-0468/). The underlying reason for the usage of this feature is the enhanced readability of the resulting code and an increased robustness to code refactoring of function signatures.

Documentation and Continuous Integration

In PyGromosTools, each function and module is expected to contain a docstring description. The chosen docstring style is following the numpydoc guidelines (https://numpydoc.readthedocs.io/en/latest/format.html) from the popular NumPy package.[179] The docstring style is supported in various IDEs and automatically collected in the documentation generated by sphinx.[314] As part of the documentation, PyGromosTools is accompanied by several Jupyter notebooks, which feature code examples applying the package in different use cases.[173] In addition to a comprehensive documentation, a continuous integration pipeline implements several steps that give insights into the code quality and correctness. The first step is performing unit tests that are implemented to check the core functionality of the PyGromosTools package. Every pull request to the main branch is automatically tested before merging, providing immediate information on potential bugs.

Moreover, in the next step, a tool for test coverage was added such that developers can easily detect parts of the package for which no tests have been added yet. Nevertheless, all functionality added to the package must come with the corresponding unit tests. The final step is a syntax style guide that provides suggestions for improved readability and standardized code writing.

Programming Paradigms

Python3 is a multi-paradigm language that allows mixing of object-oriented programming (OOP)[315] and functional programming paradigms. OOP enables the benefits of object inheritance and polymorphism.[313] These concepts are used extensively in PyGromosTools with state-driven contexts like file representations or the submission system classes. Especially the base class *_general_gromos_file* is a representative for the application of inheritance in the package. This class contains the fundamental read and write procedures that are identical for all GROMOS files and therefore only implemented once (see `https://github.com/rinikerlab/PyGromosTools/blob/main/pygromos/files/_basics/_general_gromos_file.py`).

Functional programming enables implementation of functions with Python's built-in map, apply, and zip operations, or using higher-order functional programming concepts like Python decorators.[313] As an example for the benefit of decorators, the principle of currying[316] was realized for the GROMOS++ integration into the *Gromos_System*, such that dynamically generated functions update the attribute files of the *Gromos_System* automatically and those do not need to be provided to the function call (compare Figure 5.5 versus Figure 5.8, and function *___SystemConstructionUpdater* in `https://github.com/rinikerlab/PyGromosTools/blob/main/pygromos/files/\gromos_system/`

`gromos_system.py`).

5.2.2 Code Structure

PyGromosTools itself can be split into a high-level and a low-level layer. The low-level layer interfaces the operating system, the job queueing system, and the GROMOS wrappers. These functions are used in the high-level layer to build more complex structures like the **simulations** modules or the *Gromos_System*. In the following, we will discuss the implementation of different modules of PyGromosTools. In general, the package consists of four modules: **data**, **file**, **simulations**, and **analysis**.

Files

The **file** module is a collection of classes that represent the different GROMOS files in Python (Figure 5.1). The design of the module is based on considerations discussed in the context of OOP. It makes extensive use of inheritance and polymorphism to build up a class hierarchy consistent with the GROMOS file structure with minimal duplication of code. In the GROMOS file structure, each file contains multiple blocks. These blocks in turn contain either a table of data or a sorted list of values. PyGromosTools represents this file structure such that the experienced behavior remains similar to the design of GROMOS. The structural basis is placed in the *_general_gromos_file* class, which contains the fundamental functionality encoded in the general file structure. Resulting functions are *read_file*, *write*, and *str*-operator overloading. This generic file class encapsulates classes derived from *_generic_gromos_blocks*, which provide generic functions like read and write as well as operator overloading. The most basic data structure is used to store the content of generic blocks. This

structure is for many blocks the *_generic_field*, either a primitive or a Pandas data frame. All these compartments generate specific classes for the different files used in the GROMOS environment. Features of these classes are:

- IO functionalities that not only allow writing of files but also writing obj states or converting files.
- Direct access of any type of GROMOS data in the objects enables a lean integration into Python3 scripts.
- Additional functionality that directly works on the file class includes calculating the root-mean-square deviation of MD trajectories or removing residues from coordinate files.

Most GROMOS files were implemented in PyGromosTools with an individual class derived from the base *_general_gromos_file* class, and sorted into the different categories according to their functionality (Figure 5.1). Due to the consequent class hierarchy, new blocks or files can be included very easily. The MD package involves many diverse types of files such as coordinate files, topology and simulation parameter files, coordinate and energy trajectory files, force-field files (FF), and others like replica-exchange outputs or NMR GROMOS++ program output files. The force-field class *Forcefield_System* represents a whole force field and contains the functionality to parametrize molecules.

With PyGromosTools, the MD trajectory files generated by GROMOS can be translated into a Pandas data frame for post-processing in Python. The data frame is stored in the compressed hf5 format[317] to make it highly storage efficient and fast on I/O. Pandas data frames have the advantage that data is stored in NumPy arrays, which are internally implemented as C arrays. As a consequence, Pandas data frames are very memory efficient and

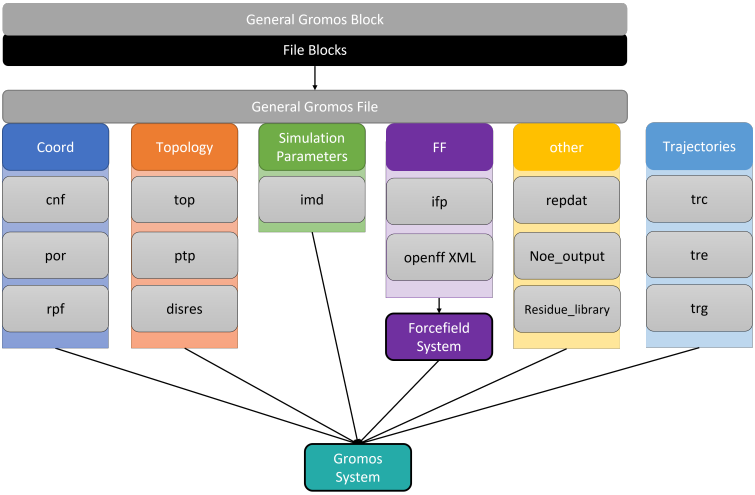


FIGURE 5.1: PyGromosTools **file** module is implemented with an OOP structure based on the GROMOS file structure. The base classes are the *_generic_GROMOS_block* and the *_general_GROMOS_file*. From these base classes, all other GROMOS blocks and GROMOS files are derived (with the exception of trajectories) to provide a consistent user-experience. Only the implemented file classes are shown for clarity. The *Gromos_System* class collects all files and makes them easily accessible for simulation approaches or other functionality that requires multiple GROMOS files.

basic mathematical operations on them (e.g., calculation of the mean, addition, etc.) are very fast, while these data frames still behave like Python objects.

In theory, PyGromosTools could also be used to introduce new file encodings without breaking legacy code. For example, the simulation parameters file could be translated into a JSON[238] or XML[318] data format, making file handling and readability easier by taking advantage of the key-value policy. In that regard, the Python ecosystem features a plethora of libraries that allow reading and writing these canonical file systems.

Gromos system All the different file formats implemented in PyGromosTools can be stored centralized in the *Gromos_System* class, which belongs to the high-level layer of PyGromosTools. This class is the central structure of PyGromosTools, from which systems are created or modified, simulated, and subsequently analyzed. The class can be used with a minimal set of files, i.e., a coordinate file (.cnf), topology file (.top), and a simulation parameter file (.imd), or just with a molecule SMILES or RDKit molecule to perform a complete automatic parametrization with a selected force field. Two force fields can currently be selected, the GROMOS Force Fields or the the open force field (OpenFF).[109] PyGromosTools provides tools to convert the OpenFF topology format to GROMOS format, perform sanity checks of the parameter files, or generate coordinate files from RDKit molecules. Additionally, the *Gromos_System* can adapt the simulation parameter file to use the correct functional forms required for the selected force field. All functions can also be called "manually".

Simulation Module

The submodules of **simulations** can be arranged into three complexity layers. At the base are the **gromos** and **hpc-queuing** submodules. These submodules are used by the **simulation** module to provide simulation functionality to carry out energy minimizations or MD simulations. At the top is the **approaches** submodule, which uses the lower layers to facilitate complex simulation approaches like RE-EDS or for the calculation of the heat of vaporization. This layer structure enables fast adaptation and extension of functionality (Figure 5.2). The submodule **gromos** contains the Python API to the GROMOS[94] and GROMOS++[252] binaries, providing quick access to their functionalities. The APIs of GROMOS and GROMOS++ are currently realized with bash

wrappers, although a proper C++ Python integration would likely be beneficial to improve the communication between the layers. A **pyGROMOSPP** compartment has been added that contains efficient Python implementations of several programs available in GROMOS++. In general, we anticipate that the advantages of the Python language will lead to more GROMOS++ functionality being implemented directly into the Python layer. Note that the **gromos** submodule can be used isolated from all other modules.

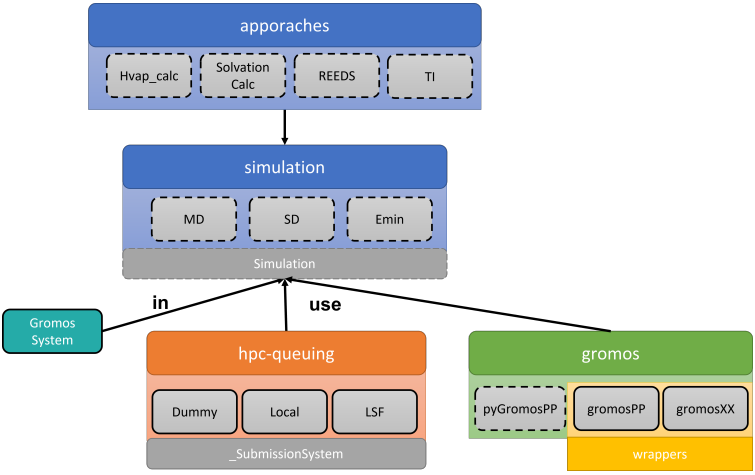


FIGURE 5.2: PyGromosTools **simulations** module contains four submodules, **hpc-queuing**, **gromos**, **simulation**, **approaches**. **gromos** is the API to all GROMOS and GROMOS++ functionality written in C++ as well as to a new module containing GROMOS++-like functionality in Python. The **hpc-queuing** submodule is used to set up simulations in different environments quickly (e.g., local execution or queuing with the LSF job management tool on a cluster). This is possible due to the commonly shared parent class *_SubmissionSystem*. Dashed lines around boxes represent functions, while bold lines around boxes imply an underlying class structure.

The **hpc-queuing** submodule builds an interface to job management tools like LSF by IBM, and provides the functionality

of job queueing for PyGromosTools scripts. The submodule is divided into a submission system and a job scheduling part. The **submission systems** is structured into an OOP-based structure with the parent class *_SubmissionSystem* that functions as an interface, ensuring the correct implementation of the child classes. Already implemented classes in **submission systems** are:

- *DUMMY* – a class meant for testing that only prints out strings.
- *LOCAL* – a class to execute a submitted job directly on the local machine via the operating system.
- *LSF* – a class to use the LSF queueing system for scheduling simulations on a HPC cluster. The class is currently optimized for the ETH HPC cluster Euler (`https://scicomp.ethz.ch/wiki/Euler`).

Besides the submission systems, the **hpc-queuing** submodule contains job scheduling tools that implement a scheduler-worker pattern (Figure 5.3). In this pattern, a scheduler function submits worker sub-scripts, which are dynamically generated from template workers, to the queueing system, effectively executing the submitted steps in order. Implemented workers are the simulation, the clean-up, and the analysis worker. The separation of these three tasks is essential to guarantee their correct execution. A good usage of the submission systems is to first develop and debug a pipeline with the DUMMY or LOCAL submission system, and only move to the HPC cluster once the scripts work as intended.

The two described submodules are combined in the *simulation* function, which executes the simulation. As input, the *simulation* function requires only a *Gromos_System* with all input files. If no system parameter file (.imd) is provided, a template file for

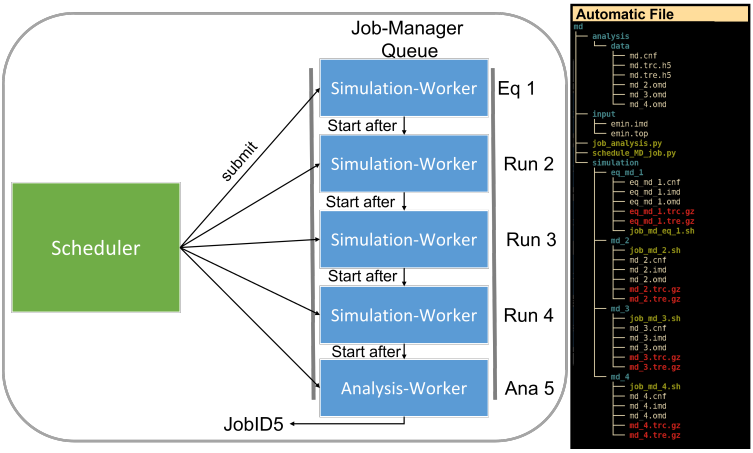


FIGURE 5.3: (Left): HPC-queuing submission pattern using a scheduler function, which submits worker scripts that perform the whole work or a part of it. The worker script is directly executed or submitted to a job management tool based on the provided submission system class. (Right): Automatically generated file structure. Note that the simulation directory is compressed in the process. The md directory is decompressed in the figure to illustrate the concept.

the chosen simulation approach will be mapped onto the system. The default submission system is *LOCAL*, which can be changed to *LSF* for larger jobs. Besides automatic scheduling of tasks, the **simulation** module also provides automatic file management (Figure 5.3). The output of the *simulation* function is a copy of the input *Gromos_System* with the additional output files of the simulation. If the simulation is executed locally, these files are directly parsed into the class. If the simulation is sent to a queue, on the other hand, all not-yet-present output files are marked with a *_future_promise* flag. This flag prevents the *Gromos_System* from executing tasks on the object that requires the presence of the simulation results. In more complex schemes with simulation chaining, these tagged files can be used to submit follow-up

simulation steps (e.g., RE-EDS *s*-optimization or energy offset rebalancing[239]). Once a file is created at a later time point, the *_future_promise* flag is removed by the *_check_promises* function, and the file is automatically loaded into the *Gromos_System* object. Additional smaller features are: checking whether simulations or analysis scripts have finished successfully, checking whether the current job is already submitted to the queue, concatenating files, and storing of trajectories as compressed .h5 files in the analysis/data directory (Figure 5.3).

The final component of the **simulations** module is the **approaches** submodule. In this submodule, top-level simulation approaches are stored, e.g., for calculating solvation free energies, binding free energies, or heats of vaporization. The provided functions offer a high level of automation. For example, the only required input for the calculation of the heat of vaporization is a valid *Gromos_System* created from a SMILES string. PyGromosTools will automatically create multiple systems (gas phase, liquid phase) and simulations (energy minimization, equilibration, and production run). After all the simulations have been submitted and executed, the analysis will be performed. In this case, the calculated heat of vaporization will be returned for the provided SMILES.

Analysis and Data Modules

The **analysis** module provides functionality for analyzing simulation properties (e.g., atom-positional RMSD, free-energy differences, etc.) and visualizing 3D structures directly in the Jupyter notebook using py3dmol.[319] Also time series of properties can be visualized using matplotlib.[182] The **analysis** module is the "youngest" one in PyGromosTools and will be further extended in the future. Finally, the **data module** provides parameters for the

GROMOS 54A7 force field[102] together with template simulation parameter files for good starting points.

5.3 APPLICATIONS AND EXAMPLES

5.3.1 Gromos System and Simulation Modules

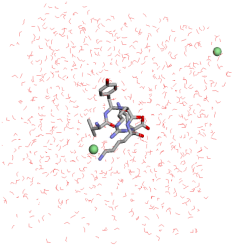
Figure 5.4 shows how PyGromosTools can be used to perform the simulation set-up of a short peptide in solution, following the example from the official GROMOS tutorial.[320] The set-up steps include (i) generating a topology file for the given peptide residue sequence, (ii) generating a GROMOS coordinate file, (iii) solvating the peptide in water, and (iv) adding two Na+ ions to the system to neutralize the overall charge. In this procedure, all actions on files contained in the *Gromos_System* are directly accessed and stored in the *Gromos_System*, leading to a simplified function call of the GROMOS++ functions.

After the system generation, the *Gromos_System* can be combined with the **simulations** module to carry out different types of simulations. For standard simulations, the default settings provided in the simulation block can be used. An example is shown in Figure 5.5 for the energy minimization and production run. If a more complex set-up is required, the user can manually set a simulation parameter file (.imd) in the *Gromos_System* and modify it as required. An important feature of the *simulation* function is that the given *Gromos_System* will not be modified by it (immutability principle). Only the returned system will be a modified version of the initial input. This immutability approach

Code: Building a System

```
from pygromos.files import gromos_system
from pygromos.utils import bash #bash wrappers
from pygromos.data.ff import Gromos54A7 #Force Field - Parameter Files

work_dir = bash.make_folder(project_dir+"/compact")
build_system = gromos_system.Gromos_System(work_folder=work_dir, system_name='peptide')

#build topology for system in gromos_system
build_system.make_top(in_sequence="NH3+ VAL TYR ARG LYSH GLN COO-",
            in_solvent="H2O",
            in_building_block_lib_path = Gromos54A7.mtb,
            in_parameter_lib_path= Gromos54A7.ifp) # generate topology

#build coordinate file in gromos_system from provided Tutorial PDB
build_system.pdb2gromos(in_pdb_path=build_system.work_folder+"/../Tutorial_System/input/peptide.pdb
")
build_system.add_hydrogens() #add missing hydrogen atoms to the system

#Solvate the box
from pygromos.data.solvent_coordinates import spc
build_system.sim_box(in_solvent_cnf_file_path=spc,
            periodic_boundary_condition="r",
            minwall=0.8,
            threshold=0.23,
            rotate=True)

#Add Ions
build_system.ion(negative=[2, "CL-"])

#write out all files into workfolder
build_system.rebase_files()

#visualize start structure
build_system.cnf.visualize()
```

Output:



Automatic File

```
compact
├── peptide.cnf
└── peptide.top
```

FIGURE 5.4: PyGromosTools code to perform the simulation set-up the pentapeptide in water from the GROMOS tutorial.[320] The *Gromos_System* is the central object for the system generation, and all functions can be called from there. The visualization of the start structure is called in the last line of code. The function *rebase_files()* triggers an automatic file management function writing out all files, that are indlcuded in the *Gromos_System*.

for *Gromos_System* in the *simulation* function avoids confusion when the system state changes during scripting.

To perform longer simulations on a HPC cluster, the simulation can be split into multiple parts, which are submitted by changing

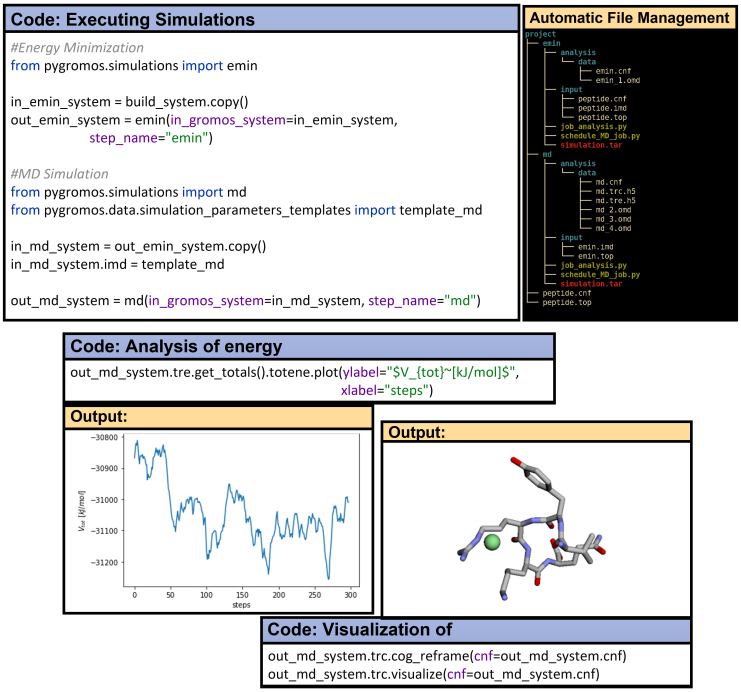


FIGURE 5.5: PyGromosTools code to perform an energy minimization (emin) and MD simulation (md) using the default settings (top left). After the simulation, the output coordinates can be visualized in the Jupyter notebook (bottom right), and the energy trajectory can be directly analyzed (bottom left). The simulation approach is realized with automatic file management in the background that is easy to understand (top right).

the number of *simulation_runs* (Figure 5.6). This parameter controls how often the simulation with the given parameters is executed. To submit the jobs to the queue, only the *submission_system* parameter needs to be changed from *LOCAL* to *LSF*.

Code: HPC - Simulation

```
#MD Simulation
from pygromos.simulations.modules.preset_simulation_modules import md
from pygromos.simulations.submission_systems import LSF

lsf=LSF(nmpi=6) #submission system on Euler
in_md_system = out_emin_system.copy()

out_md_system = md(in_gromos_system=in_md_system,
                   step_name="md",
                   submission_system=lsf,
                   equilibration_runs=1,
                   simulation_runs=3)
```

FIGURE 5.6: PyGromosTools code to perform a longer MD simulation on a HPC cluster in three parts (*simulation_runs*=3). The adaptation of the code is minimal, as only a submission system has to be changed to *LSF*.

5.3.2 Further Examples

Example of File Handling

The PyGromosTools package is able to read and write GROMOS files, modify parameter values, or use output files to perform further analysis. In Figure 5.5, a code example is given in which the template simulation parameter file (.imd) from PyGromosTools is modified by the user to change the number of time steps and set the temperature to 600 K. Finally, the parameter file is written out (e.g. to start a GROMOS simulation in the command shell).

GROMOS Wrappers

The GROMOS API provides users with many functions from the GROMOS environment, including documentation and reasonable defaults suited for most simulation set-ups. The GROMOS and GROMOS++ binaries are used from the operating system *PATH* variable or can be redirected by providing a binary directory

Code: Files

```
from pygromos.files.simulation_parameters import imd
from pygromos.data.simulation_parameters_templates import template_md

temperature=600 #K

imd_file = imd.Imd(template_md)
imd_file.STEP.NSTLIM = 3000 #change simulation steps number
imd_file.WRITETRAJ.NTWX = 300 #change write out of coordinates
imd_file.WRITETRAJ.NTWX = 300 #change write out of energies

#adapt temperature
imd_file.MULTIBATH.TEMP0 = [temperature for x in
                            range(imd_file.MULTIBATH.NBATHS)]

imd_file.write(out_imd_path) #write out imd to file
```

FIGURE 5.7: PyGromosTools code to modify the template system parameter file (.imd) as desired (e.g., changing the number of time steps or setting the temperature). The different parameters can be directly accessed and modified via their GROMOS name. Afterwards, the object can be written out.

path to the object construction. After constructing the wrappers, the full functionality is accessible from the object. The return value of the functions will always be the output file generated by the command. The API functionality of these wrappers is used throughout PyGromosTools to accomplish more complex tasks (Figure 5.8).

Code: Gromos Wrappers

```
from pygromos.gromos.gromosPP import GromosPP
from pygromos.gromos.gromosXX import GromosXX

#Init Gromos Classes
gromPP = GromosPP()
gromXX = GromosXX()

#build topology for system in gromos_system
out_top_path = gromPP.make_top(in_building_block_lib_path=Gromos54A7.mtb,
            in_parameter_lib_path=Gromos54A7.ifp,
            in_sequence="NH3+ VAL TYR ARG LYSH GLN COO-",
            in_solvent="H2O",out_top_path=topo_dir+"/peptide.top")

#Energy Minimization
out_emin_vacuum = out_dir + "/" + out_prefix
out_emin_log = gromXX.md_run(in_imd_path=imd.path,
          in_topo_path=top.path,
          in_coord_path=cnf.path,
          out_prefix=out_emin_vacuum, verbose=True)
```

FIGURE 5.8: The GROMOS and GROMOS++ wrappers facilitate the usage of GROMOS functionality from Python. After constructing the GROMOS wrapper classes, the programs of GROMOS++ and GROMOS are accessible as a function of the object. Note that the GROMOS 54A7 force-field parameters are taken directly from the data module inside the package.

5.4 CONCLUSION AND OUTLOOK

This chapter introduced PyGromosTools, an API for the GROMOS software package and the GROMOS++ package of programs that facilitates easy and fast set-up, running, and analysis of GROMOS MD simulations. The structure of the four modules (**file**, **simulation**, **analysis**, and **data**) was presented and illustrated with examples. The package is used already in multiple scientific projects and is a central element of the RE-EDS pipeline (Chapter 5).

Modern software development concepts are realized in PyGromosTools such that the API can be used for fast and sustainable development of complex solutions in different projects. Key elements of the package are (i) making data and functionality accessible in Python, and (ii) supporting the simulation set-up by providing classes for file management and job queueing on HPC clusters. With its underlying code structure and development decisions, PyGromosTools enables writing of readable and transferable code, which will result in increased shareability and enhanced reproducibility of scientific work.[321]

6

Modulation of the Passive Permeability of Semipeptidic Macrocycles: A Computational Investigation *

"... everything that living things do can be understood in terms of the jigglings and wigglings of atoms "

Richard Feynman, Lectures on Physics[16]

Incorporating small modifications to peptidic macrocycles can have a major influence on their properties. For instance, N-methylation has been shown to impact permeability. A better understanding of the relationship between permeability and structure is of key importance as peptidic drugs are often associated

* This Chapter is reproduced in part from Christian Comeau†, Benjamin Ries†, Thomas Stadelmann, Jacob Tremblay, Sylvain Poulet, Ulrike Fröhlich, Jérôme Côté, Pierre-Luc Boudreault, Rabeb Mouna Derbali, Philippe Sarret, Michel Grandbois, Grégoire Leclair, Sereina Riniker, and Éric Marsault, J. Med. Chem. 64 (2021) 5365–5383 , with permission from the American Chemical Society. † These authors contributed equally.

with unfavorable pharmacokinetic profiles. A collection of 36 semipeptidic macrocycles was generated by our collaborators, exploring two small structural changes: peptide-to-peptoid substitution and various methyl placements on the nonpeptidic linker. The permeability of these compounds was assessed in parallel artificial membrane permeability assays (PAMPA) and Caco-2 assays. A permeability cliff was identified triggered by the stereochemistry change of a single methyl group in the linker. This cliff was studied using a combination of MD simulations and NMR measurements, resulting in a hypothesis on how the change modifies the conformational behavior of the macrocycles.

6.1 INTRODUCTION

Macrocycles have recently gathered increased interest in medicinal chemistry as beyond rule-of-5 (bRO5) molecules.[322–329] A key feature of these molecules is their conformational complexity that can be leveraged in drug design to target protein-protein interactions.[330–334] Such protein-protein interactions are typically characterized by large flat binding sites that are difficult to target with small molecules.[49] If the macrocycles are peptidic, their toxicity is often relatively low.[335] Most Food and Drug Administration (FDA)-approved macrocyclic drugs belong to natural products (e.g., erythromycin, tacrolimus) or peptides (e.g., sandostatin, eptifibatide).[336] Peptidic or semipeptidic scaffolds bridge the gap between small molecules and biologics. An advantage of this molecule class is that they are relatively easy to synthesis and allow a broad choice of natural and non-natural amino acids required for rapid and thorough pharmacophoric exploration. The main challenge with peptides resides in their physicochemical and pharmacokinetics-ADME (absorption, distribution, metabolism, and excretion) properties. While cyclic peptides are typically more stable to proteases compared to their linear counterparts, their high polarity often translates into low bioavailability.[50,337] However, some cyclic peptides were found to cross cell membranes.[50,338,339] Developing tools and knowledge to optimize and better predict their structure–permeability relationship is, therefore, a requirement for the field. Such quest found inspiration in studies of the cyclic undecamer cyclosporine A, which is administered orally. One prominent structural feature of this natural macrocycle is the high number of N-methylated residues (7 out of 11) and its dynamic structural adaptation

to its environment described as chameleonic behavior.[39,53,340] The effect of N-methylation on the permeability of cyclic hexa- and heptapeptides has been systematically investigated since the number and position of N-methylations may be beneficial or detrimental for permeability.[339,341,342,342–344] Less explored are the N-alkylated glycines – aka peptoids – in which the side chain has been moved from the α-carbon to the amide nitrogen.[345] Similar to N-methylation, this modification removes one H-bond donor and removes one stereogenic center, and induces glycine-like secondary structures. The peptoid amide also facilitates cis-trans isomerization compared to the corresponding N-methylation.[346]

More recently, the impact of the dynamics of macrocycles in response to their environment, which can range from polar in water, nonhomogeneous in the presence of its target, to lipophilic in the membrane, has been appreciated.[39–41,53,347] Studying macrocycles with computational methods leads to multiple criteria identified as being possibly essential for chameleonic behavior. Examples of these criteria are the presence of intramolecular H-bonds, 3D polar surface areas (3D-PSA), or kinetic Markov models as metrics for how macrocyclic structures yield polar atoms and rigidification of the backbone cycle into certain polar/apolar states.[38–41,348] A powerful tool to modulate the properties of peptidic macrocycles is the inclusion of a nonpeptidic tether unit.[349–351] This tether can serve multiple purposes: in the context of a target interacting with a specific sequence, various tethers can be screened without modifying the peptide recognition sequence, while providing a simple handle for modulating affinity and pharmacokinetic properties. Small modifications in size, shape, or functional groups on the tether can dramatically influence on this kind of constrained system.[352] The relationship between structure and permeability is known to be elusive for this class of compounds,

with small structural modifications often yielding permeability cliffs.[338,341–343,351,353–356]

To investigate the structural effects of a tether with a length of five atoms and the peptide-peptoid change on the compound permeability, our collaborators synthesized a collection of 36 semipeptidic macrocycles.[351,357] The structure of the compounds was composed of a tripeptide tethered head-to-tail with a nonpeptidic linker (Figure 6.1). Two classes of modifications were explored: single peptoid replacement and regio- and stereocontrolled linker C-methylation.

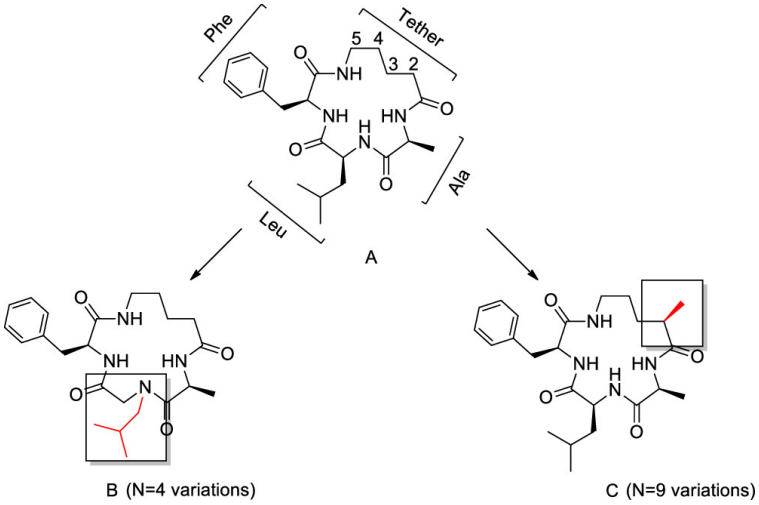


FIGURE 6.1: Synthesis strategy of our collaborators for model compound (**A**) and two types of modifications: Nala, Nleu, and Nphe peptoids (**B** showing Nleu) and regio/stereocontrolled C-methylation (**C** showing 2R methylation).[357]

The passive permeability of the resulting macrocycles was measured by our collaborators in the parallel artificial membrane permeability assay (PAMPA)[358,359] and their cellular permeability in the Caco-2 assay[359,360] (Figure 6.2).[357]

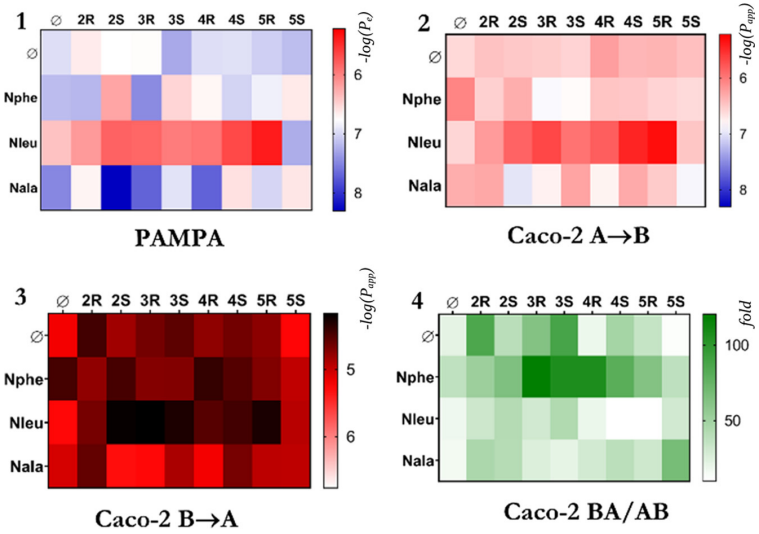
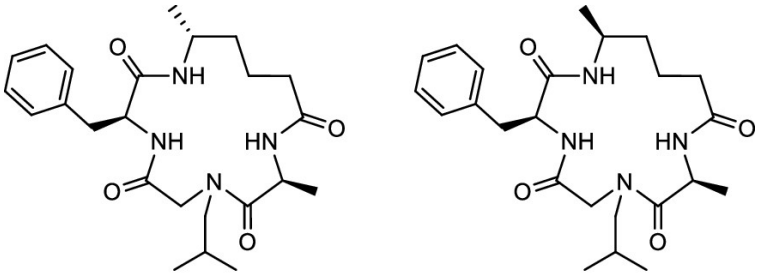


FIGURE 6.2: Permeability results in the form of heatmaps. For heatmaps 1–3, the values are expressed as $-log(P_{app})$, so lower values mean higher permeability (in order of increasing permeability: blue, white, red, and black). Heatmap 4 shows the BA/AB ratio, which represents a measure of efflux.

Based on the permeability data, we selected two pairs of diastereomers that differ only by their stereochemistry of the tether methyl group (Figure 6.3). While one pair (Nleu-5R/S) differs greatly in their passive permeability behavior, the second one (Nleu-2R/S) does not. Prior studies on cyclosporine A showed that the conformational behavior of cyclic peptides in the context of membrane permeability can be studied by performing extensive molecular dynamics (MD) simulations in apolar and polar environments (e.g., chloroform and water) to mimic the behavior outside and inside a membrane.[38–41] Therefore, we carried out MD simulations of each of the four selected macrocycles in water and chloroform. The simulations results were validated by comparing



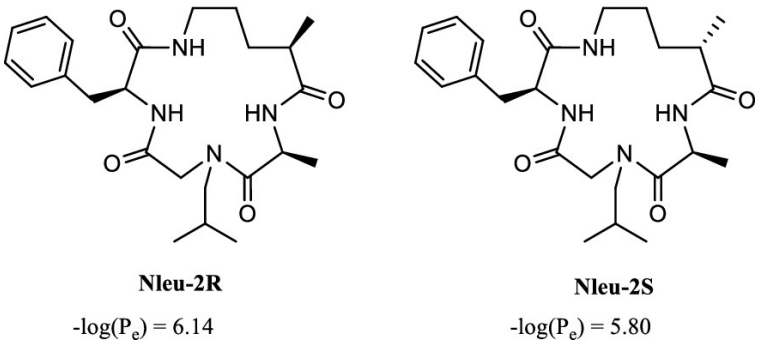


FIGURE 6.3: Four semipeptidic macrocycles were selected from the collection. In contrast to the pair Nleu-2R/S (bottom), the pair Nleu-5R/S (top) behave significantly different in the permeability assays. All molecules were studied with experimental NMR analysis and molecular dynamics (MD) simulations in a polar and apolar environment.

to solution NMR measurements of the compounds.[361,362] Finally, we used different metrics such as torsional angles, hydrogen-bond formation, and 3D-PSA[363] to assess and compare the conformational behavior of the compounds.

6.2 COMPUTATIONAL DETAILS

In the computational studies, two pairs of structurally similar cyclic peptides were selected, i.e., Nleu-5R/Nleu-5S and Nleu-2R/Nleu-2S (Figure 6.3). The first pair presents a "permeability cliff", i.e., the two peptides show a large difference in the passive permeability in the PAMPA assay (Nleu-5R: $-log(P_e) = 5.4$; Nleu-5S: $-log(P_e) = 7.2$), despite a high structural similarity. In contrast, the second pair is similar in both structure and permeability (Nleu-2R: $-log(P_e) = 6.1$; Nleu-2S: $-log(P_e) = 5.8$). For each of these four peptides, 250 starting coordinates were generated using the macrocycle variant of the OMEGA conformer generator from OpenEye.[51,364,365] Conformers were energy-minimized for maximum 2000 steps with the steepest descent[271] approach using the GROMOS software package[94] with the GROMOS 54A7 force field.[102] Each minimized starting conformation was solvated in a cubic box of simple-point-charge (SPC) water[250] (on average, 4172 solvent molecules) or chloroform[366] (on average, 980 solvent molecules). For each system, an MD simulation of 101 ns length was performed under isothermal–isobaric (NPT) conditions with the leap-frog integration algorithm[123,367] and a time step of 2 fs. The first 1 ns was discarded as equilibration. Bond lengths were constrained with SHAKE[112] and a tolerance of 10^{-4} nm. Nonbonded interactions were calculated using a twin-range scheme

with a short-range cutoff of 0.8 nm and a long-range cutoff of 1.4 nm. The electrostatic nonbonded contributions beyond the long-range cutoff were calculated with the reaction-field[88] approach, setting the dielectric permittivity to 61.0[251] for water, and to 4.8[366] for chloroform. The temperature was kept constant at 300 K using the weak coupling scheme[125] and a coupling time of 0.1 ps^{-1}. The pressure was kept at 1.031 bar (1 atm) with the same type of algorithm, a coupling time of 0.5 ps^{-1}, and an isothermal compressibility of 0.001654 bar^{-1} for chloroform and 0.0004575 bar^{-1} for water. Translational motion of the center of mass of the simulation box was removed every 2 ps. Energies and coordinates were written every 5 ps.

Trajectory analysis was performed with PyEmma[368] and MDTraj.[369] The selection of features for the structural clustering consisted of the distances between all pairs of polar atoms and the backbone torsional angles, resulting in total 57 features. This selection was reduced to three to five dimensions (depending on the peptide) with TICA[370] using a cumulative variance of 0.9 as criterion and a TICA correlation lag time of 50 ps. Based on these TICs, the frames were clustered with a common nearest neighbor (CNN) algorithm[371,372] using a cutoff of 0.2 and a similarity of 20. Comparison of selected clusters with NMR experiments was performed with the GROMOS++ package of programs.[252] The coefficients for the Karplus curve were taken from Vögeli *et al.*[373] Analysis of hydrogen bonds and torsional angles was performed with MDTraj. The 3D-PSA was calculated with our implementation[40] of the workflow in Ref. 348 using PyMol.[222] Statistical analysis of all results was carried out using the Python packages Pandas, NumPy and SciPy.[178]

6.3 RESULTS AND DISCUSSION

From the PAMPA and Caco-2 assays performed by our collaborators,[357] a large "permeability cliff" between Nleu-5R and 5S could be identified (Figure 6.2). Permeability cliffs were also observed between some other pairs of epimers, but to a lesser extent (e.g., Nala-4R vs 4S with $-log(P_e) = 7.6$ and 6.6, Nphe-2R vs 2S with $-log(P_e) = 7.2$ and 6.3, and Nphe-3R vs 3S with $-log(P_e) = 7.5$ and 6.5). Extensive MD simulations were carried out in a polar and apolar environment (i.e., water and chloroform) to study the influence of the stereocenter change on the conformational behavior of the molecules. As a negative control group, we used the molecule pair Nleu-2R and 2S.

6.3.1 Starting Configurations

The starting conformations used for the MD simulations were generated with the macrocycle variant of the OMEGA conformer generator from OpenEye.[51,364,365] The generated conformers showed similar distributions in terms of hydrogen bonds (H-bonds) and backbone torsional angles for the enantiomer pairs (Tables 6.1 and 6.2). Therefore, we started the MD simulations for each enantiomer pair from similar starting points. In these conformer ensembles, approximately 50% of the structures had a trans-peptoid bond for each molecule, and 50% had a cis-peptoid bond.

TABLE 6.1: Hydrogen-bond occurrence in percentage for the starting conformations of Nleu-5R, Nleu-5S, Nleu-2R, and Nleu-2S.

Hydrogen bond [%]	Nleu-2R	Nleu-2S	Nleu-5R	Nleu-5S
NLEU-O tether-NH	14	15	9	9
ALA-O tether NH	<1	<1	<1	<1
PHE-O Ala-NH	<1	<1	<1	<1
ALA-O PHE-NH	24	7	21	19

TABLE 6.2: Percentage of starting conformations with zero, one, or two hydrogen bonds for Nleu-5R, Nleu-5S, Nleu-2R, and Nleu-2S.

Hydrogen bond [%]	0	1	2
Nleu-2R	36	52	1
Nleu-2s	39	51	1
Nleu-5R	34	52	13
Nleu-5S	37	51	12

6.3.2 CNN CLUSTERING

The cumulative 25 μs simulation data for each peptide and solvent were clustered separately based on the backbone dihedrals and the polar atom distances. The resulting clusters could be structurally classified depending on the conformation of the peptoid bond (i.e., cis or trans; see Tables 6.3 and 6.4). The number of generated clusters varied but usually the size of the clusters decreased rapidly. Conformations with a cis or trans-peptoid bond were cleanly separated into different clusters.

The cis-trans isomerization represents a very slow process in the simulations, which occurred only rarely (Table 6.5). Due to the low number of transitions, the process could not be modeled robustly. Therefore, the clusters with the cis- and trans-peptoid bond are analyzed separately in the following.

TABLE 6.3: List of clusters identified in the simulations in chloroform with the respective population (in percentage). Clusters with the peptoid bond in trans-conformation are labeled with *. In the other clusters, the peptoid bond is in cis-conformation.

Molecule	Cluster	Size [%]	Molecule	Cluster	Size [%]
NLeu-5R	1*	36.2	NLeu-5S	1*	41.0
	2	18..1		2	16.0
	3	16.1		3	9.2
	4*	15.4		4	7.4
	5	3.5		5*	6.6
	6	1.5		6*	0.6
	7	0.5		7	0.6
	8	0.3		8	0.2
	9	0.2		9	0.2
	Noise	8.2		10	0.1
				Noise	18.0
Nleu-2R	1*	53.3	Nleu-2S	1*	26.2
	2	17.6		2*	18.3
	3	11.7		3	15.7
	4	10.2		4	6.5
	5	2.8		5	6.3
	Noise	4.4		6	5.2
				7*	5.1
				8	1.3
				9*	0.6
				10*	0.5
				11	0.3
				12*	0.3
				Noise	13.0

TABLE 6.4: List of clusters identified in the simulations in water with the respective population (in percentage). Clusters with the peptoid bond in trans-conformation are labeled with *. In the other clusters, the peptoid bond is in cis-conformation.

Molecule	Cluster	Size [%]	Molecule	Cluster	Size [%]
NLeu-5R	1*	46.7	NLeu-5S	1*	51.3
	2	22.4		2	16.8
	3	20.8		3	8.7
	4	6.1		4	8.0
	5	1.5		5	3.7
	6*	0.6		6	3.6
	7	0.4		7	1.7
	8*	0.4		8*	1.7
	9*	0.2		9*	0.8
	Noise	1.0		10	0.4
				11*	0.2
				Noise	3.2
Nleu-2R	1*	50.2	Nleu-2S	1*	42.0
	2	21.0		2	19.0
	3	6.6		3	12.0
	4	5.6		4	7.7
	5	4.7		5	6.3
	6	2.6		6	3.5
	7	1.6		7	2.7
	Noise	7.8		8*	1.4
				9*	0.1
				Noise	5.0

TABLE 6.5: Occurrence of cis-trans isomerization of the peptoid bond during the MD simulations (25 μs in water and chloroform, respectively.

Molecule	Chloroform		Water	
	Cis →Trans	Trans →Cis	Cis →Trans	Trans →Cis
Nleu-5R	14	1	3	0
Nleu-5S	9	0	0	1
Nleu-2R	15	7	0	0
Nleu-2S	6	1	9	5

6.3.3 NMR Validation

The NMR experiments in chloroform-d ($CDCl_3$) showed that the four compounds adopt at least two different conformations in solution.[357] The major conformer was identified with all amides in trans conformation. It was not possible to assign the minor conformers due to signal overlap and low intensity. In the case of Nleu-5R and Nleu-5S, a third conformer could be identified based on exchange spectroscopy (EXSY) cross-peaks in the nuclear Overhauser enhancement spectroscopy (NOESY) spectrum, which is barely detectable in the ^{1}H spectrum. The corresponding conformer ratios are listed in Table 6.6.

Table 6.6: Ratios of conformer population observed in the NMR spectra ($CDCl_3$).

Compound	Ratio
Nleu-2R	100:8
Nleu-2S	100:3
Nleu-5R	100:4:0
Nleu-5S	100:16:1

The results from the MD simulations were compared to the NMR data of the major conformer, in particular the $^3J_{\mathrm{HN–H}\alpha}$ coupling constants and the NOE-derived interproton distances (Tables 6.5 and 6.9).

The clusters with all amides in trans conformation are in good agreement with the $^3J_{\mathrm{HN–H}\alpha}$ coupling constants (Figure 6.4), whereas the clusters containing the cis-peptoid bond deviate significantly from the experimental values. For Nleu-2R, the $^3J_{\mathrm{HN–H}\alpha}$ coupling analysis is missing as we could not determine the $^3J_{\mathrm{HN–H}\alpha}$ couplings reliably due to line broadening in the spectrum. The NOE-derived upper distance bounds are also generally reproduced in these clusters (Figures 6.6 -6.9). Based on

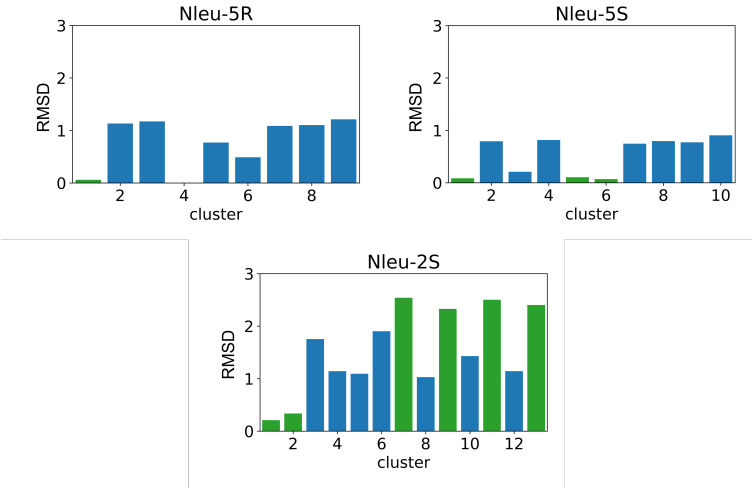


FIGURE 6.4: Root-mean-square deviation (RMSD, in hertz) between $^{3}J_{HN-H\alpha}$ coupling constants in chloroform from NMR measurements and from MD simulations. Clusters with the peptoid bond in trans conformation are shown in green.

these findings, we focused the analysis in the following on those clusters, which have a reasonable agreement with the NMR data (i.e., clusters 1 and 4 for Nleu-5R, clusters 1, 5, and 6 for Nleu-5S, cluster 1 for Nleu-2R, and clusters 1 and 2 for Nleu-2S).

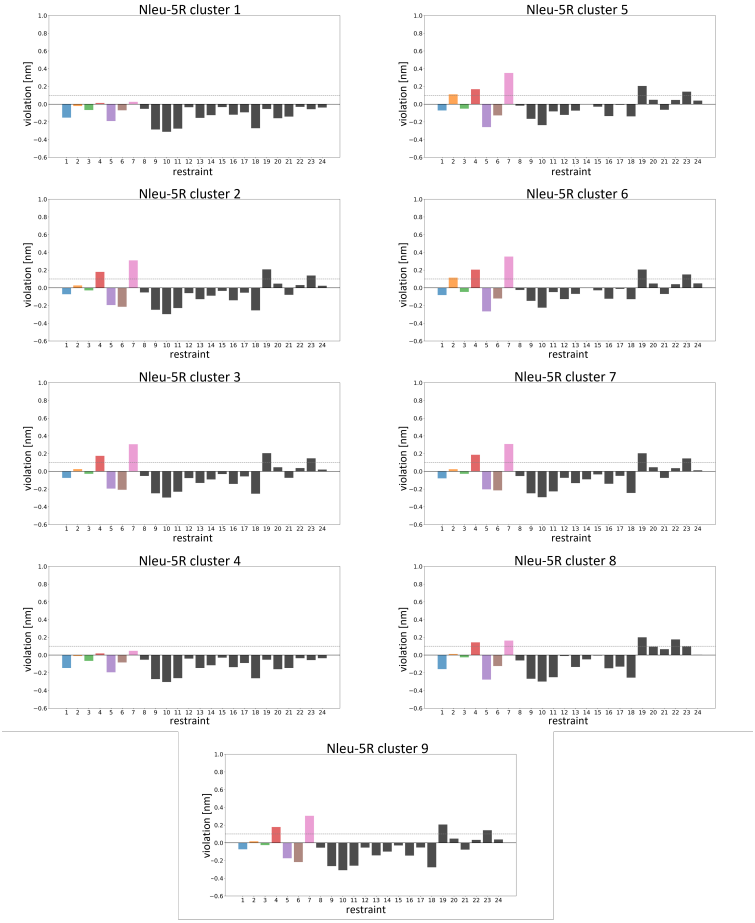


FIGURE 6.5: Violations of the NOE-derived upper distance bounds of Nleu-5R in chloroform by the clusters identified in the simulations in chloroform. Distances between residues across the backbone ring are colored. Distances between neighboring residues are shown in black. The dashed line indicates the expected uncertainty of the experimental upper bounds.

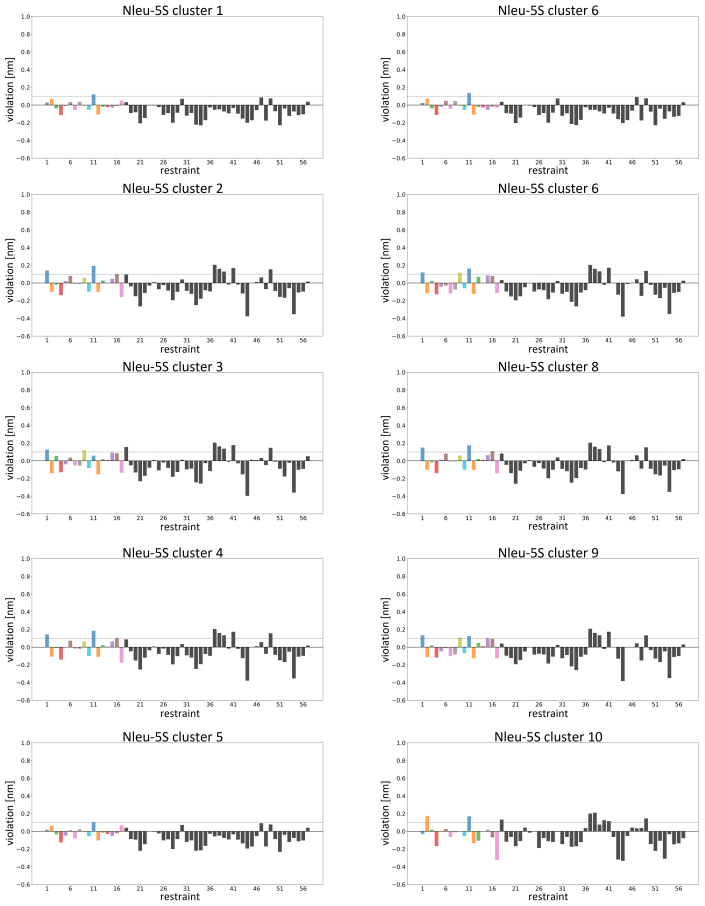


FIGURE 6.6: Violations of the NOE-derived upper distance bounds of Nleu-5S in chloroform by the clusters identified in the simulations in chloroform. Distances between residues across the backbone ring are colored. Distances between neighboring residues are shown in black. The dashed line indicates the expected uncertainty of the experimental upper bounds.

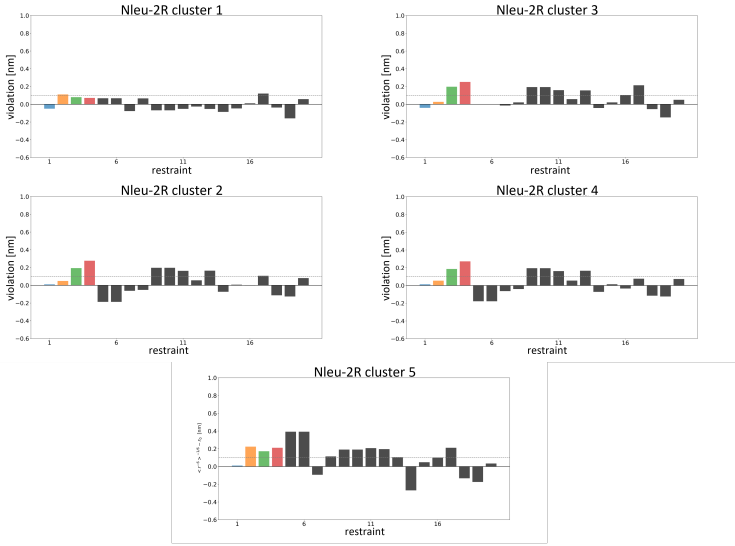


FIGURE 6.7: Violations of the NOE-derived upper distance bounds of Nleu-2R in chloroform by the clusters identified in the simulations in chloroform. Distances between residues across the backbone ring are colored. Distances between neighboring residues are shown in black. The dashed line indicates the expected uncertainty of the experimental upper bounds.

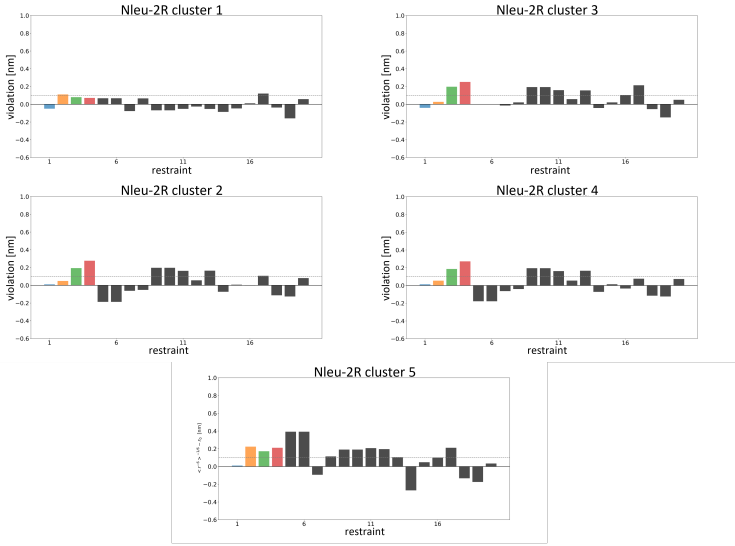


FIGURE 6.8: Violations of the NOE-derived upper distance bounds of Nleu-2S in chloroform by clusters 1–6 identified in the simulations in chloroform. Distances between residues across the backbone ring are colored. Distances between neighboring residues are shown in black. The dashed line indicates the expected uncertainty of the experimental upper bounds.

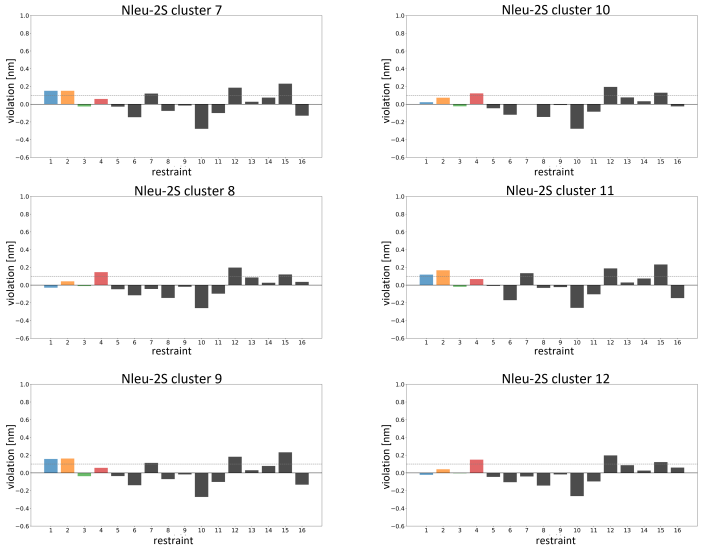


FIGURE 6.9: Violations of the NOE-derived upper distance bounds of Nleu-2S in chloroform by clusters 7–12 identified in the simulations in chloroform. Distances between residues across the backbone ring are colored. Distances between neighboring residues are shown in black. The dashed line indicates the expected uncertainty of the experimental upper bounds.

6.3.4 Conformation Analysis

A necessary condition for good membrane permeability is the adoption of conformations that shield polar groups optimally from the apolar environment.[348,363,374] Therefore, we first analyzed the hydrogen-bonding patterns in the clusters in chloroform. For the peptides in this study, a maximum number of two H-bonds can be formed in a conformation due to ring strain. As can be seen in Table 6.7, the percentage of sampled conformations with two H-bonds differs significantly between Nleu-5R (30%) and Nleu-5S (7%). At the same time, the percentage of conformations without a H-bond is increased for Nleu-5S (25%) compared to Nleu-5R (8%). For the other pair, Nleu-2R and Nleu-2S, the percentages are more similar and in between those of Nleu-5R and Nleu-5S.

Table 6.7: Percentage of sampled conformations with zero, one, or two hydrogen bonds in chloroform. Analysis was restricted to the clusters with the trans-peptoid bond.

Number of hydrogen bonds [%]	0	1	2
Nleu-5R	8	63	30
Nleu-5S	25	68	7
Nleu-2R	15	64	21
Nleu-2S	13	74	13

For a given molecule in an apolar environment, having access to conformations in which polar groups are shielded – such as by H-bonding – should be energetically favorable. To assess this effect, we extracted the potential energy of the peptides (i.e., intramolecular and peptide-solvent contributions) from the trajectories. The normality of each potential-energy distribution was confirmed by the Shapiro–Wilk test[375] (Table 6.8).

The Fisher t-test[376] was employed to determine if the means of the distributions differ statistically significantly ($p < 0.05$). This

TABLE 6.8: Average potential energy of the peptides (i.e., sum of the intramolecular ⟨V⟩ contributions and the peptide–solvent contributions) together with the p-value of the Shapiro-Wilk test for the simulations in chloroform and water, respectively. The significance limit for the p-value was 0.05.

Molecule	Chloroform		Water	
	⟨V⟩[kJ/mol]	$p_{\text{Shapiro-Wilk}}$	⟨V⟩[kJ/mol]	$p_{\text{Shapiro-Wilk}}$
Nleu-5R	-217.08	0.0	-117.77	0.0
Nleu-5S	-208.39	$5.9 * 10^{-8}$	-115.64	$5.24 * 10^{-10}$
Nleu-2R	-211.17	0.0	-117.84	0.0
Nleu-2S	-216.63	$7.6 * 10^{-39}$	-116.91	0.0

TABLE 6.9: Results of the Fisher t-test to validate the significance of the deviations in the average potential energy of the peptides. The significance limit for the p-value was 0.05.

Molecule	Chloroform	Water
Nleu-5R - Nleu-5S	∼0.0	∼0.0
Nleu-2R - Nleu-2S	∼0.0	$2.7 * 10^{-77}$

was found to be the case for each pair of distributions (Table 6.9). On average, the potential energy of Nleu-5R is 9 kJ/mol lower (i.e., more favorable) in chloroform compared to Nleu-5S, whereas the difference in the average potential energy between Nleu-2R and Nleu-2S is 6 kJ/mol. In many studies in the literature, it was found that the 3D-PSA is a good measure for the degree of polar shielding in conformations.[351,363,377,378] However, for the present set of four peptides, no correlation was observed between the 3D-PSA and the potential energy (Figure 6.10). The ring strain in the relatively small backbone cycle of the peptides affects the geometry of the intramolecular H-bonds, which is likely not reflected appropriately in the 3D-PSA calculation.

In summary, the ranking Nleu-5R < Nleu-2S < Nleu-2R < Nleu-5S, which was found in terms of both hydrogen-bonding patterns and potential energies, matches well with the experimental

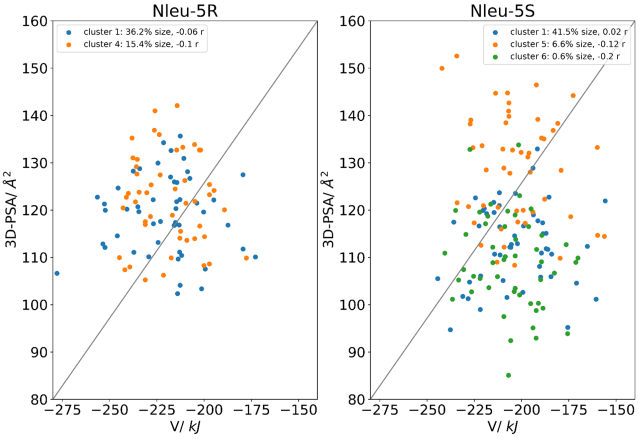


FIGURE 6.10: Correlation between the 3D-PSA and the potential energy of the corresponding conformation (i.e., sum of intramolecular contributions and peptide–solvent contributions) for Nleu-5R and Nleu-5S in chloroform. The 100 structures closest to the cluster center were taken for the clusters with the trans-peptoid bond. The trend for an expected linear correlation is shown as gray line. The legend contains the cluster population in percentage and the Spearman correlation coefficient r.

permeability data. The findings described above indicate that the change in stereochemistry of the methyl group in position 5 between Nleu-5R and Nleu-5S leads to different conformational behavior. A detailed analysis of the H-bonds showed that only Nleu-5S forms a H-bond between Ala-O and the tether-NH with an occurrence of 24% in chloroform (Table 6.10). This H-bond across the ring of Nleu-5S prevents the formation of other H-bonds (Figure 6.11B). Such a conformation with a single H-bond is likely less favorable (compared to one with more H-bonds) in chloroform because less polar groups are shielded. In the dominant conforma-

tion of Nleu-5R, on the other hand, two H-bonds can be formed across the ring (Figure 6.11A).

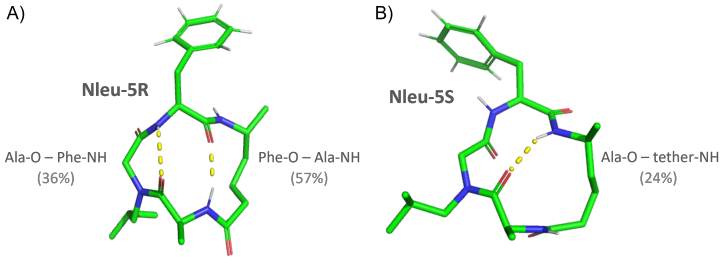


FIGURE 6.11: Snapshots of Nleu-5R (**A**) and Nleu-5S (**B**) from MD simulations in chloroform. Hydrogen bonds are shown with their percentage of the absolute occurrence in chloroform in the trans-peptoid clusters. Pictures were generated with PyMol.[222]

TABLE 6.10: Hydrogen bond occurrence in percentage for the sampled conformations in chloroform. Analysis was restricted to the clusters with the trans-peptoid bond.

H-bond [%]	Nleu-2R	Nleu-2S	Nleu-5R	Nleu-5S
Nleu-O tether-NH	74	37	28	33
Ala-O tether-NH	<1	<1	<1	24
Phe-O Ala-NH	<1	35	57	<1
Ala-O Phe-NH	27	25	36	17

Next, we analyzed the torsional-angle distributions in the backbone ring of the peptides. The change in stereochemistry of the methyl group at position 5 leads to a shift in the torsional-angle distributions of the tether units for Nleu-5S compared to Nleu-5R (Figure 6.12A). This shift results in a bent conformation of the ring (Figure 6.12B), which allows only one H-bond to form between Ala-O and tether-NH (Figure 6.12B). There is also a shift in the backbone torsional-angle distributions between Nleu-2R

and Nleu-2S, however, to a much smaller extent (Figure 6.14). Noteably a longrange effect in the dihedral distribution for the phenylalanine residue was detected. The missing hydrogen bond and the resulting rotation of the carbonyl group seem to hinder the free rotation of the phenylalanine rest (Figure 6.13).

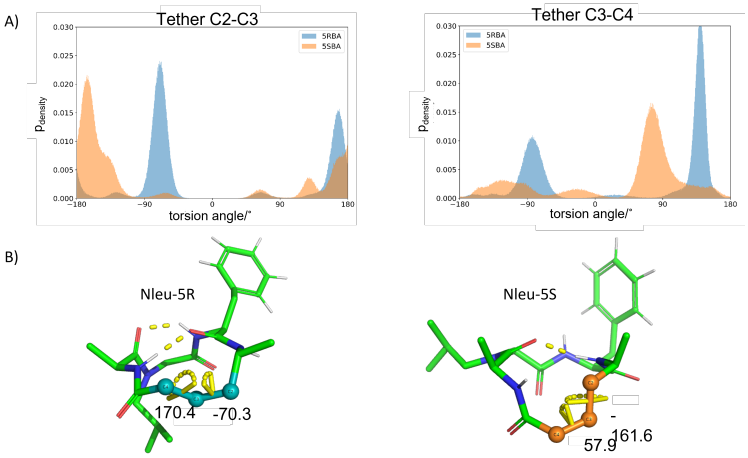


FIGURE 6.12: (**A**) Torsional-angle distributions of the tether in Nleu-5R (blue) and Nleu-5S (orange) in chloroform. The analysis was restricted to the clusters with the trans-peptoid bond. (**B**) Torsional angles of the tether (shown in cyan and orange) corresponding to the peaks of the distributions. Pictures were generated with PyMol.[222] The change in the stereocenter also affects the χ_1-angle of the phenylalanine residue as the tether conformation hinders the rotation around this torsion due to a steric clash with the carbonyl group that is facing out of the backbone ring (Figure 6.13).

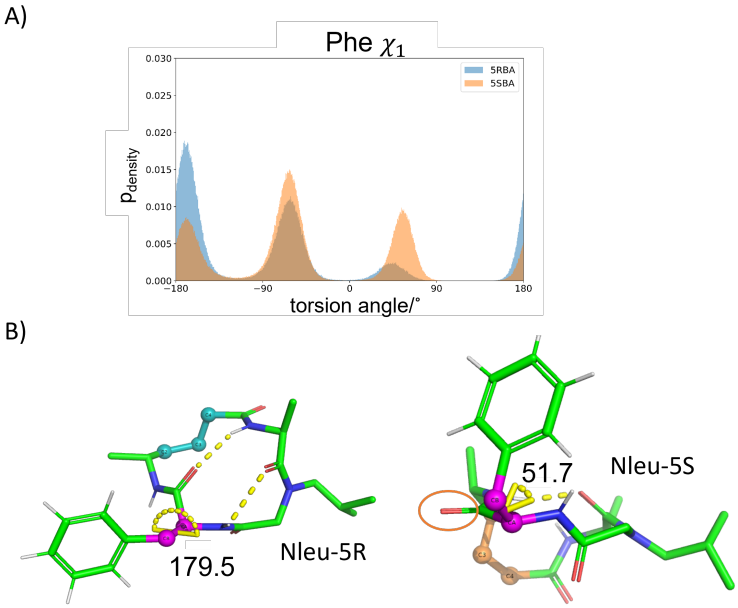


FIGURE 6.13: (**A**) Torsional-angle distributions of the χ_1 torsional angle of the phenylalanine residue in Nleu-5R (blue) and Nleu-5S (orange) in chloroform. Analysis was restricted to the clusters with the trans-peptoid bond. (**B**) χ_1 torsional angle of the phenylalanine residue (shown in purple) corresponding to the peaks of the distributions. The backbone carbonyl interferes with the rotation around this torsion is highlighted with a red circle. Pictures were generated with PyMol.[222]

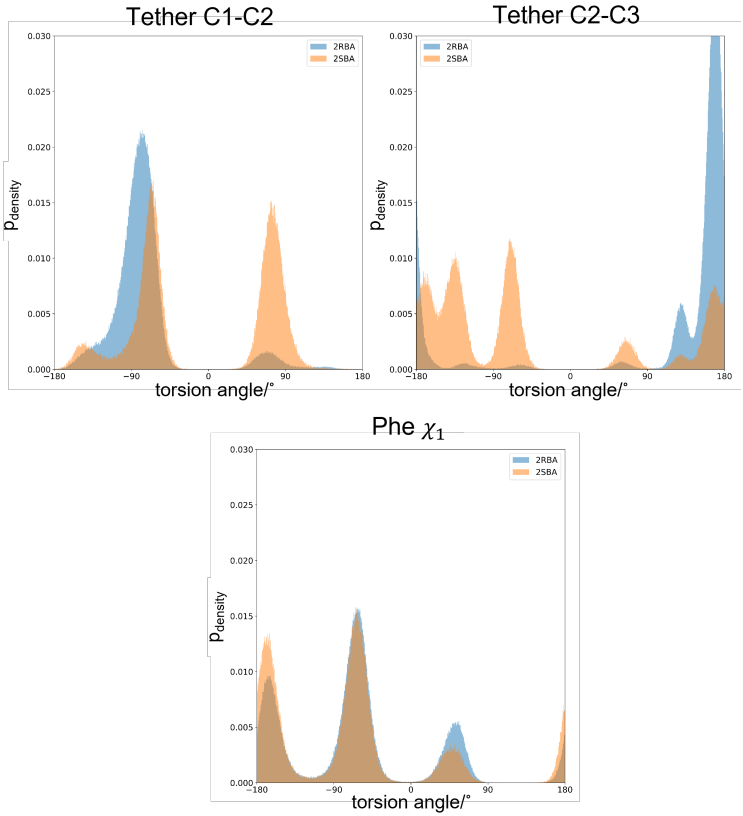


FIGURE 6.14: Torsional-angle distributions of the tether in Nleu-2R (blue) and Nleu-2S (orange) in chloroform. Analysis was restricted to the clusters with the trans-peptoid bond. The distribution density is plotted as a function of the torsional angle, with a bin size of 0.5 deg.

TABLE 6.11: Percentage of sampled conformations with zero, one, or two hydrogen bonds for Nleu-5R, Nleu-5S, Nleu-2R, and Nleu-2S in water. Analysis was restricted to the clusters with the trans-peptoid bond.

Number of hydrogen bonds [%]	0	1	2
Nleu-5R	87	12	1
Nleu-5S	74	26	0
Nleu-2R	78	22	0
Nleu-2S	76	24	0

The analysis of the hydrogen-bonding patterns in water showed a significant decrease for the intramolecular H-bond populations, as they competed with the intermolecular H-bonds to the water. Specifically, Nleu-5R has a higher percentage (about 10%) of conformers with no H-bonds compared to Nleu-2R, Nleu-2S, and Nleu-5S (Table 6.11). The conformations containing two intramolecular H-bonds nearly vanished. The positions of the intramolecular H-bonds are mainly focused on the Nleu-O and tether-NH position (Table 6.12). Nevertheless, it could be observed that the Ala-O and tether-NH, which was unique to Nleu-5S in the apolar environment, is again most present in water for Nleu-5S in contrast to the other possible intramolecular H-bonds. In general, however, no significant differences between the peptides could be observed in water.

TABLE 6.12: Hydrogen-bond occurrence in percentage for the sampled conformations in water for Nleu-5R, Nleu-5S, Nleu-2R, and Nleu-2S. The analysis was restricted to the clusters with the trans-peptoid bond.

Hydrogen bond [%]	Nleu-2R	Nleu-2S	Nleu-5R	Nleu-5S
Nleu-O tether-NH	14.5	18.3	12	9.75
Ala-O tether-NH	5.5	3.6	<1	15.25
Phe-O Ala-NH	<1	<1	<1	1
Ala-O Phe-NH	2	1	<1	<1

The findings, taken together, suggest that the permeability cliff observed between Nleu-5R and Nleu-5S is related to their propensity for conformations with a maximized number of intramolecular H-bonds in the apolar environment. Their ability to adopt such conformations is in turn affected by the stereochemistry of the methyl group at position 5 in the tether as it determines the preferred torsional angles of the tether.

6.4 CONCLUSION

Combining the permeability data generated by our collaborators, NMR measurements, and MD simulations allowed us to draw some conclusions on how the structural differences between the selected macrocycles influence their membrane permeability. The pair of Nleu-2R/S did not show a significant change in permeability depending on the stereocenter change. In contrast, the second pair Nleu-5R/S showed a significant effect on permeability behavior. Nleu-5R is the most permeable compound from the compound collection synthesized by our collaborators, while Nleu-5S is with its low permeability the exception among the Nleu compounds. In the MD simulations, we observed different H-bond patterns for Nleu-5R and Nleu-5S in the chloroform. While Nleu-5R frequently adopted a conformation with the maximum number of two H-bonds (optimal shielding of the polar groups), such a conformation was rare for Nleu-5S. A detailed analysis of the torsional angle preferences highlighted the underlying steric effects.

In contrast to other studies, we could not retrieve a correlation between the 3D-PSA and the PAMPA permeability for the four selected macrocycles. The backbone cycle of the peptides is relatively small, thus minor structural changes affecting the geometry of the intramolecular H-bonds are likely not reflected appropriately in the 3D-PSA calculation. In summary, we studied the relationship between small structural changes and the resulting permeability behavior for four semipeptidic macrocycles. The location and especially the stereochemistry of the methyl group played an important role in the intramolecular hydrogen-bonding pattern, impacting the passive membrane permeability of the compounds.

7 Outlook

"Jede wahre Geschichte ist eine unendliche Geschichte."
"Every real story is a never ending story."

Michael Ende,
Die Unendliche Geschichte[379]

In Chapters 2 - 4, developments of free-energy methodology and its application for binding free energy calculations were presented. Chapter 6 described a study of the conformational behavior of semi-peptidic macrocycles to connect the change of a single stereocenter with their lipid-membrane permeability. Finally, Chapters 2 and 5 illuminated aspects of software development in science.

7.1 DEVELOPMENT OF SCIENTIFIC SOFTWARE

Software development is and will be an essential part of computational chemistry for different reasons. Software in this area becomes steadily more complex in order to increase computational efficiency and the amount of available functionality. Further, many

published studies are difficult to reproduce or methods cannot be further developed because the used source code is unavailable or not transferable to different platforms.[321,380] The open-source movement, which has become an important driving force of academic sciences, can be considered a role model for improving this situation.[287,321] In this line, many journals have started to request non-commercial software of a publication to be open source.[381–383] Overall, these developments will help to increase the readability and transferability of code.[321] The latter issue can be solved by using programming environment tools such as pip or anaconda for Python.[293,294] All software packages developed in this thesis are open source and can be accessed via the GitHub repository rinikerlab.

Next, an outlook for the PyGromosTools package is provided. In our opinion, PyGromosTools combines scripting and programming languages in a productive way, making the package easy to use and efficient, thus fulfilling key conditions of modern scientific codes of conduct of scientific journals. A long-term vision is to build from PyGromosTools a PyGROMOS package. This package should integrate GROMOS++[252] into the Python layer to make it easier and faster to extend its functionality. Efficiency issues could be solved by using Numba or other efficiency-improving tools. In addition, the GROMOS MD engine[94] should be integrated tighter with the use of binding tools like pyBinds or SWIG.[288,290] These changes are expected to lead to a more "future-ready" GROMOS environment that provides easier access to its functionality. The package could be compiled by the Python package managing tools.

7.2 PERSPECTIVES FOR RE-EDS

In recent years, an increasing amount of publications on path-free multi-state methods has appeared.[170,221,225,240,259] An attractive aspect of such methods is their computational efficiency. This effect can be attributed to (i) the phase-space overlap that allows simultaneous sampling of multiple end-states, and (ii) no predefined paths for the sampling of the end-state transitions is required, thus allowing the system to find an optimal spanning tree of the state graph dynamically. Insufficient sampling resulting from the choice of difficult state transitions was reported in the literature as a reason for the efficiency loss of pairwise path-dependent methods.[195] To assess how well each end-state is sampled, robust metrics are needed for path-free multi-state methods. With such metrics in hand, the simulation parameters can be refined to ensure equal sampling of all end-states, as described for RE-EDS in Chapter 4.

7.2.1 Method Development

A pre-processing pipeline was established to optimize all RE-EDS parameters based on the defined general metrics. In order to reduce the optimization time of the pipeline, multiple options can be explored. First, the information about the replica-exchange gap region included in the initial short simulation to obtain energy offsets could be used to refine the initial s-distribution (instead of a logarithmic distribution).

A second idea is to investigate whether the initial state optimization process could be used to estimate the energy offsets. This could be done by using the Jarzynski equality and the work that is required to change the initial maximally contributing end-state

of the system to the desired end-state.[384,385]

$$\Delta F_{BA} = -\frac{1}{\beta} \ln \langle e^{-\beta W_{BA}} \rangle_R, \tag{7.1}$$

where the work W is defined as,[385]

$$W_{BA}(t) = \int_0^t \frac{\partial H(t)}{\partial t} dt. \tag{7.2}$$

Another significant improvement could be the integration of the information from all replicas in the final free-energy estimation, not only from replica $s = 1.0$. For this, a free-energy estimator like M-BAR[386] or any other multi-state (here in the sense of replicas) reweighting scheme may be used. Note that BAR[168] was already applied to λ-EDS in Ref. 387.

Finally, the sampling of the implemented 2D-RE-EDS approach, exchanging both s-parameters and energy offsets, needs to be tested. For conformational studies, it may be of interest to develop a 2D - RE-EDS variant that exchanges both s-parameters and temperature to enhance sampling further.

7.2.2 RE-EDS Software Development

From an implementation point of view, the RE-EDS pipeline[239] could be made more dynamic such that it is decided on-the-fly which modules of the pipeline are needed to be applied during optimization. Such a dynamic modular approach could improve the robustness and efficiency of the pipeline.

7.2.3 RE-EDS Applications

In the future, aspects such as the complexity of transformation and the number of end-states in RE-EDS simulations will be further investigated. This experience gained from these studies could be used to develop a robust high-throughput approach with RE-EDS, where feasible subsets of ligands are selected from databases, e.g. by clustering. The subsets will share one or multiple reference ligands such that all relative binding free energies can be calculated. The clustering metric could be based on simple topological and 3D-structure-based criteria, or employ molecular dynamic fingerprints (MDFP),[388] thus including the dynamic ligand behavior into the clustering.

Another possible RE-EDS application could be to validate docking results. Docking is a commonly utilized method to generate ligand-protein complex.[389–391] However, the validation of such results is often non-trivial, especially as the docking scoring functions are relatively simplistic.[392] Therefore, MD simulations are usually employed to check the robustness of docking poses.[389,393–395] With RE-EDS, a performant approach could be established that evaluates the docking results. For this, a separated dual topology approach[213] with weak position restraints could be employed. A challenge might thereby be the undersampling region of the s-distribution when the end-states are clearly separated in the coordinate space.

7.3 MEMBRANE PERMEABILITY BEYOND RULE OF 5

Cyclic peptides belong to the so-called bRO5 class of compounds, and have a complex conformational behavior. Some cyclic peptides are able to passively cross membranes despite their size, which is one of their fascinating aspects.[340,355,396,397] A reason for this is hypothesized to be a chameleonic character in terms of their conformational behavior, which allows them to adapt to apolar and polar environments.[38–41] This interesting property might give rise to new concepts in rational drug design for bRO5 molecules. Important factors appear to be the shielding of polar atoms and the rigidification of the cyclic structure in the permeable conformation.[40,41] The ongoing modeling of how cyclic peptides pass through cell membranes could further increase our understanding of the mechanism of membrane permeation, and help to identify essential structural features of bioavailable cyclic peptides.

Abbreviations

1SS	one starting state
API	application programming interface
ATB	automated topology builder
BAR	Bennett acceptance ratio
bRO5	beyond Lipinski's rule-of-five
CHV	convex hull volume
CNN	common nearest neighbor
COG	center of geometry
EMIN	energy minimization
EDS	enveloping distribution sampling
FEP	free-energy perturbation
HPC	high performance computing
IDE	integrated development environment
JSON	JavaScript object notation
MAE	mean absolute error
MC	Monte Carlo
MD	molecular dynamics
N-LRTO	multi-state local round trip optimizer
N-GRTO	multi-state global round trip optimizer
NMR	nuclear magnetic resonance
OOP	object oriented programming
PAMPA	parallel artificial membrane permeability assay
PEOE	parallel energy offset estimation
PEP	Python enhancement proposal

PSA	polar surface area
RBFE	relative binding free energy
RE	replica exchange
RMSE	root mean square error
SD	stochastic dynamics
SMILES	simplified molecule input line entry system
SPC	simple point-charge model
SSM	starting state mixing
TI	thermodynamic integration

Bibliography

[1] J. W. Goethe, *Faust: Eine Tragoedie*, Cotta, 1825.

[2] G. Wagner, *An Account of NMR in Structural Biology*, Nat. Struct. 4 (1997) 841–844.

[3] Y. Shi, *A Glimpse of Structural Biology Through X-Ray Crystallography*, Cell. 159 (2014) 995–1014.

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

[5] L. O. Elkin, *Rosalind Franklin and the Double Helix*, Phys. Today. 56 (2003) 42–48.

[6] F. Crick, *Central Dogma of Molecular Biology*, Nature. 227 (1970) 561–563.

[7] L. Pauling, R. B. Corey, *Atomic Coordinates and Structure Factors for Two Helical Configurations of Polypeptide Chains*, Proc. National. Acad. Sci. USA 37 (1951) 235–240.

[8] M. F. C. Ladd, R. A. Palmer, R. A. Palmer, *Structure Determination by X-Ray Crystallography*, Springer, 1977.

[9] N. E. Jacobsen, *NMR Spectroscopy Explained: Simplified Theory, Applications and Examples for Organic Chemistry and Structural Biology*, John Wiley & Sons, 2007.

[10] P. R. L. Markwick, T. e. Malliavin, M. Nilges, *Structural Biology by NMR: Structure, Dynamics, and Interactions*, PLoS Comput. Biol. 4 (2008) 1–7.

[11] A. Doerr, *Single-Particle Cryo-Electron Microscopy*, Nat. Method. 13 (2016) 23–23.

[12] D. Agard, Y. Cheng, R. M. Glaeser, S. Subramaniam, *Chapter Two - Single-Particle Cryo-Electron Microscopy (Cryo-EM): Progress, Challenges, and Perspectives for Further Improvement*, in: Advances in Imaging and Electron Physics, Elsevier, 2014, pp. 113–137.

[13] Y. Cheng, R. M. Glaeser, E. Nogales, *How Cryo-EM Became So Hot*, Cell. 171 (2017) 1229–1231.

[14] W. Kühlbrandt, *The Resolution Revolution*, Science. 343 (2014) 1443–1444.

[15] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, D. C. Phillips, *A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-ray Analysis*, Nature. 181 (1958) 662–666.

[16] M. Karplus, J. A. McCammon, *Molecular Dynamics Simulations of Biomolecules*, Nat. Struct. Biology. 9 (2002) 646–652.

[17] D. Phillips, *Biomolecular Stereodynamics*(1981).

[18] H. Frauenfelder, G. A. Petsko, D. Tsernoglou, *Temperature-Dependent X-ray Diffraction as a Probe of Protein Structural Dynamics*, Nature. 280 (1979) 558–563.

[19] K. Wüthrich, G. Wagner, *NMR Investigations of the Dynamics of the Aromatic Amino Acid Residues in the Basic Pancreatic Trypsin Inhibitor*, FEBS Lett. 50 (1975) 265–268.

[20] D. A. Torchia, *Solid State NMR Studies of Protein Internal Dynamics*, Annu. Rev. Biophys. Bioeng. 13 (1984) 125–144.

[21] C. Dobson, M. Karplus, *Internal Motion of Proteins: Nuclear Magnetic Resonance Measurements and Dynamic Simulations*, in: Methods in Enzymology, Elsevier, 1986, pp. 362–389.

[22] J. A. McCammon, B. R. Gelin, M. Karplus, *Dynamics of Folded Proteins*, Nature. 267 (1977) 585–590.

[23] A. R. Leach, *Molecular Modelling – Principles and Applications*, Pearson Education Limited, 2001.

[24] M. Chavent, T. Reddy, J. Goose, A. C. E. Dahl, J. E. Stone, B. Jobard, M. S. P. Sansom, *Methodologies for the Analysis of Instantaneous Lipid Diffusion in MD Simulations of Large Membrane Systems*, Faraday. Discus. 169 (2014) 455–475.

[25] S. A. Hollingsworth, R. O. Dror, *Molecular Dynamics Simulation for All*, Neuron. 99 (2018) 1129–1143.

[26] W. F. vanGunsteren, H. J. C. Berendsen, *Computer Simulation of Molecular Dynamics: Methodology, Applications, and Perspectives in Chemistry*, Angew. Chem. Int. Ed. 29 (1990) 992–1023.

[27] W. F. vanGunsteren, J. Dolenc, A. E. Mark, *Molecular Simulation as an Aid to Experimentalists*, Curr. Opin. Struct. Biol. 18 (2008) 149–153.

[28] B. L. Tembre, J. Mc Cammon, *Ligand-Receptor Interactions*, Comput. Chem. 8 (1984) 281–283.

[29] J. D. Durrant, J. A. McCammon, *Molecular Dynamics Simulations and Drug Discovery*, BMC. Biol. 9 (2011) 71.

[30] J. D. Chodera, D. L. Mobley, M. R. Shirts, R. W. Dixon, K. Branson, V. S. Pande, *Alchemical Free Energy Methods for Drug Discovery: Progress and Challenges*, Curr. Opin. Struct. Biology. 21 (2011) 150–160.

[31] M. Aldeghi, A. Heifetz, M. J. Bodkin, S. Knapp, P. C. Biggin, *Accurate Calculation of the Absolute Free Energy of Binding for Drug Molecules*, Chem. Sci. 7 (2016) 207–218.

[32] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, *Comparison of Simple Potential Functions for Simulating Liquid Water*, J. Chem. Phys. 79 (1983) 926–935.

[33] W. L. Jorgensen, J. Tirado-Rives, *The OPLS [Optimized Potentials for Liquid Simulations] Potential Functions for Proteins, Energy Minimizations for Crystals of Cyclic Peptides and Crambin*, J. Am. Chem. Soc. 110 (1988) 1657–1666.

[34] C. D. Christ, T. Fox, *Accuracy Assessment and Automation of Free Energy Calculations for Drug Design*, J. Chem. Inf. Model. 54 (2014) 108–120.

[35] Z. Cournia, B. Allen, W. Sherman, *Relative Binding Free Energy Calculations in Drug Discovery: Recent Advances and Practical Considerations*, J. Chem. Inf. Model. 57 (2017) 2911–2937.

[36] Z. Cournia, B. K. Allen, T. Beuming, D. A. Pearlman, B. K. Radak, W. Sherman, *Rigorous Free Energy Simulations in Virtual Screening*, J. Chem. Inf. Model. 60 (2020) 4153 – 4169.

[37] K. Meier, J. P. Bluck, C. D. Christ, *Use of Free Energy Methods in the Drug Discovery Industry*, ACS Publications, 2021, Ch. 2, pp. 39–66.

[38] J. Witek, B. G. Keller, M. Blatter, A. Meissner, T. Wagner, S. Riniker, *Kinetic Models of Cyclosporin A in Polar and Apolar Environments Reveal Multiple Congruent Conformational States*, J. Chem. Inf. Model. 56 (2016) 1547–1562.

[39] J. Witek, M. Mühlbauer, B. G. Keller, M. Blatter, A. Meissner, T. Wagner, S. Riniker, *Interconversion Rates Between Conformational States as Rationale for the Membrane Permeability of Cyclosporines*, ChemPhysChem. 18 (2017) 3309.

[40] J. Witek, S. Wang, B. Schroeder, R. Lingwood, A. Dounas, H. J. Roth, M. Fouché, M. Blatter, O. Lemke, B. Keller, S. Riniker, *Rationalization of the Membrane Permeability Differences in a Series of Analogue Cyclic Decapeptides*, J. Chem. Inf. Model. 59 (2019) 294.

[41] S. Wang, G. König, H.-J. Roth, M. Fouché, S. Rodde, S. Riniker, *Effect of Flexibility, Lipophilicity, and the Location of Polar Residues on the Passive Membrane Permeability of a Series of Cyclic Decapeptides*, J. Med. Chem. 64 (2021) 12761–12773.

[42] S. J. Marrink, H. J. C. Berendsen, *Permeation Process of Small Molecules Across Lipid Membranes Studied by*

Molecular Dynamics Simulations, J. Phys. Chem. 100 (1996) 16729–16738.

[43] D. Bemporad, J. W. Essex, C. Luttmann, *Permeation of Small Molecules Through a Lipid Bilayer: A Computer Simulation Study*, J. Phys. Chem. 108 (2004) 4875–4884.

[44] A. L. Lomize, I. D. Pogozheva, *Physics-Based Method for Modeling Passive Membrane Permeability and Translocation Pathways of Bioactive Molecules*, J. Chem. Inf. Model. 59 (2019) 3198–3213.

[45] H. N. Hoang, T. A. Hill, D. P. Fairlie, *Connecting Hydrophobic Surfaces in Cyclic Peptides Increases Membrane Permeability*, Angew. Chem. Int. Ed. 60 (2021) 8385–8390.

[46] M. Sugita, S. Sugiyama, T. Fujie, Y. Yoshikawa, K. Yanagisawa, M. Ohue, Y. Akiyama, *Large-Scale Membrane Permeability Prediction of Cyclic Peptides Crossing a Lipid Bilayer Based on Enhanced Sampling Molecular Dynamics Simulations*, J. Chem. Inf. Model. 61 (2021) 3681–3695.

[47] K. M. Corbett, L. Ford, D. B. Warren, C. W. Pouton, D. K. Chalmers, *Cyclosporin Structure and Permeability: from A to Z and Beyond*, J. Med. Chem. 64 (2021) 13131–13151.

[48] C. A. Lipinski, F. Lombardo, B. W. Dominy, P. J. Feeney, *Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings*, Adv. Drug. Deliv. Rev. 46 (2001) 3–26.

[49] B. C. Doak, J. Zheng, D. Dobritzsch, J. Kihlberg, *How Beyond Rule of 5 Drugs and Clinical Candidates Bind to Their Targets*, J. Med. Chem. 59 (2016) 2312–2327.

[50] M. R. Naylor, A. T. Bockus, M. J. Blanco, R. S. Lokey, *Cyclic Peptide Natural Products Chart the Frontier of Oral Bioavailability in the Pursuit of Undruggable Targets*, Curr. Opin. Chem. Biol. 38 (2017) 141.

[51] V. Poongavanam, E. Danelius, S. Peintner, L. Alcaraz, G. Caron, M. D. Cummings, S. Wlodek, M. Erdelyi, P. C. D. Hawkins, G. Ermondi, J. Kihlberg, *Conformational Sampling of Macrocyclic Drugs in Different Environments: Can We Find the Relevant Conformations?*, AC. Omega. 3 (2018) 11742.

[52] A. Furukawa, J. Schwochert, C. R. Pye, D. Asano, Q. D. Edmondson, A. C. Turmon, V. G. Klein, S. Ono, O. Okada, R. S. Lokey, *Drug-Like Properties in Macrocycles Above MW 1000: Backbone Rigidity Versus Side-Chain Lipophilicity*, Angew. Chem. Int. Ed. 59 (2020) 21571–21577.

[53] E. Danelius, V. Poongavanam, S. Peintner, L. H. E. Wieske, M. Erdélyi, J. Kihlberg, *Solution Conformations Explain the Chameleonic Behaviour of Macrocyclic Drugs*, Chem. . Eur. J. 26 (2020) 5231.

[54] E. P. Barros, B. Ries, L. Böselt, C. Champion, S. Riniker, *Recent Developments in Multiscale Free Energy Simulations*, Curr. Opin. Struct. Biol. 72 (2022) 55–62.

[55] H. M. Senn, W. Thiel, *QM/MM Methods for Biomolecular Systems*, Angew. Chem. Int. Ed. 48 (2009) 1198–1229.

[56] X. Sheng, M. Kazemi, A. Żądło-Dobrowolska, W. Kroutil, F. Himo, *Mechanism of Biocatalytic Friedel-Crafts Acylation by Acyltransferase from Pseudomonas Protegens*, AC. Catal. 10 (2020) 570–577.

[57] M. Kazemi, J. Åqvist, *Chemical Reaction Mechanisms in Solution from Brute Force Computational Arrhenius Plots*, Nat. Commun. 6 (2015) 7293.

[58] U. Ryde, *Combined Quantum and Molecular Mechanics Calculations on Metalloproteins*, Curr. Opin. Struct. Biol. 7 (2003) 136–142.

[59] I. B. Gábor Nŕay-Szabánd, *Computer Modelling of Enzyme Reactions*, J. Mol. Struct. 666–667 (2003) 637–644.

[60] M. Rivera, M. Dommett, R. Crespo-Otero, *ONIOM(QM:QM') Electrostatic Embedding Schemes for Photochemistry in Molecular Crystals*, J. Chem. Theory Comp. 15 (2019) 2504–2516.

[61] X. Li, M. Wang, S. Zhang, J. Pan, Y. Na, J. Liu, B. Åkermark, L. Sun, *Noncovalent Assembly of a Metalloporphyrin and an Iron Hydrogenase Active-Site Model: Photo-Induced Electron Transfer and Hydrogen Generation*, J. Chem. Phys. 112 (2008) 8198–8202.

[62] M. Askerka, G. W. Brudvig, V. S. Batista, *The O2-Evolving Complex of Photosystem II: Recent Insights from Quantum Mechanics/Molecular Mechanics (QM/MM), Extended X-Ray Absorption Fine Structure (EXAFS), and Femtosecond X-Ray Crystallography Data*, Acc. Chem. Re. 50 (2017) 41–48.

[63] A. Schenkmayerova, G. P. Pinto, M. Toul, M. Marek, L. Hernychova, J. Planas-Iglesias, V. Daniel Liskova, D. Pluskal, M. Vasina, S. Emond, M. Dörr, R. Chaloupkova, D. Bednar, Z. Prokop, F. Hollfelder, U. T. Bornscheuer,

J. Damborsky, *Engineering the Protein Dynamics of an Ancestral Luciferase*, Nat. Commun. 12 (2021) 3616.

[64] V. Tozzini, *Coarse-Grained Models for Proteins*, Curr. Opin. Struct. Biol. 15 (2005) 144–150.

[65] C. Hyeon, D. Thirumalai, *Capturing the Essence of Folding and Functions of Biomolecules Using Coarse-Grained Models*, Nat. Commun. 2 (2011) 487.

[66] V. K. Shen, J. K. Cheung, J. R. Errington, T. M. Truskett, *Insights Into Crowding Effects on Protein Stability from a Coarse-Grained Model*, J. Biomech. Eng. 131 (2009) 071002–1 – 071002–6.

[67] F. Hong, J. S. Schreck, P. Šulc, *Understanding DNA Interactions in Crowded Environments with a Coarse-Grained Model*, Nucleic. Acid. Re. 48 (2020) 10726–10738.

[68] M. Friedel, D. J. Sheeler, J.-E. Shea, *Effects of Confinement and Crowding on the Thermodynamics and Kinetics of Folding of a Minimalist β -Barrel Protein*, J. Chem. Phys. 118 (2003) 8106–8113.

[69] X. Daura, A. E. Mark, W. F. vanGunsteren, *Parametrization of Aliphatic CHn United Atoms of GROMOS96 Force Field*, J. Comput. Chem. 19 (1998) 535–547.

[70] P. M. Morse, *Diatomic Molecules According to the Wave Mechanics. II. Vibrational Levels*, Phys. Rev. 34 (1929) 57–64.

[71] L. Pauling, M. L. Huggins, *Covalent Radii of Atoms and Interatomic Distances in Crystals Containing Electron-Pair Bonds*, Z. Krist. Cryst. Mater. 87 (1934) 205–238.

[72] C. S. Gillmor, *Coulomb and the Evolution of Physics and Engineering in Eighteenth-Century France*, Princeton University Press, 2017.

[73] J. Mackerell, D. Alexander, *Empirical Force Fields for Biological Macromolecules: Overview and Issues*, J. Comp. Chem. 25 (2004) 1584–1604.

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

[75] C. Oostenbrink, A. Villa, A. E. Mark, W. F. vanGunsteren, *A biomolecular Force Field Based on the Free Enthalpy of Hydration and Solvation: The GROMOS Force-Field Parameter Sets 53A5 and 53A6*, J. Comp. Chem. 25 (2004) 1656–1676.

[76] J. W. Ponder, D. A. Case, *Force Fields for Protein Simulations*, in: Protein Simulations, Academic Press, 2003, pp. 27–85.

[77] W. L. Jorgensen, J. Tirado-Rives, *Potential Energy Functions for Atomic-Level Simulations of Water and Organic and Biomolecular Systems*, Proc. Natl. Acad. Sci. USA. 102 (2005) 6665–6670.

[78] S. Riniker, *Fixed-Charge Atomistic Force Fields for Molecular Dynamics Simulations in the Condensed Phase: An Overview*, J. Chem. Inf. Model. 58 (2018) 565–578.

[79] S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, P. Weiner, *A new Force*

Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins, J. Am. Chem. Soc. 106 (1984) 765–784.

[80] M. F. Iozzi, T. Helgaker, E. Uggerud, *Assessment of Theoretical Methods for the Determination of the Mechanochemical Strength of Covalent Bonds*, Mol. Phys. 107 (2009) 2537–2546.

[81] W. F. vanGunsteren, S. Billeter, A. Eising, P. Hünenberger, P. Krüger, A. Mark, W. Scott, I. Tironi, *Biomolecular Simulation: The GROMOS96 Manual and User Guide*, Vdf. Hochschulverlag. AG. ETH. Zürich. Zürich. 86 (1996) 1–1044.

[82] L. Pauling, *The Nature of the Chemical Bond. Application of Results Obtained from the Quantum Mechanics and from a Theory of Paramagnetic Susceptibility to the Structure of Molecules*, J. Am. Chem. Soc. 53 (1931) 1367–1400.

[83] J. C. Slater, *Molecular Energy Levels and Valence Bonds*, Phys. Rev. 38 (1931) 1109–1144.

[84] A. Blondel, M. Karplus, *New Formulation for Derivatives of Torsion Angles and Improper Torsion Angles in Molecular Mechanics: Elimination of Singularities*, J. Comp. Chem. 17 (1996) 1132–1141.

[85] C. Dugave, L. Demange, *Cis–Trans Isomerization of Organic Molecules and Biomolecules: Implications and Applications*, Chem. Rev. 103 (2003) 2475–2532.

[86] H. L. Strauss, H. M. Pickett, *Conformational Structure, Energy, and Inversion Rates of Cyclohexane and Some Related Oxanes*, J. Am. Chem. Soc. 92 (1970) 7281–7290.

[87] P. Atkins, J. De Paula, *Atkins' Physical Chemistry*, OUP Oxford, 2014.

[88] I. G. Tironi, R. Sperb, P. E. Smith, W. F. vanGunsteren, *A generalized Reaction Field Method for Molecular Dynamics Simulations*, J. Chem. Phys. 102 (1995) 5451.

[89] T. Darden, D. York, L. Pedersen, *Particle Mesh Ewald: An $N \cdot \log(N)$ Method for Ewald Sums in Large Systems*, J. Chem. Phys. 98 (1993) 10089–10092.

[90] S. Kawai, A. S. Foster, T. Björkman, S. Nowakowska, J. Björk, F. F. Canova, L. H. Gade, T. A. Jung, E. Meyer, *Van Der Waals Interactions and the Limits of Isolated Atom Models at Interfaces*, Nat. Comm. 7 (2016) 11559.

[91] H. Margenau, *Van Der Waals Forces*, Rev. Mod. Phys. 11 (1939) 1–35.

[92] J. E. Jones, *On the Determination of Molecular Fields. I. from the Variation of the Viscosity of a Gas with Temperature*, Proc. R. Soc. Lond. 106 (1924) 441–462.

[93] H. J. C. Berendsen, D. vander Spoel, R. vanDrunen, *GROMACS: A Message-Passing Parallel Molecular Dynamics Implementation*, Comp. Phys. Comm. 91 (1995) 43–56.

[94] N. Schmid, C. D. Christ, M. Christen, A. P. Eichenberger, W. F. vanGunsteren, *Architecture, Implementation and Parallelisation of the GROMOS Software for Biomolecular Simulation*, Comput. Phys. Commun. 183 (2012) 890.

[95] P. Eastman, V. Pande, *OpenMM: A Hardware-Independent Framework for Molecular Simulations*, Comput. Sci. Eng. 12 (2010) 34–39.

[96] J. vanMeel, A. Arnold, D. Frenkel, S. Portegies Zwart, R. Belleman, *Harvesting Graphics Power for MD Simulations*, Mol. Simul. 34 (2008) 259–266.

[97] M. Levitt, S. Lifson, *Refinement of Protein Conformations Using a Macromolecular Energy Minimization Procedure*, J. Mol. Biol. 46 (1969) 269–279.

[98] P. K. Weiner, P. A. Kollman, *AMBER: Assisted Model Building with Energy Refinement. A general Program for Modeling Molecules and Their Interactions*, J. Comput. Chem. 2 (1981) 287–303.

[99] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, P. Kollman, *AMBER, a Package of Computer Programs for Applying Molecular Mechanics, Normal Mode Analysis, Molecular Dynamics and Free Energy Calculations to Simulate the Structural and Energetic Properties of Molecules*, Comput. Phys. Commun. 91 (1995) 1–41.

[100] K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. Klepeis, R. Dror, D. Shaw, *Improved SideChain Torsion Potentials for the AMBER FF99SB Protein Force Field Proteins*, J. Am. Chem. Soc. 78 (2010) 1950–1958.

[101] L. D. Schuler, X. Daura, W. F. vanGunsteren, *An Improved GROMOS96 Force Field for Aliphatic Hydrocarbons in the Condensed Phase*, J. Comp. Chem. 22 (2001) 1205–1218.

[102] N. Schmid, A. P. Eichenberger, A. Choutko, S. Riniker, M. Winger, A. E. Mark, W. F. vanGunsteren, *Definition and Testing of the GROMOS Force-Field Versions 54A7 and 54B7*, Eur. Biophys. J. 40 (2011) 843.

[103] A. K. Malde, L. Zuo, M. Breeze, M. Stroet, D. Poger, P. C. Nair, C. Oostenbrink, A. E. Mark, *An Automated Force Field Topology Builder (ATB) and Repository: Version 1.0*, J. Chem. Theory Comput. 7 (2011) 4026–4037.

[104] M. Stroet, B. Caron, K. M. Visscher, D. P. Geerke, A. K. Malde, A. E. Mark, *Automated Topology Builder Version 3.0: Prediction of Solvation Free Enthalpies in Water and Hexane*, J. Chem. Theory Comput. 14 (2018) 5834–5845.

[105] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. a. Swaminathan, M. Karplus, *CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations*, J. Comput. Chem. 4 (1983) 187–217.

[106] A. D. MacKerell Jr, J. Wiorkiewicz-Kuczera, M. Karplus, *An All-Atom Empirical Energy Function for the Simulation of Nucleic Acid*, J. Am. Chem. Soc. 117 (1995) 11946–11975.

[107] A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, M. Karplus, *All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins*, J. Phys. Chem. 102 (1998) 3586–3616.

[108] W. L. Jorgensen, D. S. Maxwell, J. Tirado-Rives, *Development and Testing of the OPLS all-Atom Force Field on Conformational Energetics and Properties of Organic Liquids*, J. Am. Chem. Soc. 118 (1996) 11225–11236.

[109] Y. Qiu, D. G. A. Smith, S. Boothroyd, H. Jang, D. F. Hahn, J. Wagner, C. C. Bannan, T. Gokey, V. T. Lim, C. D. Stern, A. Rizzi, B. Tjanaka, G. Tresadern, X. Lucas, M. R. Shirts, M. K. Gilson, J. D. Chodera, C. I. Bayly, D. L. Mobley, L.-P. Wang, *Development and Benchmarking of Open Force Field V1.0.0 – The Parsley Small-Molecule Force Field*, J. Chem. Theory Comput. 17 (2021) 6262–6280.

[110] K. Sprenger, V. W. Jaeger, J. Pfaendtner, *The General AMBER Force Field (GAFF) Can Accurately Predict Thermodynamic and Transport Properties of Many Ionic Liquids*, J. Phys. Chem. 119 (2015) 5882–5895.

[111] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, A. D. Mackerell Jr., *CHARMM General Force Field: A Force Field for Drug-Like Molecules Compatible with the CHARMM All-Atom Additive Biological Force Fields*, J. Comp. Chem. 31 (2010) 671–690.

[112] J. P. Ryckaert, G. Ciccotti, H. J. C. Berendsen, *Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of N-Alkanes*, J. Comput. Phys. 23 (1977) 327.

[113] G. Ciccotti, J. Ryckaert, *Molecular Dynamics Simulation of Rigid Molecules*, Comput. Phys. Rep. 4 (1986) 346–392.

[114] S. Miyamoto, P. A. Kollman, *Settle: An Analytical Version of the SHAKE and RATTLE Algorithm for Rigid Water Models*, J. Comput. Chem. 13 (1992) 952–962.

[115] B. Hess, H. Bekker, H. J. C. Berendsen, J. G. E. M. Fraaije, *LINCS: A Linear Constraint Solver for Molecular Simulations*, J. Comput. Chem. 18 (1997) 1463–1472.

[116] P. Debye, *Näherungsformeln Für Die Zylinderfunktionen Für Große Werte Des Arguments Und Unbeschränkt Veränderliche Werte Des Index*, Math. Ann. 67 (1909) 535–558.

[117] M. R. Hestenes, E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, J. Re. Natl. Bur. Stand. 49 (1952) 409–436.

[118] F. Cazals, T. Dreyfus, D. Mazauric, C.-A. Roth, C. H. Robert, *Conformational Ensembles and Sampled Energy Landscapes: Analysis and Comparison*, J. Comput. Chem. 36 (2015) 1213–1231.

[119] W. K. Hastings, *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, Biometrika. 57 (1970) 97–109.

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

[121] I. Newton, *Philosophiæ Naturalis Principia Mathematica*, Innys, 1726.

[122] I. B. Cohen, A. Whitman, J. Budenz, *The Principia: Mathematical Principles of Natural Philosophy*, 1st Edition, University of California Press, 1999.

[123] R. W. Hockney, *The Potential Calculation and Some Applications*, Methods Comput. Phys., 1970.

[124] B. J. Leimkuhler, S. Reich, R. D. Skeel, *Integration Methods for Molecular Dynamics*, Springer New York, 1996, Ch. 10, pp. 161–185.

[125] H. J. C. Berendsen, J. P. M. Postma, W. F. vanGunsteren, A. DiNola, J. R. Haak, *Molecular Dynamics with Coupling to an External Bath*, J. Chem. Phys. 81 (1984) 3684.

[126] S. Nosé, *A Molecular Dynamics Method for Simulations in the Canonical Ensemble*, Mol. Phys. 52 (1984) 255–268.

[127] S. Nosé, *A Unified Formulation of the Constant Temperature Molecular Dynamics Methods*, J. Chem. Phys. 81 (1984) 511–519.

[128] W. G. Hoover, *Canonical Dynamics: Equilibrium Phase-Space Distributions*, Phys. Rev. A. 31 (1985) 1695–1697.

[129] G. J. Martyna, M. L. Klein, M. Tuckerman, *Nosé-Hoover Chains: The Canonical Ensemble via Continuous Dynamics*, J. Chem. Phys. 97 (1992) 2635–2643.

[130] H. von Helmholtz, *Die Thermodynamik Chemischer Vorgänge*, Sitz. K. Akad. Wiss. Berlin, 1882.

[131] M. Parrinello, A. Rahman, *Polymorphic Transitions in Single Crystals: A New Molecular Dynamics Method*, Int. J. Appl. Phys. 52 (1981) 7182–7190.

[132] S. Nosé, M. Klein, *Constant Pressure Molecular Dynamics for Molecular Systems*, Mol. Phys. 50 (1983) 1055–1076.

[133] J. W. Gibbs, *On the Equilibrium of Heterogeneous Substances*, Trans. Conn. Acad. Art. Sci. 3 (1879) 108–248.

[134] P. Kollman, *Free Energy Calculations: Applications to Chemical and Biochemical Phenomena*, Chem. Rev. 93 (1993) 2395–2417.

[135] K. A. Armacost, S. Riniker, Z. Cournia, *Novel Directions in Free Energy Methods and Applications*, J. Chem. Inf. Model. 60 (2020) 1–5.

[136] C. D. Christ, A. E. Mark, W. F. vanGunsteren, *Basic Ingredients of Free Energy Calculations: A Review*, J. Comput. Chem. 31 (2009) 1569–1582.

[137] N. Hansen, W. F. vanGunsteren, *Practical Aspects of Free-Energy Calculations: A Review*, J. Chem. Theory Comput. 10 (2014) 2632–2647.

[138] L. Boltzmann, *Weitere Studien über Das Wärmegleichgewicht Unter Gasmolekülen*, Sitzungsber. Kais. Akad. Wi. Wien. Math. Naturwiss. Cl. 66 (1872) 275–370.

[139] F. M. Ytreberg, D. M. Zuckerman, *Simple Estimation of Absolute Free Energies for Biomolecules*, J. Chem. Phys. 124 (2006) 104105– 1 – 9.

[140] J. G. Kirkwood, *Statistical Mechanics of Fluid Mixtures*, J. Chem. Phys. 3 (1935) 300–313.

[141] R. W. Zwanzig, *High-Temperature Equation of State by a Pertubation Method. I. Nonpolar Gases*, J. Chem. Phys. 22 (1954) 1420–1426.

[142] M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*, 2nd Edition, OUP Oxford, 2017.

[143] M. J. Abraham, T. Murtola, R. Schulz, S. Pall, J. C. Smith, B. Hess, E. Lindahl, *Gromacs: High Performance Molecular Simulations Through Multi-Level Parallelism from Laptops to Supercomputers*, SoftwareX. 1-2 (2015) 19–25.

[144] P. Jupyter, M. Bussonnier, J. Forde, J. Freeman, B. Granger, T. Head, C. Holdgraf, K. Kelley, G. Nalvarte, A. Osheroff, M. Pacer, Y. Panda, F. Perez, B. Ragan-Kelley, C. Willing, *Binder 2.0 - Reproducible, Interactive, Sharable Environments for Science at Scale* , in: Proceedings of the 17th Python in Science Conference, 2018, pp. 113 – 120.

[145] E. Bisong, *Google Colaboratory*, Apress, 2019, Ch. 7, pp. 59–64.

[146] T. Huber, A. E. Torda, W. F. vanGunsteren, *Local Elevation: A Method for Improving the Searching Properties of Molecular Dynamics Simulation*, J. Comput. Aided Mol. Des. 8 (1994) 695–708.

[147] A. Laio, M. Parrinello, *Escaping Free-Energy Minima*, Proceed. Natl. Acad. Sci. USA. 20 (2002) 12562–12566.

[148] C. D. Christ, W. F. vanGunsteren, *Enveloping Distribution Sampling: A Method to Calculate Free Energy Differences from a Single Simulation*, J. Chem. Phys. 126 (2007) 184110.

[149] G. König, S. Boresch, *Non-Boltzmann Sampling and Bennett's Acceptance Ratio Method: How to Profit from Bending the Rules*, J. Comput. Chem. 32 (2012) 1082–1090.

[150] G. König, N. Glaser, B. Schroeder, A. Kubincová, P. H. Hñënberger, S. Riniker, *An Alternative to Conventional λ-Intermediate States in Alchemical Free Energy Calculations: λ-Enveloping Distribution Sampling*, J. Chem. Inf. Model. 60 (2020) 5407–5423.

[151] S. Donnini, R. T. Ullmann, G. Groenhof, H. Grubmüller, *Charge-Neutral Constant pH molecular Dynamics Simula-*

tions Using a Parsimonious Proton Buffer, J. Chem. Theory Comput. 12 (2016) 1040–1051.

[152] R. G. Weiß, P. Setny, J. Dzubiella, *Solvent Fluctuations Induce Non-Markovian Kinetics in Hydrophobic Pocket-Ligand Binding*, J. Phys. Chem. B. 120 (2016) 8127–8136.

[153] O. Lemke, B. G. Keller, *Common Nearest Neighbor Clustering – A Benchmark*, Algorithms. 11 (2018) 19.

[154] R. D. Peng, *Reproducible Research in Computational Science*, Science. 334 (2011) 1226–1228.

[155] V. Stodden, M. McNutt, D. H. Bailey, E. Deelman, Y. Gil, B. Hanson, M. A. Heroux, J. P. Ioannidis, M. Taufer, *Enhancing Reproducibility for Computational Methods*, Science. 354 (2016) 1240–1241.

[156] S. Chacon, B. Straub, *Pro Git*, Springer Nature, 2014.

[157] L. N. Naden, D. G. A. Smith, *Cookiecutter for Computational Molecular Sciences (CMS) Python Packages*(2018).

[158] G. vanRossum, F. L. Drake, *Python 3 Reference Manual*(2009).

[159] B. Stroustrup, *The C++ Programming Language*, 4th Edition, Addision-Wesley, 1995.

[160] E. Lindahl, B. Hess, D. vander Spoel, *GROMACS 3.0: A Package for Molecular Simulation and Trajectory Analysis*, J. Mol. Model. 7 (2001) 306–317.

[161] D. vander Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, H. J. Berendsen, *GROMACS: Fast, Flexible, and Free* , J. Comput. Chem. 26 (2005) 1701–1718.

[162] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L. P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, V. S. Pande, *OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics*, PLoS Comput. Biol. 13 (2017) e1005659.

[163] B. R. Brooks, C. L. Brooks III, A. D. Mackerell Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, M. Karplus, *CHARMM: The Biomolecular Simulation Program*, J. Comput. Chem. 30 (2009) 1545–1614.

[164] W. F. vanGunsteren, H. J. Berendsen, *A leap-Frog Algorithm for Stochastic Dynamics*, Mol. Sim. 1 (1988) 173–185.

[165] A. Brünger, C. L. Brooks, M. Karplus, *Stochastic Boundary Conditions for Molecular Dynamics Simulations of ST2 Water*, Chem. Phys. Lett. 105 (1984) 495–500.

[166] G. M. Torrie, J. P. Valleau, *Nonphysical Sampling Distributions in Monte Carlo Free-Energy Estimation: Umbrella Sampling*, J. Comput. Phys. 23 (1977) 187–199.

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

[168] C. H. Bennett, *Efficient Estimation of Free Energy Differences from Monte Carlo Data*, J. Comput. Phys. 22 (1976) 245–268.

[169] C. D. Christ, W. F. vanGunsteren, *Multiple Free Energies from a Single Simulation: Extending Enveloping Distribution Sampling to Nonoverlapping Phase-Space Distributions*, J. Chem. Phys. 128 (2008) 174112.

[170] D. Sidler, A. Schwaninger, S. Riniker, *Replica Exchange Enveloping Distribution Sampling (RE-EDS): A Robust Method to Estimate Multiple Free-Energy Differences from a Single Simulation*, J. Chem. Phys. 145 (2016) 154114.

[171] D. F. Hahn, P. H. Hünenberger, *Alchemical Free-Energy Calculations by Multiple-Replica -Dynamics: The Conveyor Belt Thermodynamic Integration Scheme*, J. Chem. Theory Comput. 15 (2019) 2392–2419.

[172] A. Pohorille, C. Jarzynski, C. Chipot, *Good Practices in Free-Energy Calculations*, J. Phys. Chem. B. 114 (2010) 10235–10253.

[173] T. Kluyver, B. Ragan-kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, J. Development Team, *Jupyter Notebooks – A Publishing Format for Reproducible Computational Workflows*, Elpub. - (2016) 87–90.

[174] Github, *GitHub* (2020).
URL `https://github.com/`

[175] H. Krekel, B. Oliveira, R. Pfannschmidt, F. Bruynooghe, B. Laugher, F. Bruhin, E. Al, *Pytest: Helps You Write Better Programs*(2004).

[176] G. Brandl, *Sphinx Documentation Tool*(2008).

[177] Github, *Github Actions* (2007).
URL `https://docs.github.com/en/actions/reference`

[178] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. vander Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. vanMulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G. L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodrǵuez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, Y. Vázquez-Baeza, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nat. Method. 17 (2020) 261–272.

[179] S. vanDer Walt, S. C. Colbert, G. Varoquaux, *The NumPy Array: A Structure for Efficient Numerical Computation*, Comput. Sci. Eng. 13 (2011) 22–30.

[180] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz, *SymPy: Symbolic Computing in Python*, Peerj. Comput. Sci. 3 (2017) 103.

[181] W. McKinney, *Data Structures for Statistical Computing in Python*, Proc. 9th. Python. Sci. Conf. 445 (2010) 51–56.

[182] J. D. Hunter, *Matplotlib: A 2D Graphics Environment*, Comput. Sci. Eng. 9 (2007) 99–104.

[183] T. E. Cheatham, J. L. Miller, T. Fox, T. A. Darden, P. A. Kollman, *Molecular Dynamics Simulations on Solvated Biomolecular Systems: The Particle Mesh Ewald Method Leads to Stable Trajectories of DNA, RNA, and Proteins*, J. Am. Chem. Soc. 117 (1995) 4193–4194.

[184] Y. Sugita, A. Kitao, Y. Okamoto, *Multidimensional Replica-Exchange Method for Free-Energy Calculations*, J. Chem. Phys. 113 (2000) 6042–6051.

[185] M. Yamauchi, H. Okumura, *Development of Isothermal-Isobaric Replica-Permutation Method for Molecular Dynamics and Monte Carlo Simulations and Its Application to Reveal Temperature and Pressure Dependence of Folded, Misfolded, and Unfolded States of Chignolin*, J. Chem. Phys. 147 (2017) 184107.

[186] H. C. Andersen, *Molecular Dynamics Simulations at Constant Pressure and/or Temperature*, J. Chem. Phys. 72 (1980) 2384–2393.

[187] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, *Scikit-Learn: Machine Learning in Python*, J. Mach. Learn. Re. 12 (2011) 2825–2830.

[188] G. König, B. R. Brooks, W. Thiel, D. M. York, *On the Convergence of Multi-Scale Free Energy Simulations*, Mol. Sim. 44 (2018) 1062–1081.

[189] J. P. Valleau, D. N. Card, *Monte Carlo Estimation of the Free Energy by Multistage Sampling*, J. Chem. Phys. 57 (1972) 5457–5462.

[190] T. P. Straatsma, J. A. McCammon, *Multiconfiguration Thermodynamic Integration*, J. Chem. Phys. 95 (1991) 1175–1188.

[191] D. Sidler, M. Cristôfol-Clough, S. Riniker, *Efficient Round-Trip Time Optimization for Replica-Exchange Enveloping Distribution Sampling (RE-EDS)*, J. Chem. Theory Comput. 13 (2017) 3020–3030.

[192] F. Schiller, *Wilhelm Tell*, H. Holt, 1898.

[193] G. Heinzelmann, M. Gilson, *Automation of Absolute Protein-Ligand Binding Free Energy Calculations for Docking Refinement and Compound Evaluation*, Sci. Rep. 11 (2021) 1116.

[194] V. Gapsys, L. Pérez-Benito, M. Aldeghi, D. Seeliger, H. van-Vlijmen, G. Tresadern, B. L. de Groot, *Large Scale Relative Protein Ligand Binding Affinities Using Non-Equilibrium Alchemy*, Chem. Sci. 11 (2020) 1140–1152.

[195] W. Jespers, M. Esguerra, J. Åqvist, H. Gutiérrez-De-Terán, *QligFEP: An Automated Workflow for Small Molecule Free Energy Calculations in Q*, J. Cheminf. 11 (2019) 26.

[196] E. P. Raman, T. J. Paul, R. L. Hayes, C. L. Brooks, *Automated, Accurate, and Scalable Relative Protein- Ligand Binding Free-Energy Calculations Using λ-Dynamics*, J. Chem. Theory Comput. 16 (2020) 7895–7914.

[197] Y.-D. Gao, Y. Hu, A. Crespo, D. Wang, K. A. Armacost, J. I. Fells, X. Fradera, H. Wang, H. Wang, B. Sherborne, A. Verras, Z. Peng, *Workflows and Performances in the Ranking Prediction of 2016 D3R Grand Challenge 2: Lessons Learned from a Collaborative Effort*, J. Comput. Aided Mol. Des. 32 (2018) 129–142.

[198] N. Tielker, L. Eberlein, O. Beckstein, S. Güssregen, B. I. Iorga, S. M. Kast, S. Liu, *Perspective on the SAMPL and D3R Blind Prediction Challenges for Physics-Based Free Energy Methods*, ACS Publications, 2021, Ch. 3, pp. 67–107.

[199] H. H. Loeffler, S. Bosisio, G. Duarte Ramos Matos, D. Suh, B. Roux, D. L. Mobley, J. Michel, *Reproducibility of Free Energy Calculations Across Different Molecular Simulation Software Packages*, J. Chem. Theory Comput. 14 (2018) 5567–5582.

[200] S. Riniker, C. D. Christ, N. Hansen, A. E. Mark, P. C. Nair, W. F. vanGunsteren, *Comparison of Enveloping Dis-*

tribution Sampling and Thermodynamic Integration to Calculate Binding Free Energies of Phenylethanolamine N-Methyltransferase Inhibitors, J. Chem. Phys. 135 (2011) 24105.

[201] L. Wang, Y. Wu, Y. Deng, B. Kim, L. Pierce, G. Krilov, D. Lupyan, S. Robinson, M. K. Dahlgren, J. Greenwood, D. L. Romero, C. Masse, J. L. Knight, T. Steinbrecher, T. Beuming, W. Damm, E. Harder, W. Sherman, M. Brewer, R. Wester, M. Murcko, L. Frye, R. Farid, T. Lin, D. L. Mobley, W. L. Jorgensen, B. J. Berne, R. A. Friesner, R. Abel, *Accurate and Reliable Prediction of Relative Ligand Binding Potency in Prospective Drug Discovery by Way of a Modern Free-Energy Calculation Protocol and Force Field*, J. Am. Chem. Soc. 137 (2015) 2695–2703.

[202] L. Wang, Y. Deng, Y. Wu, B. Kim, D. N. LeBard, D. Wandschneider, M. Beachy, R. A. Friesner, R. Abel, *Accurate Modeling of Scaffold Hopping Transformations in Drug Discovery*, J. Chem. Theory Comput. 13 (2017) 42–54.

[203] H. S. Yu, Y. Deng, Y. Wu, D. Sindhikara, A. R. Rask, T. Kimura, R. Abel, L. Wang, *Accurate and Reliable Prediction of the Binding Affinities of Macrocycles to Their Protein Targets*, J. Chem. Theory Comput. 13 (2017) 6290–6300.

[204] J. Wei, C. Chipot, B. Roux, *Computing Relative Binding Affinity of Ligands to Receptor: An Effective Hybrid Single-Dual-Topology Free-Energy Perturbation Approach in NAMD*, J. Chem. Inf. Model. 59 (2019) 3794–3802.

[205] J. L. Paulsen, H. S. Yu, D. Sindhikara, L. Wang, T. Appleby, A. G. Villasenor, U. Schmitz, D. Shivakumar, *Evaluation of*

Free Energy Calculations for the Prioritization of Macrocycle Synthesis, J. Chem. Inf. Model. 60 (2020) 3489–3498.

[206] S. Shobana, B. Roux, O. S. Andersen, *Free Energy Simulations: Thermodynamic Reversibility and Variability*, J. Phys. Chem. B. 104 (2000) 5179–5190.

[207] N. S. Bieler, P. H. Hünenberger, *Orthogonal Sampling in Free-Energy Calculations of Residue Mutations in a Tripeptide: TI Versus λ-LEUS*, J. Chem. Theory Comput. 36 (2015) 1686–1697.

[208] W. Jespers, G. V. Isaksen, T. A. Andberg, S. Vasile, A. van-Veen, J. Aqvist, B. O. Brandsdal, H. Gutiérrez-de Terñ, *QresFEP: An Automated Protocol for Free Energy Calculations of Protein Mutations in Q*, J. Chem. Theory Comput. 15 (2019) 5461–5473.

[209] D. A. Pearlman, P. A. Kollman, *The Overlooked Bond-Stretching Contribution in Free Energy Perturbation Calculations*, J. Chem. Phys. 94 (1991) 4532–4545.

[210] D. A. Pearlman, *A comparison of Alternative Approaches to Free Energy Calculations*, J. Phys. Chem. 98 (1994) 1487–1493.

[211] J. Gao, K. Kuczera, B. Tidor, M. Karplus, *Hidden Thermodynamics of Mutant Proteins: A Molecular Dynamics Analysis*, Science. 244 (1989) 1069–1072.

[212] S. Boresch, M. Karplus, *The Role of Bonded Terms in Free Energy Simulations. 2. Calculation of Their Influence on Free Energy Differences of Solvation*, J. Phys. Chem. A. 103 (1999) 119–136.

[213] G. J. Rocklin, D. L. Mobley, K. A. Dill, *Separated Topologies - a Method for Relative Binding Free Energy Calculations Using Orientational Restraints*, J. Chem. Phys. 138 (2013) 085104.

[214] M. Fleck, M. Wieder, S. Boresch, *Dummy Atoms in Alchemical Free Energy Calculations*, J. Chem. Theory Comp. 17 (2021) 4403–4419.

[215] H. H. Loeffler, J. Michel, C. Woods, *FESetup: Automating Setup for Alchemical Free Energy Simulations*, J. Chem. Inf. Model. 55 (2015) 2485–2490.

[216] M. Suruzhon, T. Senapathi, M. S. Bodnarchuk, R. Viner, I. D. Wall, C. B. Barnett, K. J. Naidoo, J. W. Essex, *ProtoCaller: Robust Automation of Binding Free Energy Calculations*, J. Chem. Inf. Model. 60 (2020) 1917–1921.

[217] D. Petrov, *Perturbation Free-Energy Toolkit: An Automated Alchemical Topology Builder*, J. Chem. Inf. Model. 61 (2021) 4382–4390.

[218] S. Liu, Y. Wu, T. Lin, R. Abel, J. P. Redmann, C. M. Summa, V. R. Jaber, N. M. Lim, D. L. Mobley, *Lead Optimization Mapper: Automating Free Energy Calculations for Lead Optimization*, J. Comput. Aided Mol. Des. 27 (2013) 755–770.

[219] L. Carvalho Martins, E. A. Cino, R. S. Ferreira, *PyAutoFep: An Automated Free Energy Perturbation Workflow for GroMacS integrating Enhanced Sampling Methods*, J. Chem. Theory Comp. 17 (2021) 4262–4273.

[220] N. Homeyer, H. Gohlke, *FEW: A workflow Tool for Free*

Energy Calculations of Ligand Binding, J. Comput. Chem. 34 (2013) 965–973.

[221] J. L. Knight, C. L. Brooks III, *Multisite λ-Dynamics for Simulated Structure–Activity Relationship Studies*, J. Chem. Theory Comput. 7 (2011) 2728–2739.

[222] L. Schrodinger, W. DeLano, *PyMOL* (2020).
URL `http://www.pymol.org/pymol`

[223] W. L. Jorgensen, C. Ravimohan, *Monte Carlo Simulation of Differences in Free Energies of Hydration*, J. Chem. Phys. 83 (1985) 3050–3054.

[224] S. Boresch, M. Karplus, *The Role of Bonded Terms in Free Energy Simulations: 1. Theoretical Analysis*, J. Phys. Chem. A. 103 (1999) 103–118.

[225] S. Donnini, F. Tegeler, G. Groenhof, H. Grubmüller, *Constant pH molecular Dynamics in Explicit Solvent with λ-Dynamics*, J. Chem. Theory Comput. 7 (2011) 1962–1978.

[226] S. Liu, L. Wang, D. L. Mobley, *Is Ring Breaking Feasible in Relative Binding Free Energy Calculations?*, J. Chem. Inf. Model. 55 (2015) 727–735.

[227] K. V. Damodaran, S. Banba, C. L. Brooks, *Application of Multiple Topology λ-Dynamics to a Host-Guest System: β-Cyclodextrin with Substituted Benzenes*, J. Phys. Chem. 105 (2001) 9316–9322.

[228] M. A. Eriksson, L. Nilsson, *Structure, Thermodynamics and Cooperativity of the Glucocorticoid Receptor Dna-binding Domain in Complex with Different Response Elements. Molecular Dynamics Simulation and Free Energy Perturbation Studies*, J. Mol. Biol. 253 (1995) 453–472.

[229] V. Gapsys, S. Michielssens, D. Seeliger, B. L. de Groot, *PMX: Automated Protein Structure and Topology Generation for Alchemical Perturbations*, J. Comput. Chem. 36 (2015) 348–354.

[230] D. Seeliger, B. L. de Groot, *Protein Thermostability Calculations Using Alchemical Free Energy Simulations*, Biophys. J. 98 (2010) 2309–2316.

[231] J. Michel, J. W. Essex, *Prediction of Protein–Ligand Binding Affinity by Free Energy Simulations: Assumptions, Pitfalls and Expectations*, J. Comput. Aided Mol. Des. 24 (2010) 639–658.

[232] C. D. Christ, W. F. vanGunsteren, *Simple, Efficient, and Reliable Computation of Multiple Free Energy Differences from a Single Simulation: A Reference Hamiltonian Parameter Update Scheme for Enveloping Distribution Sampling (EDS)*, J. Chem. Theory Comput. 5 (2009) 276–286.

[233] D. L. Mobley, J. D. Chodera, K. A. Dill, *On the Use of Orientational Restraints and Symmetry Corrections in Alchemical Free Energy Calculations*, J. Chem. Phys. 125 (2006) 084902.

[234] J. Hénin, C. Chipot, *Overcoming Free Energy Barriers Using Unconstrained Molecular Dynamics Simulations*, J. Chem. Phys. 121 (2004) 2904–2914.

[235] G. Landrum, P. Tosco, B. Kelley, S. Riniker, Ric, gedeck, R. Vianello, N. Schneider, A. Dalke, D. N, B. Cole, M. Swain, S. Turk, D. Cosgrove, A. Savelyev, A. Vaucher, M. Wójcikowski, G. Jones, D. Probst, V. F. Scalfani, G. Godin,

A. Pahl, F. Berenger, J. L. Varjo, strets123, JP, Doliath-Gavid, G. Sforna, J. H. Jensen, *Rdkit/Rdkit: 2020_09_5 (Q3 2020) Release*(Mar 2021).

[236] J. von Neumann, *Zur Theorie der Gesellschaftsspiele*, Math. Ann. 100 (1928) 295–320.

[237] R. C. Prim, *Shortest Connection Networks and Some Generalizations*, Bell. Syst. Tech. J. 36 (1957) 1389–1401.

[238] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, D. Vrgoč, *Foundations of JSON schema*, in: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 263–273.

[239] B. Ries, K. Normak, R. G. Weiß, S. Rieder, C. Candide, G. König, S. Riniker, *Relative Free-Energy Calculations for Scaffold Hopping-Type Transformations with an Automated RE-EDS Sampling Procedure*, J. Comput. Aided Mol. Des. - (2022) in press.

[240] N. S. Bieler, J. P. Tschopp, P. H. Hünenberger, *Multistate λ-Local-Elevation Umbrella-Sampling (MS-λ-LEUS): Method and Application to the Complexation of Cations by Crown Ethers*, J. Chem. Theory Comput. 11 (2015) 2575–2588.

[241] J. B. Kruskal, *On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem*, Proc. Amer. Math. Soc. 7 (1956) 48–48.

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

[243] R. Wolfenden, Y. L. Liang, M. Matthews, R. Williams, *Cooperativity and Anticooperativity in Solvation by Water: Imidazoles, Quinones, Nitrophenols, Nitrophenolate, and Nitrothiophenolate Ions*, J. Am. Chem. Soc. 109 (1987) 463–466.

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

[245] A. Nicholls, D. L. Mobley, J. P. Guthrie, J. D. Chodera, C. I. Bayly, M. D. Cooper, V. S. Pande, *Predicting Small-Molecule Solvation Free Energies: An Informal Blind Test for Computational Chemistry*, J. Med. Chem. 51 (2008) 769–779.

[246] J. P. Guthrie, *A Blind Challenge for Computational Solvation Free Energies: Introduction and Overview*, J. Phys. Chem. B. 113 (2009) 4501–4507.

[247] J. P. Guthrie, *SAMPL4, a Blind Challenge for Computational Solvation Free Energies: The Compounds Considered*, J. Comput. Aided 28 (2014) 151–168.

[248] D. L. Mobley, J. P. Guthrie, *FreeSolv: A Database of Experimental and Calculated Hydration Free Energies, with Input Files*, J. Comput. Aided Mol. Des. 28 (2014) 711–720.

[249] M. T. Lehner, B. Ries, S. Rieder, S. Riniker, *rinikerlab/pygromostools: PyGromosTools_V2 (V2.0)*(Mar 2021). doi:10.5281/zenodo.4621710.

[250] H. J. C. Berendsen, J. P. M. Postma, W. F. vanGunsteren, J. Hermans, *Interaction Models for Water in Relation to Protein Hydration*, Reidel, 1981, Ch. 14, pp. 331–342.

[251] T. N. Heinz, W. F. vanGunsteren, P. H. Hünenberger, *Comparison of Four Methods to Compute the Dielectric Permittivity of Liquids from Molecular Dynamics Simulations*, J. Chem. Phys. 115 (2001) 1125.

[252] A. P. Eichenberger, J. R. Allison, J. Dolenc, D. P. Geerke, B. A. C. Horta, K. Meier, C. Oostenbrink, N. Schmid, D. Steiner, D. Wang, W. F. vanGunsteren, *GROMOS++ Software for the Analysis of Biomolecular Simulation Trajectories*, J. Chem. Theory Comput. 7 (2011) 3379.

[253] F. Johansson, et al., *Mpmath: A Python Library for Arbitrary-Precision Floating-Point Arithmetic (Version 0.18)*, http://mpmath.org/ (December 2013).

[254] J. D. Chodera, D. L. Mobley, *Entropy-Enthalpy Compensation: Role and Ramifications in Biomolecular Ligand Recognition and Design*, Annu. Rev. Biophys. 42 (2013) 121–142.

[255] W. L. Jorgensen, J. K. Buckner, S. Boudon, J. Tirado-Rives, *Efficient Computation of Absolute Free Energies of Binding by Computer Simulations. Application to the Methane Dimer in Water*, J. Chem. Phys. 89 (1988) 3742–3746.

[256] K. M. Merz, *Carbon Dioxide Binding to Human Carbonic Anhydrase II*, J. Am. Chem. Soc. 113 (1991) 406–411.

[257] Q. Yang, W. Burchett, G. S. Steeno, S. Liu, M. Yang, D. L. Mobley, X. Hou, *Optimal Designs for Pairwise Calculation:*

An Application to Free Energy Perturbation in Minimizing Prediction Variability, J. Comput. Chem. 41 (2020) 247–257.

[258] J. Lee, B. T. Miller, A. Damjanović, B. R. Brooks, *Constant pH Molecular Dynamics in Explicit Solvent with Enveloping Distribution Sampling and Hamiltonian Exchange*, J. Chem. Theory Comput. 10 (2014) 2738–2750.

[259] J. W. Perthold, C. Oostenbrink, *Accelerated Enveloping Distribution Sampling: Enabling Sampling of Multiple End States While Preserving Local Energy Minima*, J. Phys. Chem. B. 122 (2018) 5030–5037.

[260] J. W. Perthold, D. Petrov, C. Oostenbrink, *Toward Automated Free Energy Calculation with Accelerated Enveloping Distribution Sampling (A-EDS)*, J. Chem. Inf. Model. 60 (2020) 5395–5406.

[261] X. Huang, C. C. Cheng, T. O. Fischmann, J. S. Duca, X. Yang, M. Richards, G. W. Shipps, *Discovery of a Novel Series of CHK1 Kinase Inhibitors with a Distinctive Hinge Binding Mode*, ACS Med. Chem. Lett. 3 (2012) 123–128.

[262] N. Hansen, J. Dolenc, M. Knecht, S. Riniker, W. F. van-Gunsteren, *Assessment of Enveloping Distribution Sampling to Calculate Relative Free Enthalpies of Binding for Eight Netropsin-DNA Duplex Complexes in Aqueous Solution*, J. Comput. Chem. 33 (2012) 640–651.

[263] B. Ries, S. M. Linker, D. F. Hahn, G. König, S. Riniker, *Ensembler: A Simple Package for Fast Prototyping and Teaching Molecular Simulations*, J. Chem. Inf. Model. 61 (2021) 560–564.

[264] J. Lee, B. T. Miller, A. Damjanović, B. R. Brooks, *Enhancing Constant-pH Simulation in Explicit Solvent with a Two-Dimensional Replica Exchange Method*, J. Chem. Theory Comput. 11 (2015) 2560–2574.

[265] H. G. Katzgraber, S. Trebst, D. A. Huse, M. Troyer, *Feedback-Optimized Parallel Tempering Monte Carlo*, J. Stat. Mech. 2006 (2006) P03018–P03018.

[266] W. Nadler, J. H. Meinke, U. H. Hansmann, *Folding Proteins by First-Passage-Times-Optimized Replica Exchange* , Phys. Rev. 8 (2008) 061905.

[267] M. M. H. Graf, M. Maurer, C. Oostenbrink, *Free-Energy Calculations of Residue Mutations in a Tripeptide Using Various Methods to Overcome Inefficient Sampling*, J. Comp. Chem. 37 (2016) 2597–2605.

[268] D. F. Hahn, G. König, P. H. Hünenberger, *Overcoming Orthogonal Barriers in Alchemical Free Energy Calculations: On the Relative Merits of λ-Variations, λ-Extrapolations, and Biasing*, J. Chem. Theory Comput. 16 (2020) 1630–1645.

[269] A. K. Malde, L. Zuo, M. Breeze, M. Stroet, D. Poger, P. C. Nair, C. Oostenbrink, A. E. Mark, *An Automated Force Field Topology Builder (ATB) and Repository: Version 1.0*, J. Chem. Theory Comput. 7 (2011) 4026–4037.

[270] P. Bleiziffer, K. Schaller, S. Riniker, *Machine Learning of Partial Charges Derived from High-Quality Quantum-Mechanical Calculations*, J. Chem. Inf. Model. 58 (2018) 579–590.

[271] S. Ruder, *An Overview of Gradient Descent Optimization Algorithms*, ArXiv. Prepr. abs/1609.04747 (2016) arXiv:1609.04747.

[272] A. Glättli, X. Daura, W. F. vanGunsteren, *Derivation of an Improved Simple Point Charge Model for Liquid Water: SPC/A and SPC/L*, J. Chem. Phys. 116 (2002) 9811–9828.

[273] C. E. M. Schindler, H. Baumann, A. Blum, D. Böse, H.-P. Buchstaller, L. Burgdorf, D. Cappel, E. Chekler, P. Czodrowski, D. Dorsch, M. K. I. Eguida, B. Follows, T. Fuchs, U. Grđlër, J. Gunera, T. Johnson, C. Jorand Lebrun, S. Karra, M. Klein, T. Knehans, L. Koetzner, M. Krier, M. Leiendecker, B. Leuthner, L. Li, I. Mochalkin, D. Musil, C. Neagu, F. Rippmann, K. Schiemann, R. Schulz, T. Steinbrecher, E.-M. Tanzer, A. Unzue Lopez, A. Viacava Follis, A. Wegener, D. Kuhn, *Large-Scale Assessment of Binding Free Energy Calculations in Active Drug Discovery Projects*, J. Chem. Inf. Model. 60 (2020) 5457–5474.

[274] E. Post, *Real Programmers Don't Use Pascal*, Datamation. 29 (1983) 263–265.

[275] P. Tuomi, J. Multisilta, P. Saarikoski, J. Suominen, *Coding Skills as a Success Factor for a Society*, Educ. Inf. Technol. 23 (2018) 419–434.

[276] R. Abel, L. Wang, E. D. Harder, B. J. Berne, R. A. Friesner, *Advancing Drug Discovery Through Enhanced Free Energy Calculations*, Acc. Chem. Res. 50 (2017) 1625–1632.

[277] J. Gosling, B. Joy, G. Steele, G. Bracha, *The Java Language Specification*, Addison-Wesley Professional, 2000.

[278] J. W. Backus, R. J. Beeber, S. Best, R. Goldberg, L. M. Haibt, H. L. Herrick, R. A. Nelson, D. Sayre, P. B. Sheridan, H. Stern, I. Ziller, R. A. Hughes, R. Nutt, *The FORTRAN automatic Coding System*, in: Western Joint Computer Conference: Techniques for Reliability, Association for Computing Machinery, 1957, pp. 188–198.

[279] R. Alfayez, C. Chen, P. Behnamghader, K. Srisopha, B. Boehm, *An Empirical Study of Technical Debt in Open-Source Software Systems*, in: Disciplinary Convergence in Systems Engineering Research, Springer International Publishing, 2018, pp. 113–125.

[280] V. M. Ayer, S. Miguez, B. H. Toby, *Why Scientists Should Learn to Program in Python*, Powder. Diffr. J. 29 (2014) S48–S64.

[281] K. J. Millman, M. Aivazis, *Python for Scientists and Engineers*, Comput. Sci. Eng. 13 (2011) 9–12.

[282] S. Cass, *The 2017 Top Programming Languages*(2017).

[283] S. Cass, *The 2018 Top Programming Languages*(2018).

[284] S. Cass, *The 2019 Top Programming Languages*(2019).

[285] S. Cass, *The 2020 Top Programming Languages*(2020).

[286] S. Cass, *The 2021 Top Programming Languages*(2021).

[287] T. E. Oliphant, *Python for Scientific Computing*, Comput. Sci. Eng. 9 (2007) 10–20.

[288] J. Wenzel, J. Rhinelander, D. Moldovan, *Pybind11 – Seamless Operability Between C++11 and Python*(2017).

[289] S. Koranne, *Boost C++ Libraries*, in: Handbook of Open Source Tools, Springer, 2011, pp. 127–143.

[290] D. M. Beazley, *SwiG: An Easy to Use Tool for Integrating Scripting Languages with C and C++*(1996).

[291] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, K. Smith, *Cython: The Best of Both Worlds*, Comput. Sci. . Eng. 13 (2011) 31–39.

[292] S. K. Lam, A. Pitrou, S. Seibert, *Numba: A LLVM-Based Python JIT Compiler*, in: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, Association for Computing Machinery, 2015, pp. 1–6.

[293] *Python Package Index - PypI* (2021).
URL `https://pypi.org/`

[294] *Anaconda Software Distribution* (2020).
URL `https://docs.anaconda.com/`

[295] B. W. Kernighan, D. M. Ritchie, *The C Programming Language*, Prentice-Hall, 1988.

[296] C. Lattner, *LLVM and Clang: Next Generation Compiler Technology*, in: The BSD conference, 2008, pp. 1–33.

[297] W. Jakob, J. Rhinelander, D. Moldovan, *Pybind11 – Seamless Operability Between C++11 and Python*, https://github.com/pybind/pybind11 (2017).

[298] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, G. K.-L. Chan, *PySCF: The Python-Based Simulations of Chemistry Framework*, WIRE. Comput. Mol. Sci. 8 (2018) e1340.

[299] H. L. Röst, U. Schmitt, R. Aebersold, L. Malmström, *pyOpenMs: A python-Based Interface to the OpenMs Mass-Spectrometry Algorithm Library*, Proteomics. 14 (2014) 74–77.

[300] P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, M. J. L. de Hoon, *Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics*, Bioinform. 25 (2009) 1422–1423.

[301] N. M. O' Boyle, C. Morley, G. R. Hutchison, *Pybel: A Python Wrapper for the OpenBabel Cheminformatics Toolkit*, Chem. Central. J. 2 (2008) 5.

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

[303] M. E. Irrgang, J. M. Hays, P. M. Kasson, *GMXapi: A High-Level Interface for Advanced Control and Extension of Molecular Dynamics Simulations*, Bioinformatics. 34 (2018) 3945–3947.

[304] M. R. Shirts, C. Klein, J. M. Swails, J. Yin, M. K. Gilson, D. L. Mobley, D. A. Case, E. D. Zhong, *Lessons Learned from Comparing Molecular Dynamics Engines on the SAMPL5 Dataset*, J. Comput. Mol. 31 (2017) 147–161.

[305] M. S. Friedrichs, P. Eastman, V. Vaidyanathan, M. Houston, S. Legrand, A. L. Beberg, D. L. Ensign, C. M. Bruns, V. S. Pande, *Accelerating Molecular Dynamic Simulation*

on Graphics Processing Units, J. Comput. Chem. 30 (2009) 864–872.

[306] S. Plimpton, *Fast Parallel Algorithms for Short-Range Molecular Dynamics*, J. Comput. Phys. 117 (1995) 1–19.

[307] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in' t Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, S. J. Plimpton, *LAMMPS – A Flexible Simulation Tool for Particle-Based Materials Modeling at the Atomic, Meso, and Continuum Scales*, Comput. Phys. Commun. 271 (2022) 108171.

[308] L. Talirz, L. M. Ghiringhelli, B. Smit, *Trends in Atomistic Simulation Software Usage [Article V1.0]*, Living. J. Comput. Mol. Sci. 3 (2021) 1483.

[309] M. Henning, *API Design Matters*, Commun. ACM. 52 (2009) 46–56.

[310] J. Blanchette, *The Little Manual of API Design*, Trolltech, 2008.

[311] J. Bloch, *How to Design a Good API and Why It Matters*, in: Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications, Association for Computing Machinery, 2006, pp. 506–507.

[312] K. R. M. Leino, G. Nelson, *Data Abstraction and Information Hiding*, ACM. Trans. Program. Lang. Syst. 24 (2002) 491—-553.

[313] P. S. Ganney, S. Pisharody, E. Claridge, *Chapter 9 - Software Engineering*, in: Clinical Engineering, Academic Press, 2020, pp. 131–168.

[314] G. Brandl, *Sphinx Documentation*(2021).

[315] A. C. Kay, *The Early History of Smalltalk*, in: The Second ACM SIGPLAN Conference on History of Programming Languages, Association for Computing Machinery, 1993, pp. 69–95.

[316] H. Curry, R. Feys, R. Hindley, J. P. Seldin, *Combinatory Logic*, North-Holland Publishing Co., 1958.

[317] hdfgroup, *HDF5 Version 1.10.7 Released on 2020-09-16*, an optional note (09 2020).

[318] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, *Extensible Markup Language (XML) 1.0*(2008).

[319] N. Rego, D. Koes, *3Dmol.js: Molecular Visualization with WebGl*, Bioinformatics. 31 (2014) 1322–1324.

[320] B. Lier, C. Öhlknecht, A. de Ruiter, J. Gebhardt, W. F. vanGunsteren, C. Oostenbrink, N. Hansen, *A suite of Advanced Tutorials for the GROMOS Biomolecular Simulation Software [Article V1.0]*, Living. J. Comp. Mol. Sci. 2 (2020) 18552.

[321] W. P. Walters, *Code Sharing in the Open Science Era*, J. Chem. Inf. Model. 60 (2020) 4417–4420.

[322] E. M. Driggers, S. P. Hale, J. Lee, N. K. Terrett, *The Exploration of Macrocycles for Drug Discovery – an Underexploited Structural Class*, Nat. Rev. Drug. Discov. 7 (2008) 608.

[323] J. Mallinson, I. Collins, *Macrocycles in New Drug Discovery*, Future. Med. Chem. 4 (2012) 1409.

[324] B. Doak, B. Over, F. Giordanetto, J. Kihlberg, *Oral Druggable Space Beyond the Rule of 5: Insights from Drugs and Clinical Candidates*, Chem. . Biology. 21 (2014) 1115–1142.

[325] P. G. Dougherty, Z. Qian, D. Pei, *Macrocycles as Protein–Protein Interaction Inhibitors*, Biochem. J. (2017) 1109.

[326] E. Marsault, M. L. Peterson, *Macrocycles Are Great Cycles: Applications, Opportunities, and Challenges of Synthetic Macrocycles in Drug Discovery*, J. Med. Chem. 54 (2011) 1961–2004.

[327] M. A. Abdalla, L. J. McGaw, *Natural Cyclic Peptides as an Attractive Modality for Therapeutics: A mini Review*, Molecules. (2018) 2080.

[328] E. Marsault, M. L. Peterson, *Practical Medicinal Chemistry with Macrocycles: Design, Synthesis, and Case Studies*, 1st Edition, John Wiley & Sons, Inc., 2017.

[329] G. Caron, J. Kihlberg, G. Goetz, E. Ratkova, V. Poongavanam, G. Ermondi, *Steering New Drug Discovery Campaigns: Permeability, Solubility, and Physicochemical Properties in the bRO5 Chemical Space*, ACS Med. Chem. Lett. 12 (2021) 13–23.

[330] P. Chène, *Drugs Targeting Protein-Protein Interactions*, ChemMedChem. 1 (2006) 400.

[331] J. Janin, R. P. Bahadur, P. Chakrabarti, *Protein-Protein Interaction and Quaternary Structure*, Q. Rev. Biophys. 41 (2008) 133.

[332] S. Jones, J. M. Thornton, *Principles of Protein-Protein Interactions*, Proc. National. Acad. Sci. USA 93 (1996) 13–20.

[333] D. E. Scott, A. R. Bayly, C. Abell, J. Skidmore, *Small Molecules, Big Targets: Drug Discovery Faces the Protein-Protein Interaction Challenge*, Nat. Rev. Drug. Discov. 15 (2016) 533.

[334] A. E. Modell, S. L. Blosser, P. S. Arora, *Systematic Targeting of Protein-Protein Interactions Approaches to Targeting Protein-Protein Interactions*, Trends. Pharmacol. Sci. 37 (2016) 702.

[335] A. Zorzi, K. Deyle, C. Heinis, *Cyclic Peptide Therapeutics: Past, Present and Future*, Curr. Opin. Chem. Biology. 38 (2017) 24–29.

[336] F. Giordanetto, J. Kihlberg, *Macrocyclic Drugs and Clinical Candidates: What Can Medicinal Chemists Learn from Their Properties?*, J. Med. Chem. 57 (2014) 278.

[337] K. Fosgerau, T. Hoffmann, *Peptide Therapeutics: Current Status and Future Directions*, Drug. Discov. Today. 20 (2015) 122.

[338] C. K. Wang, S. E. Northfield, B. Colless, S. Chaousis, I. Hamernig, R.-J. Lohman, D. S. Nielsen, C. I. Schroeder, S. Liras, D. A. Price, D. P. Fairlie, D. J. Craik, *Rational Design and Synthesis of an Orally Bioavailable Peptide Guided by NMR Amide Temperature Coefficients*, Proc. Natl. Acad. Sci. USA. 111 (2014) 17504.

[339] D. S. Nielsen, H. N. Hoang, R. J. Lohman, T. A. Hill, A. J. Lucke, D. J. Craik, D. J. Edmonds, D. A. Griffith, C. J. Rotter, R. B. Ruggeri, D. A. Price, S. Liras, D. P. Fairlie,

Improving on Nature: Making a Cyclic Heptapeptide Orally Bioavailable, Angew. Chem. Int. Ed. 53 (2014) 12059.

[340] A. Whitty, M. Zhong, L. Viarengo, D. Beglov, D. R. Hall, S. Vajda, *Quantifying the Chameleonic Properties of Macrocycles and Other High-Molecular-Weight Drugs*, Drug. Discov. Today. 21 (2016) 712–717.

[341] A. F. B. Räder, F. Reichart, M. Weinmüller, H. Kessler, *Improving Oral Bioavailability of Cyclic Peptides by N-Methylation*, Bioorg. Med. Chem. 26 (2018) 2766.

[342] T. R. White, C. M. Renzelman, A. C. Rand, T. Rezai, C. M. McEwen, V. M. Gelev, R. A. Turner, R. G. Linington, S. S. F. Leung, A. S. Kalgutkar, J. N. Bauman, Y. Zhang, S. Liras, D. A. Price, A. M. Mathiowetz, M. P. Jacobson, R. S. Lokey, *On-Resin N-Methylation of Cyclic Peptides for Discovery of Orally Bioavailable Scaffolds*, Nat. Chem. Biol. 7 (2011) 810.

[343] J. G. Beck, J. Chatterjee, B. Laufer, M. U. Kiran, A. O. Frank, S. Neubauer, O. Ovadia, S. Greenberg, C. Gilon, A. Hoffman, H. Kessler, *Intestinal Permeability of Cyclic Peptides: Common Key Backbone Motifs Identified*, J. Am. Chem. Soc. 134 (2012) 12125.

[344] E. Biron, J. Chatterjee, O. Ovadia, D. Langenegger, J. Brueggen, D. Hoyer, H. A. Schmid, R. Jelinek, C. Gilon, A. Hoffman, H. Kessler, *Improving Oral Bioavailability of Peptides by Multiple N-Methylation: Somatostatin Analogues*, Angew. Chem. Int. Ed. 47 (2008) 2595.

[345] J. Schwochert, R. Turner, M. Thang, R. F. Berkeley, A. R. Ponkey, K. M. Rodriguez, S. S. F. Leung, B. Khunte,

G. Goetz, C. Limberakis, A. S. Kalgutkar, H. Eng, M. J. Shapiro, A. M. Mathiowetz, D. A. Price, S. Liras, M. P. Jacobson, R. S. Lokey, *Peptide to Peptoid Substitutions Increase Cell Permeability in Cyclic Hexapeptides*, Org. Lett. 17 (2015) 2928.

[346] Q. Sui, D. Borchardt, D. L. Rabenstein, *Kinetics and Equilibria of Cis/Trans Isomerization of Backbone Amide Bonds in Peptoids*, J. Am. Chem. Soc. 129 (2007) 12042.

[347] S. Riniker, *Toward the Elucidation of the Mechanism for Passive Membrane Permeability of Cyclic Peptides*, Future. Med. Chem. 11 (2019) 637.

[348] M. Tyagi, V. Poongavanam, M. Lindhagen, A. Pettersen, P. Sjö, S. Schiesser, J. Kihlberg, *Toward the Design of Molecular Chameleons: Flexible Shielding of an Amide Bond Enhances Macrocycle Cell Permeability*, Org. Lett. 20 (2018) 5737.

[349] E. Marsault, K. Benakli, S. Beaubien, C. Saint-Louis, R. Déziel, G. Fraser, *Potent Macrocyclic Antagonists to the Motilin Receptor Presenting Novel Unnatural Amino Acids*, Bioorg. Med. Chem. Lett. 17 (2007) 4187.

[350] H. R. Hoveyda, E. Marsault, R. Gagnon, A. P. Mathieu, M. Vézina, A. Landry, Z. Wang, K. Benakli, S. Beaubien, C. Saint-Louis, M. Brassard, J. F. Pinault, L. Ouellet, S. Bhat, M. Ramaseshan, X. Peng, L. Foucher, S. Beauchemin, P. Bhérer, D. F. Veber, M. L. Peterson, G. L. Fraser, *Optimization of the Potency and Pharmacokinetic Properties of a Macrocyclic Ghrelin Receptor Agonist (Part I): Development of Ulimorelin (TZP-101) from Hit to Clinic*, J. Med. Chem. 54 (2011) 8305.

[351] A. Le Roux, E. Blaise, P.-L. Boudreault, C. Comeau, A. Doucet, M. Giarrusso, M.-P. Collin, T. Neubauer, F. Koelling, A. H. Göller, L. Seep, D. T. Tshitenge, M. Wittwer, M. Kullmann, A. Hillisch, J. Mittendorf, É. Marsault, *Structure-Permeability Relationship of Semi-Peptidic Macrocycles – Understanding and Optimizing Passive Permeability and Efflux Ratio*, J. Med. Chem. (2020) 6774.

[352] S. D. Appavoo, S. Huh, D. B. Diaz, A. K. Yudin, *Conformational Control of Macrocycles by Remote Structural Modification*, Chem. Rev. (2019) 9724.

[353] A. T. Bockus, J. A. Schwochert, C. R. Pye, C. E. Townsend, V. Sok, M. A. Bednarek, R. S. Lokey, *Going Out on a Limb: Delineating the Effects of Beta-Branching, N-Methylation, and Side Chain Size on the Passive Permeability, Solubility, and Flexibility of Sanguinamide a Analogues*, J. Med. Chem. 58 (2015) 7409–18.

[354] W. M. Hewitt, S. S. F. Leung, C. R. Pye, A. R. Ponkey, M. Bednarek, M. P. Jacobson, R. S. Lokey, *Cell-Permeable Cyclic Peptides from Synthetic Libraries Inspired by Natural Products*, J. Am. Chem. Soc. 137 (2015) 715–721.

[355] T. Rezai, J. E. Bock, M. V. Zhou, C. Kalyanaraman, R. S. Lokey, M. P. Jacobson, *Conformational Flexibility, Internal Hydrogen Bonding, and Passive Membrane Permeability: Successful in Silico Prediction of the Relative Permeabilities of Cyclic Peptides*, J. Am. Chem. Soc. 128 (2006) 14073–14080.

[356] B. Over, P. Matsson, C. Tyrchan, P. Artursson, B. C. Doak, M. A. Foley, C. Hilgendorf, S. E. Johnston, M. D. Lee, R. J. Lewis, P. McCarren, G. Muncipinto, U. Norinder, M. W. D.

Perry, J. R. Duvall, J. Kihlberg, *Structural and Conformational Determinants of Macrocycle Cell Permeability*, Nat. Chem. Biol. (2016) 1065.

[357] C. Comeau, B. Ries, T. Stadelmann, J. Tremblay, S. Poulet, U. Fröhlich, J. r. Côtè, P.-L. Boudreault, R. M. Derbali, P. Sarret, M. Grandbois, G. . ò. Leclair, S. Riniker, É. Marsault, *Modulation of the Passive Permeability of Semipeptidic Macrocycles: N- And C-Methylations Fine-Tune Conformation and Properties*, J. Med. Chem. 64 (2021) 5365–5383.

[358] G. Ottaviani, S. Martel, P.-A. Carrupt, *Parallel Artificial Membrane Permeability Assay: A new Membrane for the Fast Prediction of Passive Human Skin Permeability*, J. Med. Chem. 49 (2006) 3948–3954.

[359] L. Di, E. Kerns, *Drug-Like Properties: Concepts, Structure Design and Methods from ADME to Toxicity Optimization*, Academic press, 2015.

[360] J. Fogh, J. M. Fogh, T. Orfeo, *One Hundred and Twenty-Seven Cultured Human Tumor Cell Lines Producing Tumors in Nude Mice*, J. Natl. Cancer. Inst. 59 (1977) 221–226.

[361] A. Y. S. Balazs, R. J. Carbajo, N. L. Davies, Y. Dong, A. W. Hird, J. W. Johannes, M. L. Lamb, W. McCoull, P. Raubo, G. R. Robb, M. J. Packer, E. Chiarparin, *Free Ligand 1D NMR Conformational Signatures to Enhance Structure Based Drug Design of a Mcl-1 Inhibitor (AZD5991) and Other Synthetic Macrocycles*, J. Med. Chem. 62 (2019) 9418.

[362] T. Stadelmann, G. Subramanian, S. Menon, C. E. Townsend, R. S. Lokey, M.-O. Ebert, S. Riniker, *Connecting the*

Conformational Behavior of Cyclic Octadepsipeptides with Their Ionophoric Property and Membrane Permeability, Org. Biomol. Chem. 18 (2020) 7110–7126.

[363] M. R. Sebastiano, B. C. Doak, M. Backlund, V. Poongavanam, B. Over, G. Ermondi, G. Caron, J. Kihlberg, *Impact of Dynamically Exposed Polarity on Permeability and Solubility of Chameleonic Drugs Beyond the Rule of 5*, J. Med. Chem. (2018) 4189.

[364] P. C. D. Hawkins, A. Nicholls, *Conformer Generation with OMEGA: Learning from the Data Set and the Analysis of Failures*, J. Chem. Inf. Model. 52 (2012) 2919.

[365] P. C. D. Hawkins, A. G. Skillman, G. L. Warren, B. A. Ellingson, M. T. Stahl, *Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and Cambridge Structural Database*, J. Chem. Inf. Model. 50 (2010) 572.

[366] I. G. Tironi, W. F. vanGunsteren, *A molecular Dynamics Simulation Study of Chloroform*, Mol. Phys. 83 (1994) 381.

[367] W. F. vanGunsteren, H. J. C. Berendsen, *A Leap-Frog Algorithm for Stochastic Dynamics*, Mol. Simul. 1 (1988) 173.

[368] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, F. Noé, *PyEMMA 2: A software Package for Estimation, Validation, and Analysis of Markov Models*, J. Chem. Theory Comput. 11 (2015) 5525.

[369] R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes,

L. P. Wang, T. J. Lane, V. S. Pande, *MDTraj: A modern Open Library for the Analysis of Molecular Dynamics Trajectories*, Biophys. J. 109 (2015) 1528.

[370] L. Molgedey, H. G. Schuster, *Separation of a Mixture of Independent Signals Using Time Delayed Correlations*, Phys. Rev. Lett. 72 (1994) 3634.

[371] B. Keller, X. Daura, W. F. vanGunsteren, *Comparing Geometric and Kinetic Cluster Algorithms for Molecular Simulation Data*, J. Chem. Phys. 132 (2010) 074110.

[372] R. G. Weiß, B. Ries, S. Wang, S. Riniker, *Volume-Scaled Common Nearest Neighbor Clustering Algorithm with Free-Energy Hierarchy*, J. Chem. Phys. 154 (2021) 084106.

[373] B. Vögeli, S. Olsson, R. Riek, P. Güntert, *Compiled Data Set of Exact Noe Distance Limits, Residual Dipolar Couplings and Scalar Couplings for the Protein GB3*, Data. Br. 5 (2015) 99.

[374] A. Alex, D. S. Millan, M. Perez, F. Wakenhut, G. A. Whitlock, *Intramolecular Hydrogen Bonding to Improve Membrane Permeability and Absorption in Beyond Rule of Five Chemical Space*, MedChemComm. 2 (2011) 669.

[375] S. S. Shapiro, M. B. Wilk, *an analysis of variance test for normality*, Biometrika. 52 (1965) 591.

[376] S. Kotz, N. L. Johnson, *Breakthroughs in Statistics, Volume IIi*, Technometrics. 40 (1998) 165.

[377] T. Vorherr, I. Lewis, J. Berghausen, S. Desrayaud, M. Schaefer, *Modulation of Oral Bioavailability and Metabolism for Closely Related Cyclic Hexapeptides*, Int. J. Pept. Res. Ther. 24 (2018) 35.

[378] L. Peraro, J. A. Kritzer, *Emerging Methods and Design Principles for Cell-Penetrant Peptides*, Angew. Chem. Int. Ed. 57 (2018) 11868.

[379] M. Ende, *Die Unendliche Geschichte*, Thienemann Verlag, 2017.

[380] W. P. Walters, *Modeling, Informatics, and the Quest for Reproducibility*, J. Chem. Inf. Model. 53 (2013) 1529–1530.

[381] BioMedCentral, *Editorial Policies*, http://biomedcentral.com/getpublished/editorial-policies, accessed: 2021-12-07 (2021).

[382] ScienceJournals, *Editorial Policies*, https://www.sciencemag.org/authors/science-journals-editorial-policies, accessed: 2021-12-07 (2021).

[383] NeurIPS, *Code Submission Policy*, https://nips.cc/Conferences/2020/PaperInformation/CodeSubmissionPolicy, accessed: 2021-12-07 (2021).

[384] C. Jarzynski, *Nonequilibrium Equality for Free Energy Differences*, Phys. Rev. Lett. 78 (1997) 2690–2693.

[385] H. Xiong, A. Crespo, M. Marti, D. Estrin, A. E. Roitberg, *Free Energy Calculations with Non-Equilibrium Methods: Applications of the Jarzynski Relationship*, Theor. Chem. Acc. 116 (2006) 338–346.

[386] M. R. Shirts, J. D. Chodera, *Statistically Optimal Analysis of Samples from Multiple Equilibrium States*, J. Chem. Phys. 129 (2008) 124105.

[387] G. König, B. Ries, P. H. Hñënberger, S. Riniker, *Efficient Alchemical Intermediate States in Free Energy Calculations*

Using λ-Enveloping Distribution Sampling, J. Chem. Theory Comput. 17 (2021) 5805–5815.

[388] S. Riniker, *Molecular Dynamics Fingerprints (MDFP): Machine Learning from MD Data to Predict Free-Energy Differences*, J. Chem. Inf. Model. 57 (2017) 726–741.

[389] H. Zhao, A. Caflisch, *Molecular Dynamics in Drug Design*, Eur. J. Med. Chem. 91 (2015) 4–14.

[390] J. Eberhardt, D. Santos-Martins, A. F. Tillack, S. Forli, *AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings*, J. Chem. Inf. Model. 61 (2021) 3891–3898.

[391] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, A. J. Olson, *AutoDock4 and AutoDockTools4: Automated Docking with Selective Receptor Flexibility*, J. Comp. Chem. 30 (2009) 2785–2791.

[392] Y.-C. Chen, *Beware of Docking!*, Trends. Pharmacol. Sci. 36 (2015) 78–95.

[393] Z. Feng, L. V. Pearce, X. Xu, X. Yang, P. Yang, P. M. Blumberg, X.-Q. Xie, *Structural Insight Into Tetrameric hTRPV1 from Homology Modeling, Molecular Docking, Molecular Dynamics Simulation, Virtual Screening, and Bioassay Validations*, J. Chem. Inf. Model. 55 (2015) 572–588.

[394] P. Sokkar, S. Mohandass, M. Ramachandran, *Multiple Templates-Based Homology Modeling Enhances Structure Quality of AT1 Receptor: Validation by Molecular Dynamics and Antagonist Docking*, J. Mol. Model. 17 (2011) 1565–1577.

[395] J. Chavda, H. Bhatt, *3D-QSAR (CoMFA, CoMSIA, HQSAR and Topomer CoMFA), MD Simulations and Molecular Docking Studies on Purinylpyridine Derivatives as B-raf Inhibitors for the Treatment of Melanoma Cancer*, Struct. Chem. 30 (2019) 2093–2107.

[396] T. Rezai, B. Yu, G. L. Millhauser, M. P. Jacobson, R. S. Lokey, *Testing the Conformational Hypothesis of Passive Membrane Permeability Using Synthetic Cyclic Peptide Diastereomers*, J. Am. Chem. Soc. 128 (2006) 2510–2511.

[397] P. Matsson, J. Kihlberg, *How Big Is Too Big for Cell Permeability?*, J. Med. Chem. 60 (2017) 1662–1664.

Curriculum Vitæ

Benjamin Joachim Ries

15.03.1991
Ettlingen, Germany
German citizen

Education

2017 – 2022	PhD, ETH Zürich
2014 – 2017	Bioinformatics MSc., Universität Tübingen
20011 – 2014	Biochemistry BSc., Universität Tübingen
2002 – 2011	Abitur, Albertus Magnus Gymnasium, Ettlingen, Germany

Experience

2019	Summerschool, Universita della Svizzera Italiana, Switzerland: Effective High-Performance Computing & Data Analytics with GPUs
2017	Summerschool, University of Jyväskylä, Finland: Measuring and Modelling Proton Equilibria in Complex Macromolecular Systems
2016 – 2017	Erasmus+ internship, Uppsala Universitet, Sweden
2013 – 2014	Lab Assistant, MPI, Developmental Biology, Tübingen

Teaching Assistant

2018	Physical Chemistry I: Thermodynamics (spring semester), F. Merkt
2018, 2019	Algorithms and Programming in C++ (fall semester), S. Riniker
2019, 2020	Physical Chemistry Practicum for Biology and Pharmacy Students: Molecular Dynamics (spring semester), E. Meister
2021	Statistical Physics and Computer Simulation for CSE (spring semester), P. Hünenberger and S. Riniker
2021	Computer Simulation of BioMolecular Systems (fall semester), S. Riniker and P. Hünenberger

Commitment

2019-2021	Young Swiss Chemical Society (youngSCS), ETH Representative (2019-2020), President(2020-2021)
2012-2017	juniorGBM Tübingen, president(2016)