Application and Optimization of Contact-Guided Replica Exchange Molecular Dynamics

Zur Erlangung des akademischen Grads eines

DOKTORS DER NATURWISSENSCHAFTEN (Dr. rer. nat.)

von der KIT-Fakultät für Physik des
Karlsruher Instituts für Technologie (KIT)
angenommene

DISSERTATION

von

Arthur Voronin, M.Sc.

Tag der mündlichen Prüfung: 13.05.2022
Erstgutachter: Prof. Dr. Wolfgang Wenzel
Zweitgutachter: Prof. Dr. Alexander Schug

Zusammenfassung

Proteine sind komplexe Makromoleküle, die in lebenden Organismen eine große Vielfalt an wichtigen Aufgaben erfüllen. Proteine können beispielsweise Gene regulieren, Struktur stabilisieren, Zellsignale übertragen, Substanzen transportieren und vieles mehr. Typischerweise sind umfassende Kenntnisse von Struktur und Dynamik eines Proteins erforderlich um dessen physiologische Funktion und Interaktionsmechanismen vollständig zu verstehen. Gewonnene Erkenntnisse sind für Biowissenschaften unerlässlich und können auf viele Bereiche angewendet werden, wie z.B. für Arzneimitteldesign oder zur Krankheitsbehandlung. Trotz des unfassbaren Fortschritts experimenteller Techniken bleibt die Bestimmung einer Proteinstruktur immer noch eine herausfordernde Aufgabe. Außerdem können Experimente nur Teilinformationen liefern und Messdaten können mehrdeutig und schwer zu interpretieren sein. Aus diesem Grund werden häufig Computersimulationen durchgeführt um weitere Erkenntnisse zu liefern und die Lücke zwischen Theorie und Experiment zu schließen. Heute sind viele *in-silico* Methoden in der Lage genaue Protein Strukturmodelle zu erzeugen, sei es mit einem *de novo* Ansatz oder durch Verbesserung eines anfänglichen Modells unter Berücksichtigung experimenteller Daten.

In dieser Dissertation erforsche ich die Möglichkeiten von Replica Exchange Molekulardynamik (REX MD) als ein physikbasierter Ansatz zur Erzeugung von physikalisch sinnvollen Proteinstrukturen. Dabei lege ich den Fokus darauf möglichst nativähnliche Strukturen zu erhalten und untersuche die Stärken und Schwächen der angewendeten Methode. Ich erweitere die Standardanwendung, indem ich ein kontaktbasiertes Bias-Potential integriere um die Leistung und das Endergebnis von REX zu verbessern. Die Einbeziehung nativer Kontaktpaare, die sowohl aus theoretischen als auch aus experimentellen Quellen abgeleitet werden können, treibt die Simulation in Richtung gewünschter Konformationen und reduziert dementsprechend den notwendigen Rechenaufwand.

Während meiner Arbeit führte ich mehrere Studien durch mit dem Ziel, die Anreicherung von nativ-ähnlichen Strukturen zu maximieren, wodurch der End-to-End Prozess von geleitetem REX MD optimiert wird. Jede Studie zielt darauf ab wichtige Aspekte der verwendeten Methode zu untersuchen und zu verbessern:

1) Ich studiere die Auswirkungen verschiedener Auswahlen von Bias-Kontakten, insbesondere die Reichweitenabhängigkeit und den negativen Einfluss von fehlerhaften Kontakten. Dadurch kann ich ermitteln, welche Art von Bias zu einer signifikanten Anreicherung von nativ-ähnlichen Konformationen führen im Vergleich zu regulärem REX.

2) Ich führe eine Parameteroptimierung am verwendeten Bias-Potential durch. Der Vergleich von Ergebnissen aus REX-Simulationen unter Verwendung unterschiedlicher sigmoidförmiger Potentiale weist mir sinnvolle Parameter Bereiche auf, wodurch ich ein ideales Bias-Potenzial für den allgemeinen Anwendungsfall ableiten kann.

3) Ich stelle eine *de novo* Faltungsmethode vor, die möglichst schnell viele einzigartige Startstrukturen für REX generieren kann. Dabei untersuche ich ausführlich die Leistung dieser Methode und vergleiche zwei verschiedene Ansätze zur Auswahl der Startstruktur. Das Ergebnis von REX wird stark verbessert, falls Strukturen bereits zu Beginn eine große Bandbreite des Konformationsraumes abdecken und gleichzeitig eine geringe Distanz zum angestrebten Zustand aufweisen.
4) Ich untersuche vier komplexe Algorithmusketten, die in der Lage sind repräsentative Strukturen aus großen biomolekularen Ensembles zu extrahieren, welche durch REX erzeugt wurden. Dabei studiere ich ihre Robustheit und Zuverlässigkeit, vergleiche sie miteinander und bewerte ihre erbrachte Leistung numerisch.
5) Basierend auf meiner Erfahrung mit geleitetem REX MD habe ich ein Python-Paket entwickelt um REX-Projekte zu automatisieren und zu vereinfachen. Es ermöglicht einem Benutzer das Entwerfen, Ausführen, Analysieren und Visualisieren eines REX-Projektes in einer interaktiven und benutzerfreundlichen Umgebung.

Abstract

Proteins are complex macromolecules which fulfill a wide range of critical tasks in all kinds of living organisms. For example, proteins can regulate genes, provide stability, perform cell signaling, carry substances, and perform many other tasks. Comprehensive knowledge of a protein's structure and dynamics is typically required to fully understand the physiological function or interaction mechanisms. Gained insights are essential for life sciences and can be applied to many fields, such as advanced drug design or disease treatment. Despite the incredible progress of experimental techniques, protein structure determination remains a very challenging task. Besides, experiments can only unveil partial information and measured data can be ambiguous and hard to interpret. For this reason, computer simulations are often performed to provide additional insight and complement experimental results. Nowadays, many *in-silico* methods are capable to obtain accurate models of a protein's structure, be it either with a *de novo* approach or via refinement of an initial model under consideration of experimental restraints.

In this thesis, I explore the capabilities of replica exchange molecular dynamics (REX MD) as a physics-based approach to generate physical meaningful protein structures. More specifically, I focus on obtaining native-like structures and investigate the method's strengths and weaknesses. I extend its base application by integrating a contact-based bias potential in order to improve the performance and outcome of REX. The inclusion of native contact pairs, which can be derived from both theoretical or experimental sources, drives the simulation towards desired conformations and accordingly reduces necessary computational costs.

During my work, I performed multiple studies with the goal to maximize the enrichment of native-like structures thus optimizing the end-to-end process of contact-guided REX MD. Each study aims to investigate and improve critical aspects of the applied method:
1) I study the effects of different selections of bias contacts, in particular range dependency and the negative influence of erroneous contacts. Thus I can identify what kind of bias leads to a significant enrichment of native-like conformations when compared to regular REX.
2) I perform a parameter optimization on the applied bias potential. By comparing the outcome of REX simulations using different sigmoid-shaped potentials, I can identify good parameter ranges and infer an ideal bias potential for the general use-case.
3) I introduce a *de novo* folding method to quickly generate many unique starting structures for REX. I extensively study the performance of this method and compare two different approaches of starting structure selection. The outcome of REX is greatly enhanced when initial structures exhibit already a large variety but minimal distance to the desired native state.
4) I investigate four complex algorithm chains that are capable to extract representative structures from large biomolecular ensembles generated by REX. I study their robustness and reliability, compare them with each other, and numerically rate their performance.
5) Based on my experience with contact-guided REX MD, I developed a Python package to automate and facilitate REX projects. It allows a user to design, execute, analyze, and visualize any REX project in an interactive and user-friendly environment.

Acknowledgments

Above all, I want to express my greatest gratitude to Alexander Schug, who served as my doctoral supervisor and mentor. He granted me the most freedom and flexibility during my work, and let me realize my own ideas and projects. Nonetheless, he always took his time to discuss the latest results and provide guidance when needed. I also want to thank Wolfang Wenzel for being my other supervisor, but especially for his direct and uncomplicated interactions.

Furthermore, I want to thank Marie Weiel for being one of my most precious friends since I moved to Karlsruhe in order to study physics. I really enjoyed our daily conversations and shared passion for food, the open attitude but also the occasional ranting.

Shout-out to the entire KIT-MBS research group for the relaxed atmosphere and many great discussions throughout the years (in alphabetical order): Momin Ahmad, Julian Herold, Fathia Idiris, Ines Reinartz, and Oskar Taubert.

I also want to thank all proofreaders for being such awesome human spellcheckers and finding way too many embarrassing mistakes. In particular, I am grateful to Daniela Schnedl, Katharina Eisenach, and Victor Schleeweiß.

Lastly, special thanks to my entire family for their never lasting support during my doctoral studies.

List of Publications and Compute Time Grants

Peer-reviewed articles about the work during my doctoral studies, included in this thesis:

[1] **A. Voronin**, M. Weiel, and A. Schug, 2020. *Including residual contact information into replica-exchange MD simulations significantly enriches native-like conformations*. PLOS ONE, doi: 10.1371/journal.pone.0242072.

[2] **A. Voronin**, and A. Schug, 2021. *pyrexMD: Workflow-Orientated Python Package for Replica Exchange Molecular Dynamics*. Journal of Open Source Software, doi: 10.21105/joss.03325.

[3] **A. Voronin**, and A. Schug, 2022. *Selection of representative structures from large biomolecular ensembles*. Journal of Chemical Physics, doi: 10.1063/5.0082444.

Other contributions, not included in this thesis:

[4] A. Hautke, **A. Voronin**, F. Idiris, A. Riel, F. Lindner, A. Lelièvre, B. Appel, S. Müller, A. Schug, and S. Ebbinghaus, 2022. *Conformation, condensation and mobility of CAG triplet repeat RNA in cells*. Submitted.

Overview of all compute time grants during my doctoral studies:

HPC	Year	Project	Requested (M core-h)	Granted (M core-h)
ForHLR 2	2017/2018	DCA_REX	20	15
JUWELS	2018/2019	CHKA22	15	7.5
JUWELS	2019/2020	CHKA22	0	1
JUWELS	2020/2021	CHKA22	7	7
HoreKa	2021/2022	RNArex	16.5	15

Contents

Abbreviations

2D Two-Dimensional.

3D Three-Dimensional.

CASP Critical Assessment of Techniques for Protein Structure Prediction.

Cryo-EM Cryogenic Electron Microscopy.

DCA Direct Coupling Analysis.

DBSCAN Density-Based Spatial Clustering of Applications with Noise.

GDT Global Distance Test.

GDT HA Global Distance Test High Accuracy.

GDT TS Global Distance Test Total Score.

HPC High Performance Computer.

IDP Intrinsically Disordered Protein.

LA Local Accuracy.

MC Monte-Carlo.

MD Molecular Dynamics.

MDS Multidimensional Scaling.

ML Machine Learning.

NMR Nuclear Magnetic Resonance.

NTL9 N-Terminal of L9 Protein.

SNE Stochastic Neighbor Embedding.

TPR True-Positive Rate.

TSNE T-Distributed Stochastic Neighbor Embedding.

PDB Protein Data Bank.

REX Replica Exchange.

REX MD Replica Exchange Molecular Dynamics.

RMSD Root-Mean-Square Deviation.

VHP Villin Headpiece.

1

Introduction

This chapter provides important background information. The first part puts the thesis into a broader context and introduces key literature. The second part outlines the purpose and significance of my conducted work. The third part gives an overview of the thesis structure and outlines the content of each chapter. This chapter is based on the introduction sections of my articles[1–3].

Context

Proteins are nanosized macromolecules that fulfill a wide range of critical tasks in living organisms. They are involved in the regulation of genes, conformational transitions, energy regulation of cells, signaling, enzymatic function, structural stability, protein synthesis, etc. Detailed knowledge of a protein's structure and dynamic is essential for understanding its physiological function and associated biological processes. Structural knowledge is also pivotal in related fields such as pharmacology to understand pathogenesis on a molecular level as an essential prerequisite to effective drug design. Typically, both protein structure and function are intrinsically related and defined by the corresponding amino acid sequence[5–7]. The majority of proteins thus have a classical structure-function relation, where one native fold is representative for its biological function. One interesting exception are so-called intrinsically disordered proteins (IDPs). Such proteins are more flexible in nature and have a set of different structure ensembles, separated by low-energy barriers, instead of one stable and characteristic native fold. This heterogeneity as well as fast transitions between structure ensembles during interactions make studies of IDPs and their functional interpretation much more difficult[8,9].

Over the past years, experimental sequencing techniques have become exceptionally efficient and lead to fast growing sequence databases, such as Pfam[10,11], Uniprot[12], etc. For example, GenBank[13] has currently more than 230 million sequence records and grows by approximately 10 million entries per year[14]. In contrast, experimental structure determination methods cannot keep up as they are much more time consuming and expensive. Techniques, such as x-ray diffraction and nuclear magnetic resonance (NMR) spectroscopy, have been used for high-resolution structure determination but recently in particular cryogenic electron microscopy (Cryo-EM) has achieved spectacular successes[15,16]. Other experimental methods, such as Förster resonance energy transfer (FRET) or small-angle x-ray scattering (SAXS), do not directly provide high-resolution structures. Besides, molecular structures of proteins are relatively small and have typical diameters of a few nm[17,18]. Because measured data has to be interpreted carefully experiments are often complemented by computer simulations[15,19,20] to provide additional insight and aid inferring from experimental data.

Due to the fast-paced evolution of computer technology (cf. *Moore's law*[21,22]), a broad variety of computational methods emerged over the past years which can generate accurate protein models. Typical applications combine Monte-Carlo (MC) or molecular dynamics (MD) simulations with theoretically- or experimentally-derived information to refine structural models. In particular, MD relies on time-integration of a physics-based force field, thus offering time-resolved insight into biomolecular dynamics akin to a virtual microscope with atomic resolution. Some advanced methods are even capable to predict the three-dimensional protein structure *de novo*, by starting from the amino acid sequence and utilizing various algorithms to infer a physical meaningful structure. Homology modeling[23,24] allows the transfer of structural details from known proteins to new targets with similar sequence information. Additionally, many web servers can predict critical structure information, e.g. native contact pairs or bond angles, based on sequence data. Such information can be integrated into simulations to further improve the model quality. These web servers are specialized in certain aspects and some are even providing entirely automated workflows for structure prediction. For example, the Robetta server[25] mainly utilizes homology modeling and *ab initio* fragment assembly in `Rosetta`. Meanwhile, RaptorX[26,27] focuses on machine learning and is capable of predicting secondary and tertiary structures as well as contact or distance maps, among many other things.

Every two years a new round of the Critical Assessment of Structure Prediction (CASP)[28,29] is held, which compares and evaluates the currently available state-of-the-art protein structure prediction and refinement methods. With more and more data available, purely data-driven approaches relying on machine learning and the training of neural networks can be realized with great success. Such novel approaches are able to achieve similar or even better results than "traditional" approaches. Recent improvements have shown that high-quality protein structures can be reliably obtained[30,31]. The drawback of such methods is that they are less transparent, i.e. more difficult to interpret and sometimes alike a *black box* solution with millions or even billions of trained model parameters. Additionally, they typically lack insight into physical processes driving structure adoption and cannot be easily complemented by experimental information. Depending on the applied method, local structural motifs are often less resolved and can benefit from additional refinement[30].

Physics-driven approaches are particularly suitable for this and based on semi-empirical energy functions called force fields. MD has provided valuable insight into biomolecular folding and function by itself[32]. Studies demonstrated that current force fields are sufficiently accurate to reversibly fold proteins starting from unfolded conformations[33,34]. However, the computational costs of such *de novo* folding simulations still remain extremely high which makes them only reasonable in rare occasions. The computational demands of millisecond simulations are even so high that they can only be performed on specialized supercomputers, such as Anton[35,36].

An alternative approach is given by biased simulations, where conformational transitions can be guided towards the target structures or ensembles by including an energetic bias [37–40]. Such bias information can be derived from various sources using a theoretical, experimental or even a completely data-driven approach. It is possible to integrate, e.g., sparse NMR data [41] or contact maps from co-evolutionary analysis methods. Evolutionary pressure favors fast folding times and naturally selects amino acid sequences with minimal frustration and ideally one distinct global minimum [42,43]. Despite all that, obtained energy landscapes representing the protein conformations still indicate many competing minima separated by energetic barriers and can be frustrated or "glass-like". This is also the reason why plain MD simulations often experience conformation trapping on low temperatures. The integration of a bias can smoothen the initially rugged energy landscape, simultaneously reduce the overall sampling space and thus lower the computational costs of the applied simulation method. Using the example of Ubiquitin, it was examined to what extent the application of residual contacts can speed up protein structure determination using all-atom MD simulations when starting from extended folds [44]. Integration of a flat-bottom harmonic potential, for different numbers of randomly chosen native contacts, showed a significantly accelerated convergence to near-native structures even for a rather small number of restraints as compared to unbiased simulations. In light of these results, the question arises whether one can further decrease computational demands by enhanced sampling techniques [45–47].

Contact information of adjacent amino acids can be obtained from different sources. By themselves they provide insufficient information for structure generation. However, when combined with MC or MD simulations, such information can get leveraged and drastically improve the outcome. For instance, integration of NMR-derived distance restraints into ensemble MD simulations showed that the native state of the IDP α-Synuclein, which plays a key role in the pathogenesis of Parkinson's, is composed of a more compact conformation ensemble than would be expected for a random coil [48]. Co-evolution analysis methods, such as direct coupling analysis (DCA) [49,50], can infer contact information from large multiple sequence alignments. DCA identifies co-evolving residue pairs, which can be interpreted as spatially adjacent. This information was successfully used for structure prediction [51] even in large-scale studies of proteins [52] or for RNA [53]. DCA-derived contacts have already been combined with structure-based models to uncover conformational diversity for medium to large proteins, including hidden functional configurations and intermediate states [54]. However, it often is uncertain how error-prone available contact information actually is. NMR assignments can be wrong or DCA can contain false-positive contacts.

My contribution

For this purpose, I performed an extensive study to investigate the influence of native ("correct") and non-native ("wrong") contact information with regard to structure determination. To overcome kinetic entrapment due to the multiple-minima problem during a simulation, I apply replica exchange molecular dynamics (REX MD) as an enhanced sampling technique [55–59]. In short, REX simulates multiple copies (*replicas*) of a target system at different temperatures in parallel and allows adjacent replicas to switch places. Such exchanges between temperature levels lead to trajectory jumps which disrupts the correct system dynamics at a fixed temperature but still maintains a thermodynamically correct description of the system. Additionally, I integrate a contact-based bias which effectively guides the search towards the target structure by narrowing the conformational sampling space. By combining both contact information and REX MD, I can demonstrate a significant enrichment of native and native-like conformations in the simulated ensemble of a single run.

To systematically study and test the method's performance, I conduct many simulations by starting from an unfolded state. I investigate different scenarios by varying the bias quality, i.e. the true-positive rate, and the total number of randomly selected contact pairs. Furthermore I analyze the importance of short- and long-ranged contacts and infer the required bias-quality threshold to significantly improve the results as compared to regular REX.

Based on the promising results of this *prototype study*, I decided to analyze and optimize critical aspects of contact-guided REX MD with a focus on generating native-like protein structures. One such aspect is the optimization of the applied bias potential, which has in my case the shape of a sigmoid. The potential is designed in such a way, that it guides the simulation towards native-like conformations by applying a weak attraction on predicted native contacts. The interaction strength is the order of a weak hydrogen bond and has a natural range limitation resulting from the sigmoid shape. Hence the influence of false-positive bias contacts is effectively reduced and erroneous long-range interactions are completely suppressed. I performed many simulations using different sets of potential-defining parameters to identify good parameter values. By performing these tests on one purely α-helical and one purely β-sheet structure, I am able to define an optimized potential for a general use-case.

Another aspect was the optimization of the REX starting conditions, which can drastically reduce the required simulation time before observing native-like conformations. I present a method to quickly generate many unique REX starting structures to populate each individual replica. Furthermore, I investigate the obtained structure quality and compare two different methods of structure selection. I aim to maximize structural variety while minimizing the difference towards the native fold. This provides additional folding paths when used to initialize REX simulations.

In another study I investigate structure selection methods again. This time however, I search for a robust and reliable solution to select native-like conformations from REX-generated ensembles, representing the final task of my applied method. I introduce four variations of a complex algorithm chain and analyze their selection performance in great detail. I compare their pros and cons, investigate each algorithm's robustness and rate their reliability in selecting the wanted target structures.

Besides my conducted research utilizing contact-guided REX MD, I developed a software solution to automate and facilitate REX projects based on my acquired knowledge. However, my software also acts as an *all-purpose* tool kit and combines the most critical aspects of each biomolecular study. It allows a user to design, execute, analyze, and visualize the entire project in an interactive and user-friendly environment. This is useful for many reasons. First, it lowers the technical boundaries for inexperienced users who want to apply REX. Second, it enables a fast development of new workflows by utilizing streamlined functions. But most importantly, everything can be achieved in the same environment and does not require a mixing of different specialized tools, as it is typical for biomolecular studies.

Overall, the work presented in this thesis covers different stages of contact-guided REX MD as a physical approach to generate native-like protein conformations. I discuss the general application, investigate many critical aspects, and perform extensive studies with the goal to optimize the end-to-end process. I show that contact-guided REX MD is capable to generate high-quality protein structures and that my proposed selection algorithms can reliably select representative structures from the large pool of REX-generated structures.

Thesis outline

PART I: BACKGROUND AND FUNDAMENTALS

This part introduces required biophysical and computational topics to understand the entire thesis.

- **Chapter** 2 gives a short introduction to important biophysical basics. It covers the topics of proteins, protein structure, protein folding, and contact derivation.
- **Chapter** 3 gives a broad overview of all necessary computational methods. It covers molecular dynamics (MD) in great detail, replica exchange (REX) as an enhanced-sampling technique, and the application of a sigmoid-shaped bias potential to guide by simulations. I also introduce two methods of structure comparison and the most commonly used distances metrics. Lastly, I present my preferred algorithms to perform dimension reduction or clustering and highlight their differences.

PART II: METHOD DEVELOPMENT

This part covers all of my performed studies and contributions to apply and optimize the REX MD method when used on protein targets.

- **Chapter** 4 covers the general aspects of contact-guided REX MD as a method to generate large amounts of physical meaningful structures. I highlight how the integration of a sigmoid bias potential can significantly increase native-like conformations. I also perform an extensive study covering scenarios that use restraints with either an ideal or mixed bias-quality. By doing so, I can infer the influence of native and non-native contacts for the applied method and deduce a bias-quality threshold for optimal REX performance. In another study, I investigate sigmoid potentials with different parameter values to determine the optimal bias shape for a general use-case.
- **Chapter** 5 covers the topic of starting-structure generation in order to populate each individual replica. I outline the general benefits of varying starting structures and present a MC-based *de novo* folding algorithm that is capable to generate unique starting structures in a very short time. Additionally, I investigate the quality of the obtained structures and analyze their energy levels. Lastly, I present two methods on how to select the starting structures and compare them against each other.
- **Chapter** 6 covers the selection of representative structures from a large pool of REX-generated structures. I present four complex and robust algorithm chains that can reliably select the most native-like conformations. I compare all four pipelines in great detail, discuss their pros and cons and introduce a numerical rating to reflect each method's performance.
- **Chapter** 7 introduces `pyrexMD`, which is a self-developed Python package to automate and facilitate REX projects. I give a brief overview of its module architecture, the provided functionality and demonstrate some basic applications.

PART III: Conclusions

The final part draws a conclusion on my presented work.

- **Chapter** 8 briefly summarizes each critical topic of my work and discusses both strengths and limitations of the applied method. I also mention what can be further improved and outline possible future applications and study directions.

PART I

BACKGROUND AND FUNDAMENTALS

2

Proteins

This chapter covers the biological fundamentals of proteins. Section 2.1 introduces proteins as functional biomolecules and provides additional information regarding their structural formation and properties. Section 2.2 outlines the concept of protein folding based on the "energy landscape theory" and the principle of minimal frustration. Lastly, section 2.3 covers the broad topic of contact derivation, which is required for my work due to the integration of a contact-based bias potential. This section exemplarily highlights three different techniques that are used to obtain contact information, each based on different approaches (theoretical, experimental, data-driven).

2.1 Protein Structure

Amino acids are organic compounds which consist of a carbon atom (C_α) linked to

- a carboxyl group (-COOH),
- an amine group (-NH_2),
- and a side chain R, which determines the amino acid's name and its properties.

Polypeptide chains of linearly linked amino acids are called proteins. During the linking process the carboxyl group of amino acid n reacts with the amine group of amino acid $n+1$. This releases water and the remaining atoms form a peptide bond. In the case of eucaryotes, proteins can be made up by a total of 21 different amino acids. Their structures are summarized in Fig. 2.1 and categorized by side chain properties. Amino acids can be electrically charged, hydrophobic, or polar uncharged. Additionally, four specific amino acids are categorized as *special cases*, because their properties sets them apart from the others. These amino acids are

- **Glycine**: smallest amino acids with only an H-atom as its side chain. It is hydrophilic, very flexible and due to its size important for α-helix formation. It can also introduce kinks into α-helices.
- **Proline**: has a secondary amine instead of an amine group and thus a cyclic side chain. It is unflexible, can introduce kinks into α-helices and often acts as a *helix-breaker*.
- **Cysteine**: has a thiol group (-SH) as its side chain. In an oxidizing environment two Cysteines can form a disulfur bridge thus providing additional stability to the protein structure.
- **Selenocysteine**: identical to Cysteine but with selenium (Se) instead of sulfur (S). It is the only proteinogenic amino acid of eukaryotes which is not directly encoded by the genome [60].

In general, proteins are functional biomolecules and perform virtually all critical tasks in living organisms. For example, they can regulate genes [61], provide energy [62] or structure [63], cause biochemical reactions [64], perform cell signaling [65], maintain fluid balance [66], etc.

Although proteins are very flexible by nature, the majority of them have a classical structure-function relation. Upon synthesis they undergo conformational changes and minimize their free energy until they reach one stable fold (*native conformation* [67]) which is representative for its specialized biological function. Knowledge of a protein's structure is therefore crucial for a detailed understanding of involved interactions. Intrinsically disordered proteins (IDPs) [68] pose an interesting exception to this rule, as they do not have one single stable fold. Instead, they adapt many different conformations that are separated by low-energy barriers. Frequent and fast transitions between their structure ensemble make studies of IDPs and their functional interpretation much more difficult [8,9]. Due to the increased flexibility, some IDPs can even take part in multiple interactions. For example, different signaling proteins can bind to a given receptor or a given signaling protein can bind to different receptors [69].

The structure of proteins can be categorized into different hierarchy levels. The so-called primary structure corresponds to the amino acid sequence of a protein. Recurring structure motifs, such as α-helix or β-sheet, are labeled as secondary structure. On a larger scale, the 3D structure of a single polypeptide chain is called tertiary structure and can contain multiple secondary structures. Lastly, it is also possible that multiple polypeptide chains form a complex together. Such a complex is then referred to as quaternary structure. Overall, protein structures can be small or extremely large. Their size can range from tens up to thousands of amino acids [70].

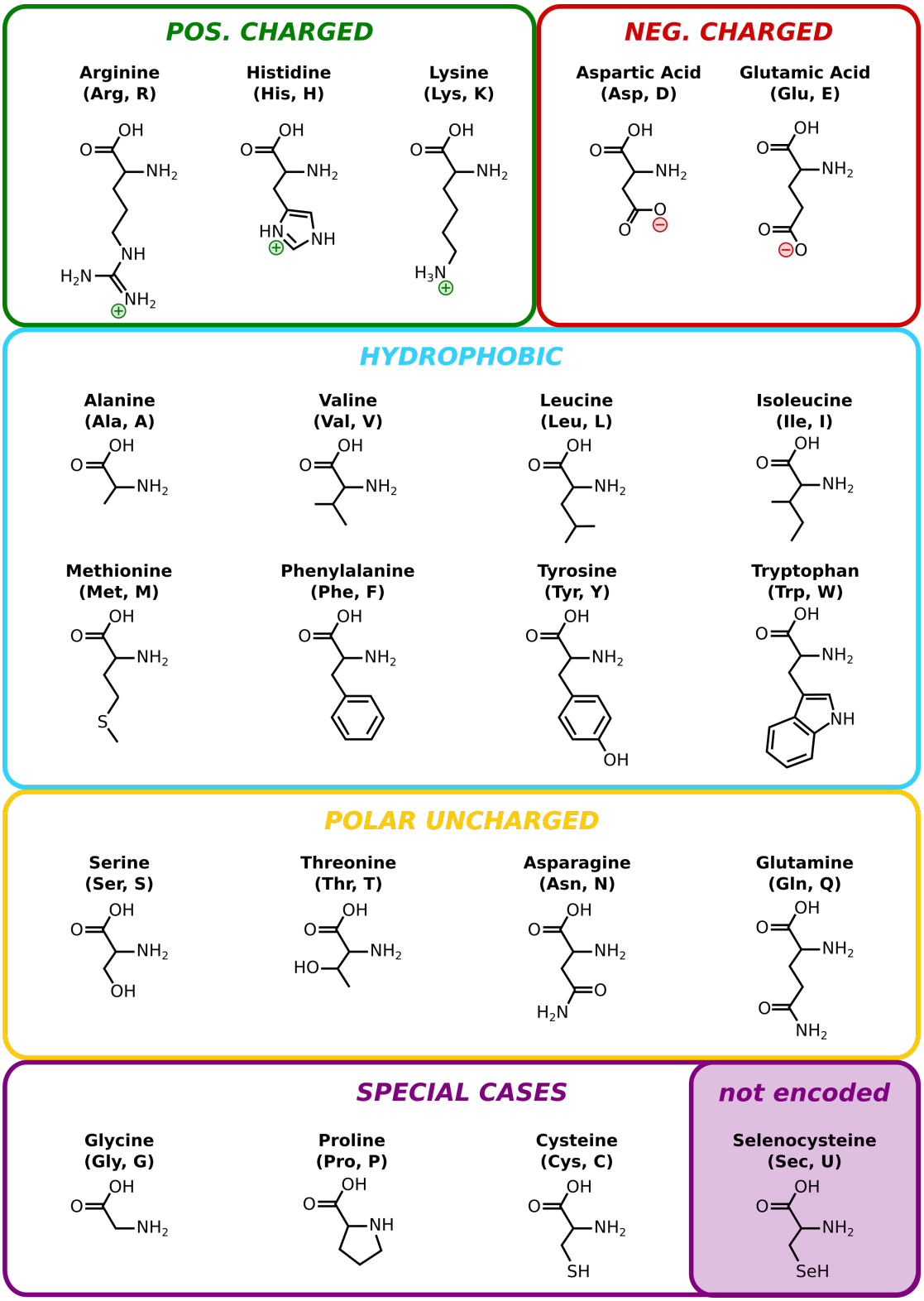

Figure 2.1. Structures of proteinogenic amino acids. Eukaryotic proteins are made up by 21 different amino acids, of which 20 are encoded in the standard genetic code (not encoded: Selenocysteine)[60]. Amino acids are sorted by side chain property. Abbreviations are denoted as 3-letter or 1-letter code. Licensed by Arthur Voronin under CC BY-SA 4.0.

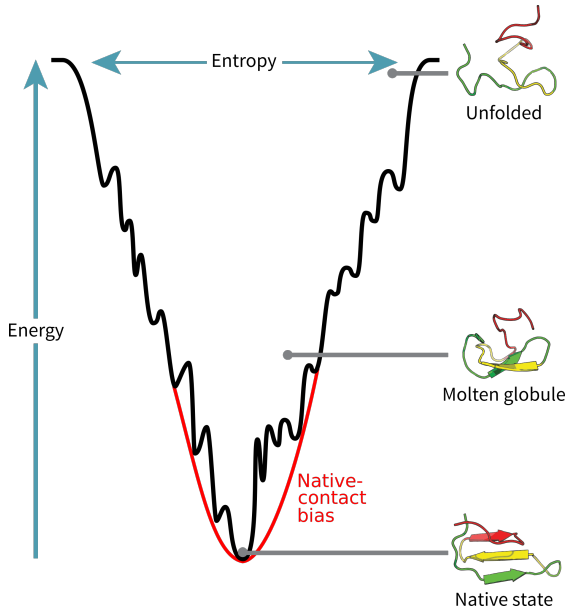

Figure 2.2. Folding funnel of a protein with/without native-contact bias. The folding pathway of a protein towards its native conformation can be illustrated by a folding funnel (black splines). Integration of an energetic bias that favors native contacts can smoothen the relatively rugged energy surface, as indicated by the red parabola. Adapted from "Folding funnel schematic" by Thomas Splettstoesser under CC BY-SA 3.0. Licensed by Arthur Voronin under CC BY-SA 4.0.

2.2 Protein Folding

Proteins are complex biomolecules with many degrees of freedom f which makes them very flexible. The total number of possible configurations can be calculated via $n = f^L$, with L being the sequence length. Levinthal's paradox[71] is a thought experiment from the year 1969 showing that proteins cannot fold randomly. In case of a protein with $L = 100$ and only $f = 3$, this would correspond to $n = 3^{100} \approx 5.2{\cdot}10^{47}$ possible conformations. Assuming that a conformational transition requires only $\tau = 10^{-15}$ s, then the required time to find the energetic minimum by sampling all conformation would be approximately $1.6 \cdot 10^{25}$ years (i.e. $1.2 \cdot 10^{15} \times$ age of universe). However, it is known that proteins fold on time scales between μs to s[72–74]. Levinthal himself mentioned that proteins undergo spontaneous folding. To achieve this, they must be "guided by the rapid formation of local interactions which then determines the further folding of the polypeptide."[71] It was shown that the introduction of a weak energetic bias can reduce Levinthal's time scale down to biologically reasonable size[75].

Among the many different models trying to explain protein folding, the energy landscape theory[76–78] is the most illustrative concept. The mapping of the Gibbs free energy G with any reaction coordinate q creates an energy surface where each point represents one structure. During folding the protein follows a path towards the global energy minimum which corresponds to the native state. In other words, this is the protein's most stable conformation where it is compact and functions properly. It was also shown that multiple folding pathways exist[79].

Note that random protein sequences imply frustration, i.e. energetic conflicts between different conformations caused by kinetic traps which restrict protein movement. This would directly interfere with the protein's folding process and in the worst case lead to malfunction. Consequentially evolution would naturally select amino acid sequences with a minimal frustration[42,43], which corresponds to relatively smooth energy landscapes with one distinct global minimum, thus favoring fast folding times.

Under perfect conditions the energy landscape of such proteins would be funnel-shaped and lead towards the native state. However, it is impossible to completely eliminate frustration of such complex molecules, which is why the energy landscape of most proteins remains partially rugged. This can be observed in MD simulations, where a proteins often get trapped in one of the many local minima. But various advanced methods are capable to overcome such entrapment. For example, it is possible to apply enhanced sampling techniques to overcome energetic barriers [46,80,81]. Alternatively it is possible to implement an energetic bias, which can smoothen the initially rugged energy landscape [82]. Fig. 2.2 illustrates the folding funnel of a protein and its smoothed variant after adding a native-contact bias.

2.3 Contact Derivation

Structural information of proteins, or more specifically contact information, can be obtained in many ways. One such example is direct coupling analysis (DCA) [49–52], which aims to predict native contacts based on the co-evolution [83,84] of spatially close protein residues. Because random amino acid mutations can destabilize a protein or even lead to its malfunction, such mutations must be naturally compensated with other mutations of spatially close residues [85]. The occurrence of such pairwise mutations can be interpreted as an evolutionary fingerprint. It can also be statistically measured by analyzing different proteins of the same family that occur in living organisms. DCA requires a multiple sequence alignment (MSA) [86–88] of a particular protein family and analyzes frequencies and correlations of occurring amino acids. It is then possible to infer spatial proximity of residues using a mathematical model, which is motivated by statistical mechanics. For instance, DCA approximates the probability of a given amino acid sequence $P(a_1, ...a_L)$ with length L using the generalized Potts model [50,52]

$$P(a_1, ...a_L) = \frac{1}{Z} \exp\left(\sum_{i<j}^{L} e_{ij}(a_i, a_j) \sum_{i}^{L} h_i(a_i)\right), \tag{2.1}$$

with the amino acids a_i, the normalization factor Z (*partition function*), and the Lagrange multipliers e_{ij} and h_i. Note that e_{ij} (*couplings*) describe the interaction strengths and thus the compatibility of the pairwise amino acids. h_i (*fields*) on the other hand correspond to the local amino-acid biases resulting from evolutionary pressure. The final goal of DCA is to generate a list of residue pairs and rank them based on the calculated couplings to predict residual contacts.

Another example for contact derivation is given by nuclear magnetic resonance (NMR) spectroscopy [89], where chemical compounds are exposed to an external magnetic field and the resonance frequencies of nuclei can be measured relative to the frequency of a standard compound. The obtained resonance spectrum can be used to infer the chemical structure of the sample, because its chemical environment affects the spin resonance and thus the measured signals. An NMR spectrum contains information based on three underlying principles [89,90].

1. **Chemical Shift:**
   The chemical shift δ (resonance signal relative to the standard signal) contains information about the chemical environment of a nucleus. Being close to a strong electronegative environment has a deshielding effect and withdraws the electrons around the nucleus. This in turn affects its resonance, such that it appears *downfield* (left side of the spectrum with higher ppm). In the opposite scenario the nucleus is further away from electronegative elements, is more shielded, and the resonance appears *upfield* (right side of the spectrum with lower ppm). The chemical shift δ is given by [91]

$$\delta = \frac{\nu_{\text{sample}} - \nu_{\text{ref}}}{\nu_{\text{ref}}}, \tag{2.2}$$

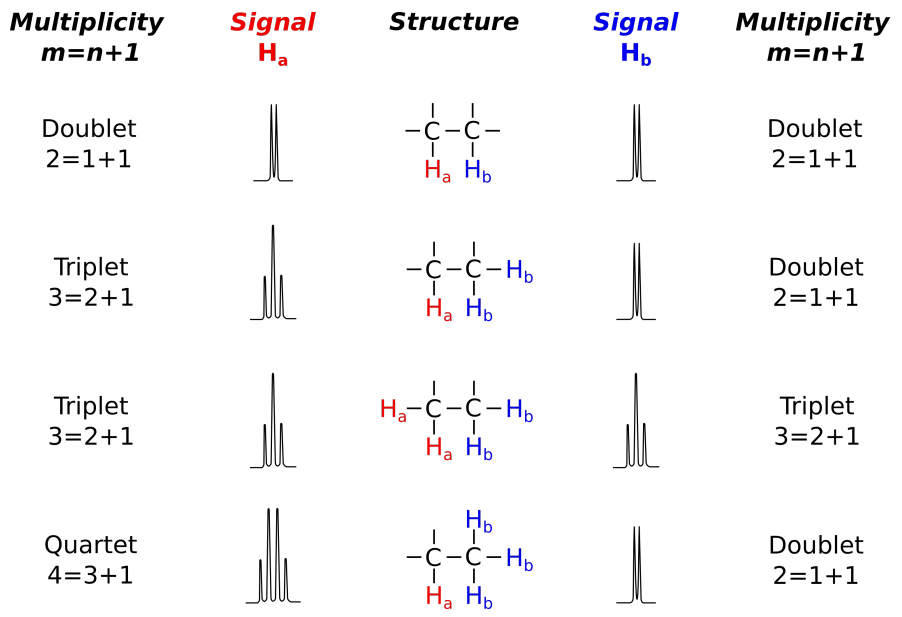

Figure 2.3. Splitting patterns for ^{1}H-NRM spectroscopy. Hydrogen has a spin of 1/2, which simplifies the splitting rule to $m = n + 1$. This figure exemplarily shows the expected signal patterns for different combinations of neighboring hydrogen atoms. Adapted from "Summary-of-signal-splitting-patterns-NMR-spectroscopy.png" with permission of Gevorg Sargsyan. Licensed by Arthur Voronin under CC BY-SA 4.0.

with the resonance frequencies ν_i of the sample or the reference compound, and δ being normally expressed in ppm (parts per million)[91]. Typical reference compounds[92] are, e.g., Tetramethylsilane (TMS)[93] and Trimethylsilylpropanoic acid (TMSP or TSP)[94].

2. **Integration:**

Chemical-equivalent nuclei (same environment) resonate at the same frequency, which enhances the signal. Its intensity can be used to directly infer the number of resonating nuclei[95].

3. **Splitting:**

Resonance signals of chemical-equivalent nuclei can have one peak or be split into multiple peaks. This splitting pattern is based on the underlying *multiplicity* of the *spin coupling*. It is an important feature for NRM structure determination, as it infers the number of neighboring nuclei according to the splitting rule $m = 2nI + 1$[89,95]. Here the multiplicity is denoted as m, the magnetic spin number as I, and the number of neighboring nuclei as n. Fig. 2.3 exemplarily shows such a splitting pattern as seen in ^{1}H-NMR spectroscopy.

As a final example, it is nowadays possible to obtain both structure or contact information using an entirely data-driven approach. Large databases, such as PDB[96], Uniprot[12], or GenBank[13], contain thousands of experimentally determined protein and RNA structures. Having access to large amounts of data allows the development of AI-based solutions. For instance, the residual deep neural network ResTriplet[97,98] is designed to predict contact maps for submitted protein sequences. On the other hand, AlphaFold[99,100] uses AI to predict distances, angles and then even folds the entire protein with a high accuracy[101]. Besides, having more and more data available improves the overall performance and prediction of AI-based solutions incrementally over time. That is because successful models can be improved with relatively low effort, i.e. simply by using a newer and larger data set to retrain and update model parameters.

3

Computational Methods

This chapter covers computational methods and algorithms that are required for my work. Section 3.1 introduces molecular dynamics (MD) as a simulation method to provide an atomic view of molecular motions and interactions. This in-silico approach aids the understanding of protein folding and functioning but can also be used to gain additional insight into complex biomolecular interactions. Section 3.2 introduces Replica Exchange (REX) as my preferred simulation method to generate large amounts of protein models. I extend this method by integrating a sigmoid bias potential to support the generation of native-like conformations, as explained in section 3.3. Sections 3.4 and 3.5 cover different structure comparison methods and applied distance metrics, respectively. Next in section 3.6 I explain the general benefits of dimension-reduction techniques and cover two specific methods which are applied during my work, i.e t-distributed stochastic neighbor embedding (TSNE) and multidimensional scaling (MDS). Similarly to this, section 3.7 introduces the general topic of data clustering and highlights two algorithms that are important for my work as well, i.e. KMEANS and DBSCAN (density-based spatial clustering of applications with noise).

3.1 Molecular Dynamics

Molecular Dynamics (MD)[102] is a computer simulation method for studying physical movements and interactions of atoms and molecules. The simulations apply physics-based models to predict the dynamic behaviour of a biomolecular system on timescales up to milliseconds. Obtained trajectories can be viewed similar to a movie and analyzed in great detail, unveiling key mechanisms on both temporal and spatial scale. This *in silico* method can therefore be viewed as a computational microscope allowing us to observe complex interactions or it can be used to complement real experiments. Depending on the simulation target, it is possible to get additional insights into, e.g., protein folding or ligand bonding.

MD simulations numerically solve Newton's equations of motions for the simulated system particles. Atomic interactions are based on empirical force fields, which were derived from conducted experiments and can replicate the results with high accuracy. Given a system of N atoms, the forces acting on the atoms i can be described via the physical potential $V(\boldsymbol{r}_1, ..., \boldsymbol{r}_N)$. Its relation is given by

$$\boldsymbol{F}_i = \frac{\partial V}{\partial \boldsymbol{r}_i} \qquad i = 1, ..., N. \tag{3.1}$$

Due to the large system sizes MD typically only applies classical/molecular-mechanical (MM) force fields, since quantum-mechanical (QM) simulations are much more demanding and thus computationally costly. However, computer technology evolved at an incredible pace and newer high performance computers are capable to perform hybrid QM/MM simulations[103,104]. There are many different force fields, such as AMBER[105] or CHARMM[106], that can be used for MD simulations. Each describes the atomic interactions using their respective model, which is based on different energy terms, correction terms, atom-dependant coupling parameters, etc. Once the forces are calculated, the simulation software computes the next time step of the molecular trajectory by solving the equations of motions, i.e.

$$m_i \frac{\partial^2 \boldsymbol{r}_i}{\partial t^2} = \boldsymbol{F}_i \qquad i = 1, ..., N. \tag{3.2}$$

The time step is of great importance as it directly influences the accuracy of the simulation. Using very small time steps makes the simulation computational costly. Using too large time steps, however, can lead to catastrophic errors, such as exploding systems due to the sudden occurrence of extremely high repulsive forces. Hence time steps are typically in the order of 1-2 femtoseconds generating trajectories with smooth molecule movements, which can be later visualized or analyzed. During my studies I explicitly used `GROMACS`[107–109] to run my simulations, which is one of the most popular MD solutions that are available.

AMBER Force Field

Force fields are computational models that describe the interacting forces between the simulated system particles. The choice of the force field is very important because it strongly affects the prediction accuracy of the MD simulation. Force field parameters are tuned in such a way that MD simulations are capable to reproduce experimental results. In my work I primarily apply the `AMBER99SB-ILDN`[110] force field during the simulations. Its potential is given by

$$V = \underbrace{V_{\text{bond}} + V_{\text{angle}} + V_{\text{dihedral}}}_{\text{bonded interactions}} + \underbrace{V_{\text{electrostatic}} + V_{\text{Van-der-Waals}}}_{\text{non-bonded interactions}}. \tag{3.3}$$

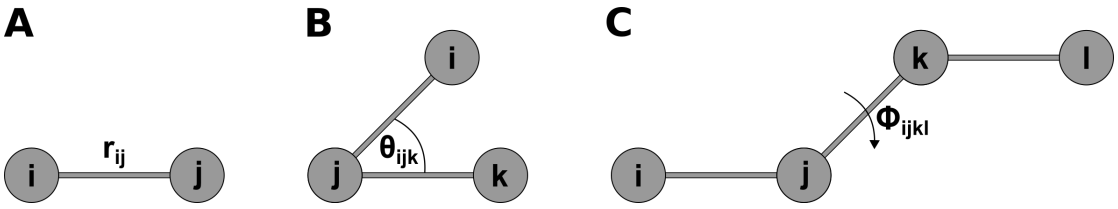

Figure 3.1. Bonded interaction types. Interactions are described by Eqs. 3.4 to 3.6 via the potentials V_{bond}, V_{angle} and V_{dihedral}. **(A)** Bond stretching (2-body). **(B)** Bond bending (3-body). **(C)** Bond rotation/torsion (4-body).

Bonded interactions describe short-ranged intramolecular interactions. As shown in Fig. 3.1, they are modelled by 2-body bond, 3-body angle or 4-body dihedral-angle potentials. Both bond and angle potentials are described by harmonic oscillators according to Hooke's law, i.e. [109]

$$V_{\mathrm{bond}}(r_{ij}) = \frac{1}{2} k^b_{ij} (r_{ij} - r^0_{ij})^2, \tag{3.4}$$

$$V_{\mathrm{angle}}(\theta_{ijk}) = \tfrac{1}{2} k^a_{ijk} (\theta_{ijk} - \theta^0_{ijk})^2. \tag{3.5}$$

In these equations the spring constants are labelled as k, equilibrium bond distances as r^0 and equilibrium angles as θ^0. On the other hand, 1-4 dihedral-angle interactions are designed as periodic potentials by [109]

$$V_{\mathrm{dihedral}}(\phi_{ijkl}) = \sum_n k^d_{ijkl} (1 + \cos(n(\phi_{ijkl} - \phi^0_{ijkl}))), \tag{3.6}$$

with the spring constants k, the equilibrium dihedral angles ϕ^0 and the multiplicity n. The last two terms of the AMBER potential are describing the non-bonded interactions which occur over long distances. To minimize computational costs the MD software typically applies user-defined distance cutoffs. The electrostatics is given by the Coulomb potential

$$V_{\mathrm{electrostatic}}(r_{ij}) = \frac{1}{4\pi\varepsilon_0\varepsilon_{\mathrm{r}}} \cdot \frac{q_i q_j}{r_{ij}}, \tag{3.7}$$

with the vacuum permittivity ε_0, the relative permittivity ε_{r}, the atom charges q_i or q_j, and the atom distances r_{ij}. Lastly, the Van-der-Waals interactions are provided by the Lennard-Jones potential. It describes the interactions of non-charged, chemical non-bonded atoms via

$$V_{\text{Van-der-Waals}}(r_{ij}) = k^{\mathrm{LJ}} \left[\left(\frac{\sigma^0_{ij}}{r_{ij}} \right)^{12} - 2 \cdot \left(\frac{\sigma^0_{ij}}{r_{ij}} \right)^{6} \right]. \tag{3.8}$$

Here, the constant k^{LJ} corresponds to the bonding energy, i.e. the potentials well depth, which is necessary to separate the molecule's atoms. The volume-exclusion radius is denoted as σ^0_{ij} and corresponds to the potential's minimum distance. The first part (exponent 12) of Eq. 3.8 reflects the Pauli principle and states the repulsion at small distances r_{ij}, whereas the second part (exponent 6) is derived from QM and describes attraction. Note that the Lennard-Jones potential can also be written differently. Using the substitutions $A = k\sigma^{12}$ and $B = 2k\sigma^6$ it can be expressed in its original form [111], which reads

$$V = \frac{A}{r^{12}} - \frac{B}{r^6}. \tag{3.9}$$

MD integrator

The numerical solving of differential equations can be achieved with algorithms called *integrators*. The default MD integrator of `GROMACS` uses the leap-frog algorithm [109]. It alternates the calculation of the atom positions $\boldsymbol{r}_i$ and their velocities $\boldsymbol{v}_i$, hence the naming *leap-frog*. Information taken from past time steps is used to calculate future time steps according to

$$\boldsymbol{r}_i(t + \Delta t) = \boldsymbol{r}_i(t) + \Delta t \cdot \boldsymbol{v}_i(t + \tfrac{1}{2}\Delta t), \tag{3.10}$$

$$\boldsymbol{v}_i(t + \tfrac{1}{2}\Delta t) = \boldsymbol{v}_i(t - \tfrac{1}{2}\Delta t) + \frac{\Delta t}{m_i} \cdot \boldsymbol{F}_i(t). \tag{3.11}$$

There are two possible options for the start of a new MD simulation:

1) Use saved velocities from an older run and continue the simulation.
2) Generate random initial velocities and start a new simulation.

The random velocities can be generated with the Maxwell-Boltzmann distribution. The momenta $p(v_i)$ are given by

$$p(v_i) = \sqrt{\frac{m_i}{2\pi k_\mathrm{B} T}} \exp\left(-\frac{m_i v_i^2}{2 k_\mathrm{B} T}\right) \qquad i = 1, ..., 3N, \tag{3.12}$$

with the atomic masses m_i, the Boltzmann constant k_B and the system temperature T. After the simulation is over, the MD trajectory then spans a time interval of

$$t_\mathrm{MD} = n_\mathrm{MD} \cdot \Delta t, \tag{3.13}$$

with the number of performed MD steps n_MD and the time step interval Δt.

Temperature Coupling

The kinetic energy E_kin of a N-particle system is given by the equations

$$E_\mathrm{kin} = \frac{1}{2} \sum_{i=1}^{N} m_i v_i^2, \tag{3.14}$$

$$E_\mathrm{kin} = \frac{f}{2} k_\mathrm{B} T, \tag{3.15}$$

with the masses m_i, velocities v_i, degrees of freedom f, Boltzmann constant k_B and the system temperature T. The degrees of freedom can be calculated via [109]

$$f = 3N - N_c - 3, \tag{3.16}$$

where N is the number of particles and N_C is the number of constraints that are imposed on the system. Because the three center of mass velocities are constants of the motion, they can be set to zero which is reflected by the -3 term.

Uncoupled MD simulations represent the microcanonical ensemble (NVE) where the number of particles N, the system volume V and the system energy E are conserved. However, experiments are typically performed by keeping either the temperature T or the pressure P constant. This corresponds to a canonical ensemble (NVT) or an isothermal-isobaric ensemble (NPT). Such ensembles can be achieved in MD by coupling the system to an external heat bath or pressure reservoir.

A temperature coupling can be achieved with, e.g., the Berendsen thermostat. It applies a weak coupling to an external heat bath at temperature T_0, such that the system temperature T follows a first-order kinetics with the coupling time constant τ. It is described by

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau}. \tag{3.17}$$

The coupling strength can be varied based on the choice of τ. It is sufficient to use $\tau = 0.01\,\mathrm{ps}$ for a quick equilibration but more reliable equilibrations should use much larger coupling times, for example $\tau = 0.5\,\mathrm{ps}$[109]. One major issue of the Berendsen algorithm is that it suppresses the fluctuations of kinetic energies and thus cannot generate a proper canonical ensemble. This can be fixed with a modification. The improved variant is called velocity-rescaling thermostat, or just v-rescale in short. It is a Berendsen algorithm that applies an additional stochastic term to correct the kinetic energy distribution. The auxiliary dynamics is described by the equation[112]

$$dK = (K_0 - K) \cdot \frac{dt}{\tau_0} + 2\sqrt{\frac{K K_0}{f}}\frac{dW}{\sqrt{\tau_0}}, \tag{3.18}$$

with the kinetic energy K, the arbitrary time parameter τ_0, the degrees of freedom f, and the Wiener process dW. The parameters τ_0 and τ typically have very close but unequal values and satisfy

$$\frac{\tau}{\tau_0} = \frac{2C_\mathrm{V}}{f k_\mathrm{B}}. \tag{3.19}$$

Denoted parameters stand for the coupling time constant τ, arbitrary time parameter τ_0, heat capacity C_V at constant volume V, degrees of freedom f, and the Boltzmann constant k_B. New velocities are rescaled to λv by this algorithm with the factor[113]

$$\lambda = \left[1 + \frac{n_\mathrm{TC}\Delta t}{\tau_0}\left(\frac{T_0}{T(t - \frac{1}{2}\Delta t)}\right)\right]^{1/2}. \tag{3.20}$$

In this case, n_TC represents the number of temperature coupling steps, Δt the integration time step, while T and T_0 denote the heat bath's and system's temperatures, respectively. The rescaling also affects the kinetic energies according to

$$\Delta E_\mathrm{kin} = (\lambda - 1)^2 E_\mathrm{kin}. \tag{3.21}$$

In order to conserve the system's energy it is necessary to subtract the sum of these changes from the total energy[109].

Pressure Coupling

Analogously, the Berendsen barostat can achieve the desired pressure coupling by interacting with an external pressure reservoir at constant pressure $\boldsymbol{P}_0$. The first-order kinetics for a system with pressure $\boldsymbol{P}$ is given by the equation

$$\frac{d\boldsymbol{P}}{dt} = \frac{\boldsymbol{P}_0 - \boldsymbol{P}}{\tau_\mathrm{P}}, \tag{3.22}$$

with the pressure coupling time constant τ_P. The pressure scaling matrix μ[109,114] is described by

$$\mu_{ij} = \delta_{ij} - \frac{n_\mathrm{PC}\Delta t}{3\tau_\mathrm{P}}\beta_{ij}(P^0_{ij} - P_{ij}). \tag{3.23}$$

n_PC represents the number of pressure coupling steps, Δt the integration time step, β_{ij} the isothermal compressibility, while P^0 and P denote the reservoir's and system's pressures, respectively.

Additionally, δ_{ij} is the Kroneckar delta and stands for

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{3.24}$$

The systems volume V is rescaled according to

$$V(t + \Delta t) = \mu V(t). \tag{3.25}$$

It is also possible to perform isotropic coupling (scaling is equal in each direction) or semi-isotropic coupling (scaling in x/y is different to z direction). The latter case is for example relevant for lipid bilayers.

The drawback of the Berendsen barostat is that it suppresses pressure and volume fluctuations, which leads to an incorrectly represented ensemble. However, slightly advanced algorithms are able to simulate a proper NPT ensemble. For example, the Parrinello-Rahman barostat [115,116] satisfies the equations of motion described by

$$\frac{d\boldsymbol{b}^2}{dt^2} = V\boldsymbol{W}^{-1}\boldsymbol{b}'^{-1}(\boldsymbol{P} - \boldsymbol{P}_0). \tag{3.26}$$

This equation denotes the simulation box vectors as $\boldsymbol{b}$, the box volume as V, whereas $\boldsymbol{P}$ and $\boldsymbol{P}_0$ are the system's and reservoir's pressures, respectively. $\boldsymbol{W}^{-1}$ represents the inverse mass parameter matrix, which determines the coupling strength and is calculated according to

$$\left(\boldsymbol{W}^{-1}\right)_{ij} = \frac{4\pi\beta_{ij}}{3\tau_{\mathrm{P}}^2 L}. \tag{3.27}$$

β_{ij} denotes the isothermal compressibility, τ_{P} the pressure coupling time constant and L is the largest box matrix element. Note that Parrinello-Rahman, which is an extended ensemble coupling, typically requires a 4-5 times longer pressure coupling time constant τ_{P} as compared to the Berendsen barostat. If the simulated system is far from equilibrium this can lead to a simulation crash due to the box oscillations. In such cases, it is better to apply a Berendsen barostat until equilibration and then switch back to Parrinello-Rahman. Using this algorithm also conserves the energy with the modified Hamiltonian

$$E_{\mathrm{pot}} + E_{\mathrm{kin}} + \sum_i P_{ii}V + \sum_{i,j} \frac{1}{2} W_{ij} \left(\frac{db_{ij}}{dt}\right)^2. \tag{3.28}$$

The derived equations of motions are given by

$$\frac{d\boldsymbol{r}_i^2}{dt^2} = \frac{\boldsymbol{F}_i}{m_i} - \boldsymbol{M}\frac{d\boldsymbol{r}_i}{dt}, \tag{3.29}$$

$$\boldsymbol{M} = \boldsymbol{b}^{-1}\left[\boldsymbol{b}\frac{d\boldsymbol{b}'}{dt} + \frac{d\boldsymbol{b}}{dt}\boldsymbol{b}'\right]\boldsymbol{b}'^{-1}. \tag{3.30}$$

In this case, atom positions are denoted as $\boldsymbol{r}_i$, their masses as m_i, acting forces as $\boldsymbol{F}_i$, and simulation box vectors as $\boldsymbol{b}$. Although the additional term is expressed like a friction, it is just the effect of the Parrinello-Rahman equations of motion being defined with all particle coordinates represented relative to the box vectors [109].

Water Model

Proteins typically appear in an aqueous environment. Cytosol, i.e. the cellular solvent, consists primarily of water ($\approx 70\,\%$), proteins ($\approx 20-30\,\%$) and different types of dissolved ions[117]. This is also reflected in MD simulations with the majority of a system consisting of water atoms, for instance 2000 protein atoms vs. 50000 water atoms and a few counter ions to neutralize the system. Water molecules are relatively small with diameters of approximately 2.8 Å[118]. They are dipolar due to the structural arrangement of atoms with an angle of approximately 104.5° and interact with other substances via non-bonded interactions. For this reason, water modelling cannot be neglected in MD simulations as it also impacts the progression of the system's trajectory. There are two main categories of water models: 1) explicit water, where individual water atoms are embedded into the simulation and interact accordingly and 2) implicit water, which represents the water as a continuous medium throughout the system. The majority of MD simulations rely on explicit water models since they tend to be more accurate and can emulate real-world experiments. In certain scenarios, however, it is sufficient to apply implicit water, which is computationally less demanding. One such example is the free-energy estimation of various solute-solvent interactions.

Most MD simulations apply 3-site water models with rigid H_2O molecules, which are computationally efficient due to their simplicity. These explicit models cover non-bonded interactions via a Coulomb potential for electrostatics and a Lennard-Jones potential for Van-der-Waals interactions. It reads

$$V = \sum_{i,j} \underbrace{\frac{1}{4\pi\varepsilon_0\varepsilon_\mathrm{r}} \cdot \frac{q_i q_j}{r_{ij}}}_{\text{Coulomb}} + \underbrace{\frac{A}{r_{\mathrm{ij}}^{12}} - \frac{B}{r_{\mathrm{ij}}^{6}}}_{\text{Lennard-Jones}} , \tag{3.31}$$

with the vacuum permittivity ε_0, relative permittivity ε_r, partial charges q_i and q_j, atom distances r_{ij}, and the Lennard-Jones parameters A and B. Table 3.1 compares different 3-site water models based on their parameters. All of my MD simulations applied the explicit water model TIP3P (transferable intermolecular potential with three points)[119,120]. Note that more complex water models include partially charged dummy atoms to improve the electrostatic distribution around the water molecule. Currently available water models can reach even up to six sites[121,122]. Needless to say, the computational costs scale proportionally to the number of distance calculations that are required for the different water models. Or more precisely, for each water molecule pair the 3-site models require $3 \times 3 = 9$, 4-site models $3 \times 3 + 1 = 10$, 5-site models $4 \times 4 + 1 = 17$, and 6-site models $5 \times 5 + 1 = 27$ distance calculations.

Table 3.1. Comparison of different 3-site water model parameters. All models are explicit. i.e. individual water atoms are simulated in MD. Non-bonded interactions are reflected by the parameters listed in this table and the potential V according to Eq. 3.31. The most notable difference is that SPC models use a tetrahedral angle of approximately 109.5° instead of the experimentally measured angle of circa 104.5°. Furthermore, SPC/E is the only water model that additionally includes an average polarization correction. Due to the fixed charges this corresponds to a total energy increase of approximately 1.25 kcal/mol or 5.22 kJ/mol.

parameter	TIPS[119]	TIP3P[120]	SPC[123]	SPC/E[124]
r_HO (Å)	0.9572	0.9572	1.0	1.0
α_HOH (deg)	104.52	104.52	109.47	109.47
A (10^3 kcal Å^{12} mol^{-1})	580.0	582.0	629.4	629.4
B (kcal Å^6 mol^{-1})	525.0	595.0	625.5	625.5
q_H (e)	+0.40	+0.417	+0.41	+0.4238
q_O (e)	−0.80	−0.834	−0.82	−0.8476

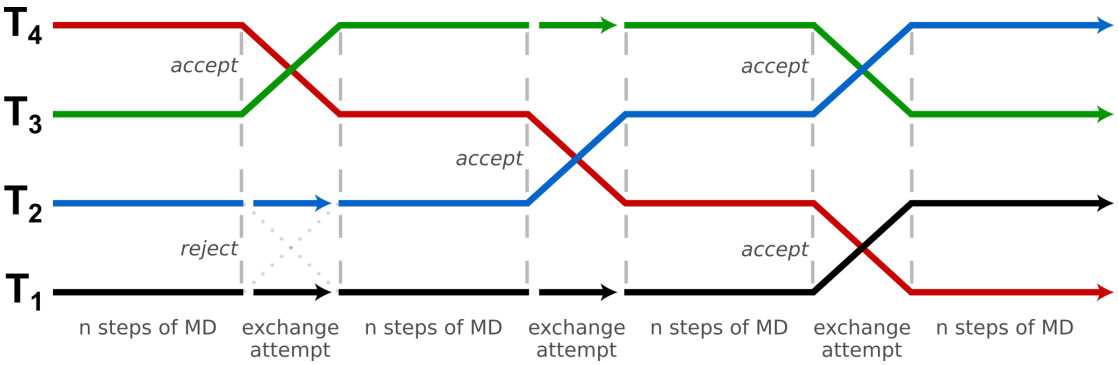

Figure 3.2. Scheme of replica exchange molecular dynamics. This figure shows the core concept of replica exchange. Four replicas (depicted in black, blue, green and red) start at different temperatures T_1 to T_4. After n steps of MD, adjacent replica pairs have a probability to get exchanged according to the Metropolis-Hastings criterion as defined by Eq. 3.32. After the next n steps of MD considered replica pairs are shifted by ± 1. This alternation allows replicas to perform a random walk in temperature space, if the temperature distribution was chosen properly. Adapted from "Schematic of a replica exchange molecular dynamics simulation" by Christopher Rowley under CC BY-SA 4.0. Licensed by Arthur Voronin under CC BY-SA 4.0.

3.2 Replica Exchange

Proteins undergo conformational transitions on timescales of the order μs to s [72–74]. Since MD simulations operate on fs time steps, observations of such large scale transitions are computationally very expensive. Furthermore, protein energy surfaces tend to be rugged and typically have multiple local minima. Hence MD simulations at low temperatures often get stuck in certain conformations as the provided thermal energies are not sufficient to overcome local energy barriers. Replica exchange (REX) [55–58], sometimes referred to as parallel tempering [59], is an enhanced sampling technique for MD that is capable to overcome this protein entrapment problem.

REX simulates N non-interacting copies (*replicas*) of a system at different temperatures T_i. As shown in Fig. 3.2, these replicas perform normal MD simulations in parallel. However, after a pre-defined amount of MD steps the energies of adjacent replicas are compared and some replicas are switched. More precisely, all atom positions and their momenta are exchanged between both systems. The acceptance probability is given by the Metropolis criterion [56,59], which is defined as

$$P(X_i \to X_j) = \min(1, e^{-\Delta}) \qquad \text{with } \Delta = (\beta_j - \beta_i)(E_i - E_j). \tag{3.32}$$

Adjacent replica states are denoted as X_i and X_j, the inverse temperature as $\beta_i = \frac{1}{k_\mathrm{B} T_i}$, the Boltzmann constant as k_B, the system's temperature as T_i, and its energy as E_i. After the accepted replicas are exchanged, the simulations continue and each replica undergoes normal MD simulations again. During the next exchange attempt considered replica pairs are shifted by ± 1, which allows individual replicas to walk over the entire temperature range. A true random walk can be achieved if exchange rates are constant for all temperatures. MD simulations typically generate ensembles where the probability distribution of each microstate is proportional to the Boltzmann distribution, i.e. $e^{-\beta E}$. Hence by choosing an exponential temperature distribution (cf. appendix A.3) it is possible to obtain constant exchange rates, as they are proportional to the overlapping area of two Boltzmann distributions. Note that exchange rates are significantly lower for large temperature differences, as shown by Δ in Eq. 3.32). For this reason, only replicas that are direct neighbors are considered during exchange attempts.

The primary goal of REX is to generate large amounts of physically meaningful structures. High temperatures provide sufficient energy to overcome local barriers, which solves the protein entrapment problem of regular MD. On the other hand, low temperatures lead to local searches of native-like conformations. Combined, this enhanced sampling technique is capable to obtain highly native-like conformations within relatively short runs. Desired exchange rates should be in the order of 20% to 30%. However, the time intervals at which exchanges are attempted are also important. The best performances can be obtained from simulations with many turnaround cycles over the entire temperature range. Some studies also indicate that REX simulations should exchange as often as possible [125,126]. Nonetheless, it is still important to stay long enough at a fixed temperature in order to properly probe the relevant conformations for such energies. REX is a physics-driven enhanced sampling technique which excels in its straight forward but diverse application. Based on the protein target and system size, it is possible to observe native-like conformations within a single REX simulation.

3.3 Sigmoid Bias Potential

All of my performed studies rely on REX simulations to generate large biomolecular ensembles. However, I additionally integrate an attractive potential which interacts only with selected bias contacts to guide my simulations towards native-like conformations. Such contact information can be obtained from various sources, e.g. theoretically-derived via DCA [49–51] or experimentally-derived from (sparse) NMR data [41,127]. It is generally unknown which residue atoms are the closest to each other. For this reason, I apply a simplistic model and let the potential interact only with selected C_α-C_α atom pairs. The selected shape of the integrated bias potential is of great importance, since it defines the strength and distance dependency of the resulting force. For instance, a regular harmonic potential becomes stronger over large pair distances. Hence it would primarily affect false-positive/non-native bias contacts which is contrary to the intended application. In my case, I require a potential that is limited and self-regulating, such that false-positive bias contacts are weakened and do not impact the simulation in a negative way. This can be achieved with a sigmoid-shaped bias potential [1], which is described by

$$V(r) = \lambda\, \sigma(r), \tag{3.33}$$

with the coupling strength λ and the sigmoid function

$$\sigma(r) = \frac{A}{1 + e^{-\alpha(r - r_0)}}. \tag{3.34}$$

A describes the upper limit of the sigmoid function and α its S shape (i.e. the transition from low to high values), while r and r_0 stand for the bias pair distance and its equilibrium distance, respectively. Note that the value of r_0 defines the inflation point which is crucial for differentiating between native and non-native contacts, as it is also the distance with the strongest attractive force. On the other hand, the choice of α defines the effective range of the force, where low α values correspond to a slow and smooth transition, whereas high α values correspond to shapes similar to a step function.

By integrating this sigmoid bias potential into REX and coupling it with selected bias contacts, the simulation becomes contact-guided and drives the structure sampling towards native-like conformations [1]. The energetic penalty also reduces the search space, meaning that the wanted target structure can be observed much sooner which in turn reduces the computational costs of REX. The sigmoid potential depicted in Fig. 3.3 corresponds to the optimized bias potential for proteins (cf. study in section 4.2). Its parameters are $A = 1$, $\alpha = 25\,\text{Å}^{-1}$ and $r_0 = 16\,\text{Å}$. $A = 1$ was chosen for simplicity because the attractive force is proportional to λA and is now solely defined by the coupling strength λ.

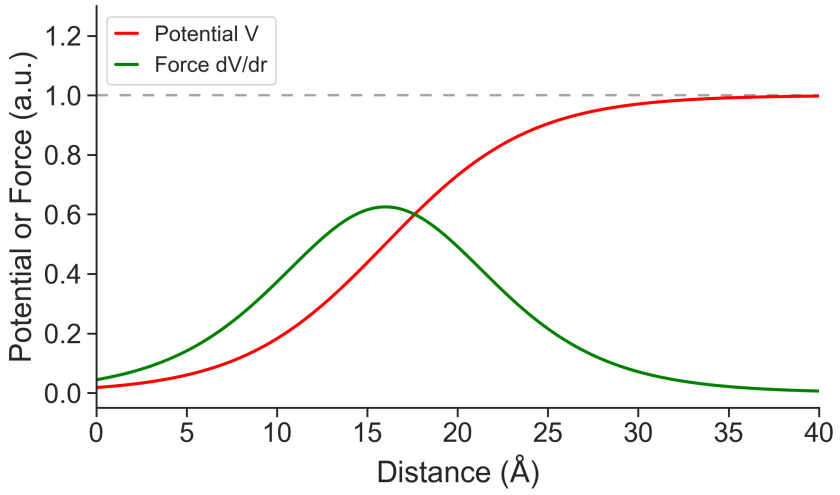

Figure 3.3. Sigmoid bias potential used in contact-guided REX MD. The sigmoid bias potential is defined by Eqs. 3.33 and 3.34. The illustrated shape corresponds to the optimized potential for proteins (cf. study in section 4.2), with the parameters $A = 1$, $\alpha = 25\,\text{Å}^{-1}$, $r_0 = 16\,\text{Å}$.

The synergy of α and r_0 leads to a potential that is distance dependant but locally confined. The resulting force (see green curve of Fig. 3.3) acts only locally up to a C_α-C_α threshold of approximately 32 Å, with the highest attractive force at 16 Å. This short-range limitation reduces the influence of erroneously used non-native contact pairs. Interactions above 32 Å distances are virtually neglected due to the asymptotic behavior of the sigmoid potential. Generally, this sigmoid potential can be used for proteins of any size as it will affect the selected contact pairs only in the mentioned region. This way, local structure motifs can get improved while unphysical compaction of structures are prevented on a larger scale. By choosing the coupling strength $\lambda = 10\,\text{kJ mol}^{-1}$, the resulting force is equivalent to a weak hydrogen bond at its maximum at $r_0 = 16\,\text{Å}$. With this the resulting interaction between single C_α-C_α pairs is of adequate strength. It drives the simulation towards native-like conformations but is sufficiently weak that it does not impose conformations[1]. A detailed instruction on how to implement the bias potential and bias contacts using `GROMACS` can be found in appendix A.2.

3.4 Structure Comparison

There are several methods to quantitatively measure protein structure similarity. These can be used to assess the quality of a new protein model, especially if an experimentally-determined structure already exists that can be used for reference.

Root-Mean-Square Deviation (RMSD)

The most popular structure comparison method is the root-mean-square-deviation (RMSD) of atomic positions. It is calculated as an averaged sum of atom distances between the *mobile* model and the *reference* model. Mathematically, this is described as

$$\text{RMSD(mob, ref)} = \sqrt{\frac{1}{N}\sum_i^N ||\boldsymbol{r}_i^{\text{mob}} - \boldsymbol{r}_i^{\text{ref}}||^2} \tag{3.35}$$

with the total number of atoms N and the atom positions $\boldsymbol{r}_i$. Usually both structures are first superimposed via translations and rotations which minimizes the RMSD. Note that the RMSD of two structures also depends on the atom selection, which is used for calculation. Typically, RMSDs are calculated using either the protein's backbone or solely the C_α atoms. Additionally, if a native conformation was used for reference, it is possible to categorize the mobile structure as being folded or unfolded based on its RMSD value. The main issue of RMSD-based structure comparison is that the RMSD value scales with the largest atom displacement [128]. If the majority of two models are perfectly aligned except for a small region (e.g. tail or loop region), then the RMSD can get disproportionally large indicating low similarity.

Global Distance Test (GDT)

An alternative method to quantitatively compare protein structures is the so-called global distance test (GDT) [129,130]. Similar to RMSD, both mobile and reference structure are first superimposed while considering translations and rotations. The structural similarity is then measured by calculating the percentage of C_α atoms that are found within a certain distance cutoff. This is done for 20 consecutive distances $d = (0.5\,\text{Å},\ 1.0\,\text{Å},\ 1.5\,\text{Å},\ ...,\ 10.0\,\text{Å})$. The resulting GDT curve (cf. Fig. 3.4A), which is obtained by mapping of calculated percentages on the x-axis vs. the distance cutoffs on the y-axis, visually captures the underlying structure similarity. Or more specifically, highly similar structures are represented by flat GDT curves that appear on the far right side, since they yield the highest sum of percentages. Furthermore, GDT curves can be used to calculate GDT scores, which numerically describe similarity of two protein structures. The two most common scores are the total score (TS),

$$GDT_{\text{TS}} = \frac{1}{4}\left(P_1 + P_2 + P_4 + P_8\right) \in [0, 100] \tag{3.36}$$

and the high-accuracy (HA) score,

$$GDT_{\text{HA}} = \frac{1}{4}\left(P_{0.5} + P_1 + P_2 + P_4\right) \in [0, 100] \tag{3.37}$$

where P_x denote the percentage of residues with displacements below a distance cutoff of x Å. In simple terms, these scores can be interpreted as the structural overlap of both protein models at different accuracy scales. Note that RMSD- and GDT-based evaluations apply anti-proportional metrics, where similar structures yield low RMSD values but high GDT scores. "A score above 90 is considered roughly equivalent to the experimentally determined structure" [100]. Since GDT-based structure comparison can take local misalignments better into account, it is preferably used in the Critical Assessment of Structure Prediction (CASP) [28,29]. This bi-annual event evaluates the performance of current state-of-the-art techniques for protein structure prediction and protein structure refinement. I summarized the latest CASP14 refinement results in Fig. 3.4B. This figure captures the performance of currently used structure refinement protocols by focusing on the GDT range of the 10 best results and the corresponding mean scores. Additionally, I highlighted the region that contains 95% of all depicted scores in yellow, thus indicating the typical range of GDT TS scores.

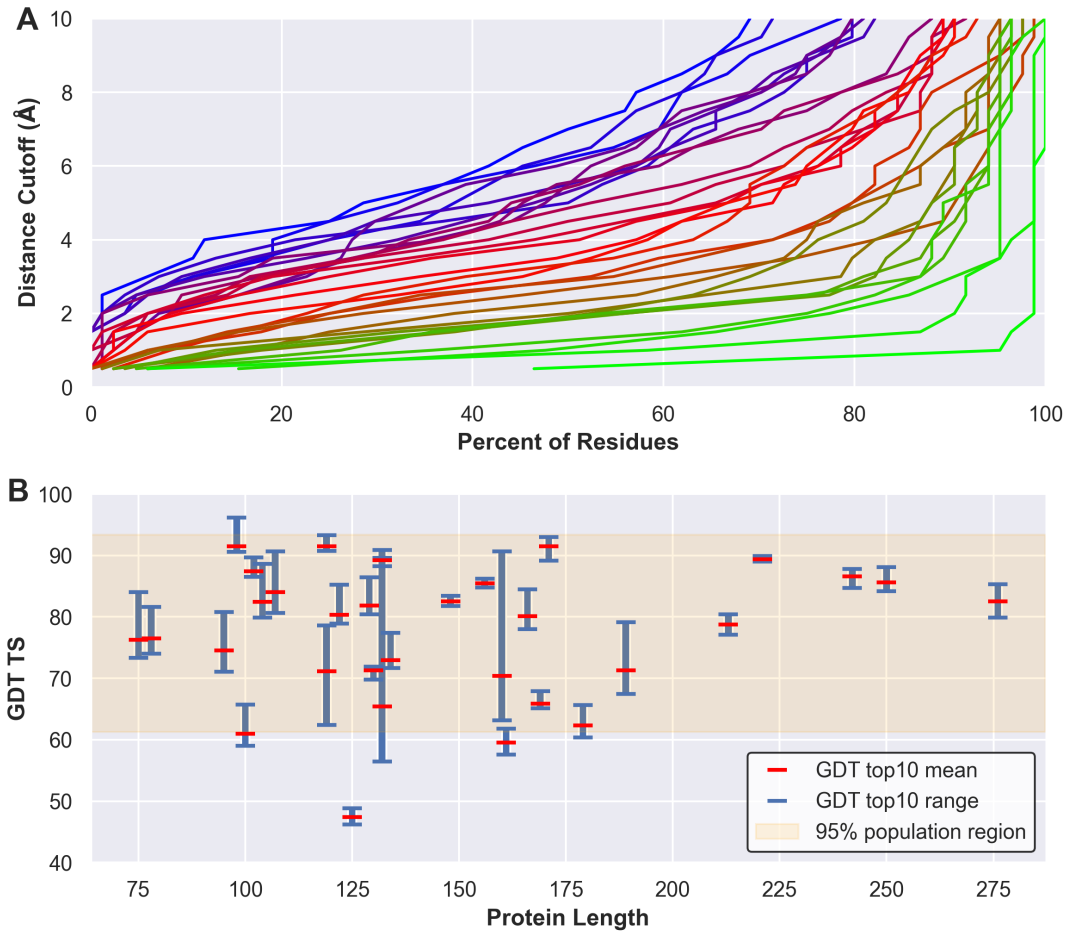

Figure 3.4. Global distance test results. **(A)** Exemplary GDT curves which can be used to compare protein structures. Curves are colored according to the spectrum (blue, red, green) to indicate (low, medium, high) structural similarity. **(B)** Top10 refinement results of CASP14[131]. The results of each protein target are indicated by a GDT score range (blue) and the corresponding mean score (red). Additionally, the yellow region contains 95% of all depicted GDT scores and indicates the region of typical GDT scores.

3.5 Distance Metric

Let X be an arbitrary set. A function $d\colon X \times X \to \mathbb{R}$ is called metric, if all $x, y, z \in X$ fulfill the axioms[132]:

$$\begin{aligned} &1)\ d(x,y) \geq 0 \quad \text{and} \quad d(x,y) = 0 \Leftrightarrow x = y && \text{(positive definiteness)} \\ &2)\ d(x,y) = d(y,x) && \text{(symmetry)} \\ &3)\ d(x,z) \leq d(x,y) + d(y,z) && \text{(triangle inequality)} \end{aligned}$$

The two most common distance metrics in computer science are the Euclidean distance and the Manhattan distance, which is especially preferred for very large and high-dimensional data sets. For two points $p = (p_1, ...p_n)$ and $q = (q_1, ..q_n)$ in n dimensions, the Euclidean distance is given by

$$d(p,q) = \sqrt{\sum_i^n (p_i - q_i)^2}, \tag{3.38}$$

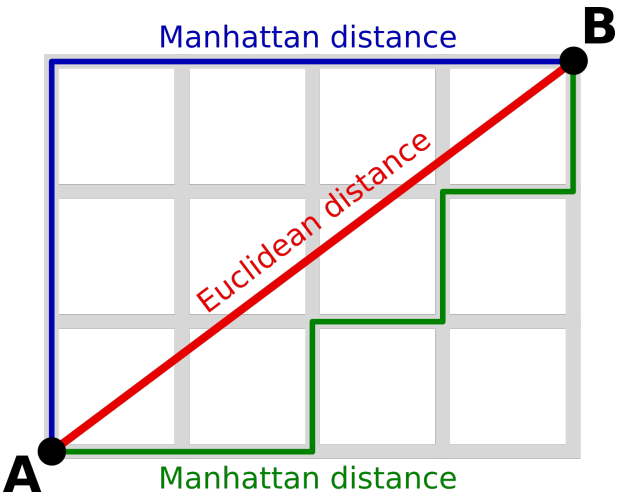

Figure 3.5. Most common distance metrics. Depicted are two points A and B and their distances. The Euclidean distance (red) is $d(A,B) = \sqrt{4^2+3^2} = 5$ according to Eq. 3.38. The Manhattan distances (green and blue) are both $d(A,B) = 7$ according to Eq. 3.39. Note that the Euclidean metric has exactly one possible path between A and B, whereas the Manhattan metric allows multiple paths with the same distance.

and the Manhattan distance is given by

$$d(p,q) = \sum_{i}^{n} |p_i - q_i|. \tag{3.39}$$

The naming is derived from Manhattan's street layout in form of a grid. Fig. 3.5 captures the difference of both metrics in a 2D plane.

3.6 Dimension-Reduction Algorithms

Dimension(ality) reduction is the transformation of a higher-dimensional data set into lower dimensions. Exploratory data analyses of feature-rich data can be quite difficult. Especially if variables/features have different scales, mixed types (e.g. numerical or categorical), or are correlated with each other. Due to the large input size such correlations can be hard to detect. To deal with this problems, it is possible to systematically check all pairwise correlations, eliminate certain features or combine them into new ones. However, such an approach is very cumbersome, time consuming and prone for errors. Instead, it is also possible to perform a dimension reduction on the existing data set, which projects the given information into a lower-dimensional embedded space, typically with just two of three dimensions. Each algorithm is designed to maintain specific aspects (e.g. distance information) of the initial data, which makes their application case-dependant. Existing dimension-reduction methods can be categorized into:

1) linear and nonlinear based on the underlying data transformations [133,134].
2) feature selection and feature extraction based on the application purpose [135].

Dimension reduction is especially useful for data visualization and data clustering. It can also reduce noise or correlations during the transformation process, which can improve the classification accuracy [136]. In some parts of my work I apply dimension-reduction techniques (TSNE and MDS), as I can use the resulting representations and their associated features to my advantage.

TSNE

T-distributed stochastic neighbor embedding (TSNE) [137,138] is a nonlinear dimension-reduction method which excels in the visualization of high-dimensional data. Using a statistical approach, it first captures the mutual distances of high-dimensional data points via Gaussian probabilities. The original variant, i.e. stochastic neighbor embedding (SNE) [139], models the low-dimensional representation with Gaussian probabilities as well. In TSNE, however, these are modelled with heavy-tailed t-distributions, which allows a better separation of similar and dissimilar objects. It makes TSNE "much easier to optimize, and produces significantly better visualizations by reducing the tendency to crowd points together in the center of the map." [137]

During the algorithm, TSNE performs a stochastic gradient descent [140] to optimize the Kullback-Leibler divergence [141,142], a cost function which measures the difference of two probability distributions. Finally, data points are projected onto lower dimensions by being either pushed away or pulled together according to their relative position on the t-distributions reflecting their (dis)similarity. This algorithm design is the biggest strength of TSNE, since the resulting low-dimensional representations visually contain data clusters which makes them easy to interpret. Adjacent data points of each native cluster are highly similar, whereas distant clusters convey differences and can be spotted immediately. This intrinsic property of TSNE representations is very convenient and can be utilized in combination with classification or clustering techniques to further improve the results.

Although distance information is not conserved during TSNE it can still be interpreted to a certain degree. For example, each native cluster has its own *distance norm*: Distances within a cluster can be interpreted to a large extent as being proportional to the difference of containing data points. However, distances between different clusters cannot be interpreted the same way because the final cluster position is a consequence of the TSNE algorithm iterating over multiple and very different t-distributions.

TSNE is a very versatile dimension-reduction method. It was successfully used for bioinformatics [143], cancer research [144], image classification [145], natural language processing [146], biomedical signal processing [147], music analysis [148], and many other fields.

Detailed Algorithm Instructions

TSNE "starts by converting the high-dimensional Euclidean distances between datapoints into conditional probabilities that represent similarities. The similarity of datapoint x_j to datapoint x_i is the conditional probability, p_{ij}, that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i. For nearby datapoints, p_{ij} is relatively high, whereas for widely separated datapoints, p_{ij} will be almost infinitesimal." [137] In the higher-dimensional space this probability is defined as

$$p_{j|i} = \frac{\exp(-||x_i - x_j||/2\sigma^2)}{\sum_{k \neq l} \exp(-||x_k - x_l||/2\sigma^2)}, \tag{3.40}$$

with the Gaussian variance σ. As the conditional probabilities are symmetric, define the joint probabilities p_{ij} as

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \tag{3.41}$$

where N is the number of high-dimensional data points. Note that p_{ij} has the properties

$$\sum_j p_{ij} = 1, \tag{3.42}$$

$$p_{ij} = p_{ji}, \tag{3.43}$$

$$p_{ii} \stackrel{!}{=} 0. \tag{3.44}$$

p_{ii} is specifically set to zero because the motivation is to model pairwise similarities.

The same concepts are applied for the lower-dimensional mapping points y_i. In TSNE, however, their probability is modelled with a heavy-tailed t-distribution with one degree of freedom (which is the same as a Cauchy distribution)[137]. The joint probabilities q_{ij} of lower-dimensional points y_i are thus given by

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}. \tag{3.45}$$

By minimizing the cost function C (here the Kullback–Leibler divergence between the distributions P and Q), the positions of the mapping points y_i are optimized to reflect the initial data (dis)similarity. Mathematically, the cost function is given by

$$C = D_{KL}(P \,||\, Q) = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right), \tag{3.46}$$

and its gradient by

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1}. \tag{3.47}$$

MDS

Multidimensional scaling (MDS)[149,150] is a form of non-linear dimension reduction based on multivariate analysis. Its main objective is to project high-dimensional data into lower dimensions for a better interpretation, while preserving the pairwise *dissimilarity* of samples. To achieve this, MDS requires a distance matrix reflecting the dissimilarity of all samples. There are many variations of MDS, such as *classical, metric and non-metric*. Classical MDS aims to solve the dimension reduction via change of basis using eigenvectors and eigenvalues. Metric MDS on the other hand applies numerical optimization techniques to approximate the dissimilarity of samples which can be measured with a metric (see section 3.5). Non-metric MDS is an extended approach, where the dissimilarities are given by non-parametric relationships (e.g. categorical relationship).

The distance preservation of MDS makes it a popular dimension-reduction method. It was successfully used in archaeology[151], biophysics/biochemistry[152], nuclear physics[153], marketing research[154], political science[155], psychology[156], etc. In my work I apply metric MDS on Euclidean distance matrices as input.

Detailed Algorithm Instructions

Metric Multidimensional Scaling (mMDS):

1. Given n objects and their dissimilarity represented by a $n \times n$ distance matrix using any metric, assign n points in $m < n$ dimensions at positions $y_i = (y_1, ..y_m)$. To obtain a representation that is better suited for interpretations m is typically set to 2 or 3.
2. Calculate the Euclidean distances between all points y_i in the low-dimensional space.

3. Compare the pairwise dissimilarities between the high-dimensional and low-dimensional representations. So-called *Stress* functions (see Eq. 3.48) are suitable *goodness-of-fit measures* to capture the difference between initial and projected distances.
4. Update the positions of points y_i using an optimization algorithm.

Repeat steps 2-4 until the Stress function is minimized.

Metric MDS aims to minimize a cost function C, which is typically called *Stress* (STandardized REsidual Sum of Squares). Mathematically it is given by

$$C = Stress = \left(\sum_{i<j} (\delta_{ij} - d_{ij})^2 \right)^{1/2}, \tag{3.48}$$

with δ_{ij} as the dissimilarity metric of the high-dimensional input data, $d_{ij} = ||y_i - y_j||$ as the Euclidean distance of the low-dimensional target data. A solution can be found with numerical optimization techniques such as gradient descent [157]. Let y_i be the low-dimensional vector that represents the coordinates of object i in $m < n$ dimensions. The gradient of C is then given by

$$\frac{\partial C}{\partial y_i} = -2 \sum_j \frac{\delta_{ij} - d_{ij}}{d_{ij}} (y_i - y_j). \tag{3.49}$$

Some MDS studies apply different Stress functions [150,158,159], which are weighted variations of Eq. 3.48. For instance, it is possible to give more weight to smaller distances than to larger distances, which corresponds to a non-uniform distance preservation.

3.7 Clustering Algorithms

Clustering is an unsupervised technique which aims to group objects taken from a data set based on similarity. Clustering does not transform the underlying data set. It just analyzes it and provides additional information to the user by assigning a cluster label to each object, such that similar objects share the same label. This can be useful for various applications. For example, data visualizations can be improved by choosing colors based on the cluster labels. It is also possible to filter and select only specific data objects (i.e. with pre-defined cluster labels) for further tasks during a complex algorithm. As a final example, clustering can be useful for outlier detection [160].

Clustering algorithms can be classified based on their approach [161,162], such as

1) Partitional clustering: K-Means [163,164], K-Medoids [165,166], etc.
2) Density-based clustering: DBSCAN [167,168], OPTICS [169], etc.
3) Grid-based clustering: WaveCluster [170], etc.
4) Model-based clustering: Gaussian mixture models [171], etc.
5) Hierarchical clustering: There are two ways to generate a dendrogram, namely
   - Agglomerative (bottom-up approach)
   - Divisive (top-down approach)

In some parts of my work I apply K-Means (from now on written as KMEANS) or DBSCAN to cluster my data.

KMEANS

KMEANS[163,164] is one of the most popular clustering methods. Its goal is to partition a given data set into k clusters, which basically divides the sampling space into k Voronoi cells[172,173]. While the general idea of KMEANS is very simple, there are many variations of the KMEANS algorithm[164,174]. They can differ, e.g., in their initialization method[175,176], optimization method, distance metric, etc. The weakness of KMEANS is that the results heavily rely on the underlying data set and the used initialization method. KMEANS has two important parameter specifications:

- k: Number of clusters.
- n_{init}: Number of KMEANS runs with independent initializations. Only the run with the lowest sum of squared errors SSE (sometimes referred to as *inertia* or *distortion*) will be selected as final KMEANS result.

Good KMEANS models have low values for k and SSE. A heuristic to determine the optimal value for k is the so-called *elbow method*[177]. By plotting k on the x-axis vs. SSE on the y-axis, an optimal k value can be derived by the point where diminishing return is observed. This point is also called *elbow* due to the shape of the curve. However, it is not always possible to clearly identify an elbow on the resulting plot[177].

In my work I apply Lloyd's KMEANS algorithm[164] with k-means++ seeding[178,179]. This initialization method cleverly selects k random centroids as initial cluster centers and guarantees to find a KMEANS solution that is "$\mathcal{O}(\log k)$-competitive with the optimal clustering."[178]

Detailed Algorithm Instructions

k-means++ initialization[178]:

1a. Select first cluster center $M = \{\mu_1\}$ uniform at random from all data points $X = \{x_1, .., x_n\}$.
1b. For each point $x \in X \backslash M$: calculate distance $D(x)$ to its nearest cluster center.
1c. Select $x_i \in X \backslash M$ as a new cluster center $\mu_i \in M$ according to the probability

$$p(x_i) = \frac{D(x_i)^2}{\sum_{x \in X \backslash M} D(x)^2}. \tag{3.50}$$

Repeat steps 1b and 1c until $M = \{\mu_1, ..., \mu_k\}$ has k members.

Lloyd's KMEANS algorithm[164,180]:

1. Select k initial cluster centers $M = \{\mu_1, ..., \mu_k\}$ (in my case with k-means++).
2. Assign all d-dimensional data points $X = \{x_1, .., x_n\}$ to their nearest cluster center μ_N:

$$N = \underset{k}{\operatorname{argmin}} ||x_i - \mu_k||^2 = \underset{k}{\operatorname{argmin}} \sum_{j=1}^{d} (x_{ij} - \mu_{kj})^2 \tag{3.51}$$

3. Recalculate each cluster center (*centroid*) μ_i based on the points x that belong to its cluster C_i:

$$\mu_i^{\text{new}} = \frac{1}{|C_i|} \sum_{x \in C_i} x, \tag{3.52}$$

Repeat steps 2 and 3 until convergence of C.

Result Selection[180]:
The results of each individual KMEANS run are analyzed with respect to the cluster variances. The best clustering results yield the lowest sum of squared errors SSE, hence the objective is to minimize

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2, \tag{3.53}$$

with the cluster centers $M = \{\mu_1, ..., \mu_k\}$, and the points x that belong to each individual cluster C_i.

DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) [167,168] is a density-based clustering method that can differentiate between meaningful cluster data and noise. DBSCAN can technically apply any distance metric and its algorithm is deterministic, i.e. multiple iterations with the same parameters yield the same results. Its main weakness is that the algorithm-defining parameters are typically case-specific and thus require fine tuning. DBSCAN has two important parameter specifications:

- ε : neighborhood distance (search radius).
- min_{pts} : density threshold. Specifies how many points within ε around sample X are required to consider X as a core point and part of a cluster.

Note that both parameters are correlated with distance, which makes DBSCAN highly parameter dependant. More precisely, the underlying data set (i.e. the density and distance between samples) define a reasonable parameter range for meaningful clustering results. However, some DBSCAN variations can automatically determine good parameters [181].

Detailed Algorithm Instructions

Abstract DBSCAN algorithm[168]:

1. For each point: find neighbors within ε and identify core points based on min_{pts}.
2. Join neighboring core points into clusters C_i.
3. For each non-core point:
   - If point is neighbor of a core point, then assign it to its cluster C_i.
   - Otherwise assign it to noise.

These instructions allow to identify a non-specified number of dense data clusters, which grows naturally due to the connection of adjacent core points and their border points. Samples, which do not satisfy the density threshold are filtered out as noise. This classification effectively reduces the sampling space of each DBSCAN application because noise points are typically left out in further analyses.

Fig. 3.6 illustrates the concept of DBSCAN clustering and the classification of samples into core points, border points and noise.

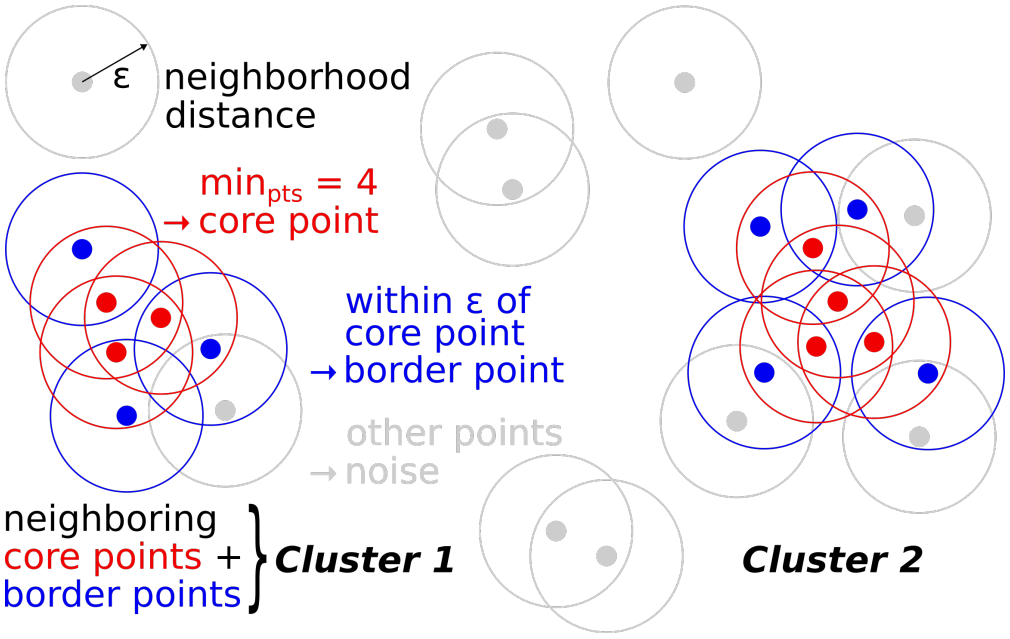

Figure 3.6. Illustration of DBSCAN algorithm. Figure explains the classification of data points for DBSCAN based on the parameters ε (*neighborhood distance*) and $min_{\text{pts}} = 4$ (*density threshold*). If a point has min_{pts} samples within its radius ε, then this point is considered a *core point* (red). Reachable neighbors within ε of core points are *border points* (blue). All other points are considered *noise* (grey). Each cluster is defined by neighboring core points and their border points. Licensed by Arthur Voronin under CC BY-SA 4.0

PART II

METHOD DEVELOPMENT

4

Contact-Guided Replica Exchange Molecular Dynamics

This chapter covers various aspects related to contact-guided REX MD and showcases it as a powerful but yet easy-to-use tool for biomolecular studies. REX, as an enhanced sampling method, can generate large amounts of physically meaningful structure ensembles while maintaining correct thermodynamic properties. The additional integration of a contact-driven bias potential largely reduces the conformational search space during REX. This speeds up the process of finding native-like conformations and lowers the overall computational costs. Section 4.1 demonstrates the general benefits of contact-guided REX MD in contrast to regular MD. Additionally, I compare different scenarios by varying the bias-quality of considered contacts. By doing so, I learn how robust the method is against noisy bias data and what the most influential factors are. In particular, I deduce conditions for selected bias contacts in order to obtain optimal simulation results. Based on these findings I formulate bias guidelines which should be considered before applying contact-guided REX MD for research studies. Section 4.2 focuses on a study to determine optimal parameters for the applied sigmoid potential, resulting in a distance-dependant attractive force between biased contact pairs. I investigate the potential's conformation-guiding strength for different parameters λ and r_0 when applied on purely α-helical or β-sheet structures. By comparing the resulting structure quality of each individual REX trajectory, I can infer an optimized bias potential.

4.1 Bias-Quality Study

This section introduces contact-guided REX MD as a method to generate native-like protein structures. The presented "prototype" study is used to demonstrate the benefits of contact-guided REX MD over regular MD. A systematic analysis of over 28 REX simulations using different bias quality allows to assess the bias influence and extract guidelines for an optimal performance. First, I outline the study concept in 4.1.1 where I lead into the main motivation, explain the used bias-quality variations and introduce the studied protein systems. Next, in 4.1.2 I provide a detailed overview of the generated structure ensembles for Trp-Cage as my first test protein. Using two different metrics (root-mean-square-deviation (RMSD) and global distance test (GDT)), I am comparing the individual REX simulations against each other and identify the optimal combination of used bias contacts and true-positive rate. Analogously, section 4.1.3 contains analyses for my second test system, the Villin Headpiece. At last I conclude all my findings in 4.1.4 by reviewing the most influential effects when considering bias contacts and present bias guidelines to obtain optimal performance during REX MD. This section is based on my article named "Including residual contact information into replica-exchange MD simulations significantly enriches native-like conformations" (2020)[1], published by PLOS ONE.

4.1.1 Study Concept

It is often uncertain how error-prone available contact information actually is. For instance, theoretic methods such as direct coupling analysis (DCA)[49–51] can predict spatially close residue pairs and assign them a confidence score. However, even the highest ranking of predicted adjacencies (from now on referred to as *contacts*) can contain false-positive cases. Similar to this, contacts derived from, e.g., NMR can have wrong assignments. For this reason, I investigated how exactly native contact information and its bias quality affects the results of contact-guided REX MD. More precisely, I performed an extensive study to analyze the influence of both native ("correct") and non-native ("wrong") contact information with regard to structure refinement. To achieve this I randomly selected contact pairs of known test structures and generated two lists, each containing only native or non-native contacts, respectively. In a subsequent step, I constructed different study scenarios by choosing contacts from both lists and specified the number of used contacts and the true-positive rate (TPR) of each scenario. To quantify the TPR of selected contact pairs, I consider them as native if they fulfill the two following conditions[1]:

$$r_{ij} = |r_i - r_j| \leq 6\text{Å} \equiv r_{\text{nc}}, \tag{4.1}$$

$$\Delta ij = |i - j| \geq 4. \tag{4.2}$$

Eq. 4.1 defines native contacts with a distance threshold for C_α distances r_{ij} being at 6 Å for two residues i and j. Eq. 4.2 on the other hand excludes short-range pairs relative to their sequence position. Such contacts appear as the main diagonal on the contact map and often correspond to one revolution of a α-helix, which makes them irrelevant as a bias. By comparing the different scenarios using purely-native or mixed contacts, I am able to answer the central question: *What is the required bias quality for an improved REX performance?*

Starting from an unfolded state, I used different sets of randomly selected contact pairs to enrich REX MD simulations. I investigated 14 different cases of varying bias quality (cf. Table 4.1), which are specified by the total number of selected contacts and their true-positive rate (TPR). In order to differentiate the conformation-guiding strength relative to the selected bias contacts, I applied a fixed coupling parameter of $\lambda = 10\ \text{kJ mol}^{-1}$ (see Eq. 3.33) during all simulations.

Table 4.1. Variation of bias quality in performance study using contact-guided REX MD[1]. Overview of the 14 REX MD scenarios investigated in the performance study for both test proteins. Listed are the true-positive rate (TPR) of used contact pairs (CP) in percent, number of restraining CP used, number of native contacts, and number of non-native contacts. A visualization of the used contacts can be looked up in appendix Figs. B.1 to B.6.

TPR (%)	ref	100	100	100	100	100	75	75	75	75	50	50	50	50
# CP	0	6	12	24	36	48	12	24	36	48	12	24	36	48
# native	0	6	12	24	36	48	9	18	27	36	6	12	18	24
# non-native	0	0	0	0	0	0	3	6	9	12	6	12	18	24

To minimize deviation-effects even further all replicas start in the same unfolded state. Each simulation generated 250 ns long REX trajectories over a wide temperature range from $T_0 = 300\,\mathrm{K}$ to $T_{max} \approx 625\,\mathrm{K}$. The extremely large temperature range was chosen intentionally to guarantee sufficient energy and to overcome any potential energetic barriers. It also encourages large-scale conformational transitions during each turnaround cycle before cooling down again. However, I primarily focused on analyzing the lowest-temperature replica because theoretically this is where the lowest-energy states are expected. Additionally, by comparing all test cases to a reference simulation without any contact information, I can estimate guidelines for an optimal bias, i.e. the number of required restraints and the resulting bias strength. The comparison with the reference case also shows how significant contact information is and to what extent it can guide proteins towards native-like conformations. Lastly, to quantify the improvement of contact-guided REX MD over regular MD, I performed additional MD simulations with and without bias.

The study involved two very small and fast-folding proteins, namely Trp-Cage (PDB id: 1l2y[182]) and Villin Headpiece (VHP, PDB id: 1vii[183]). Trp-Cage has only a length of 20 residues and its tertiary structure consists of an α-helix followed by a turn and a 3/10-helix. This designed mini-protein reaches folding times of approximately $4\,\mu\mathrm{s}$ [184] and is therefore among the fastest folding proteins. Its folding temperatures are reported to be in the range of 311 to 317 K[185]. Villin Headpiece has a sequence length of 35 residues and consists of three α helices. Its folding times are in the order of μs[186,187], whereas the folding temperatures are between 339 and 342 K[188]. Fig. 4.1 illustrates the initial and native conformations of the used proteins. Used replica temperatures can be found in appendix B. Other study-related details, such system setup or used hardware, can be looked up in Ref.[1].

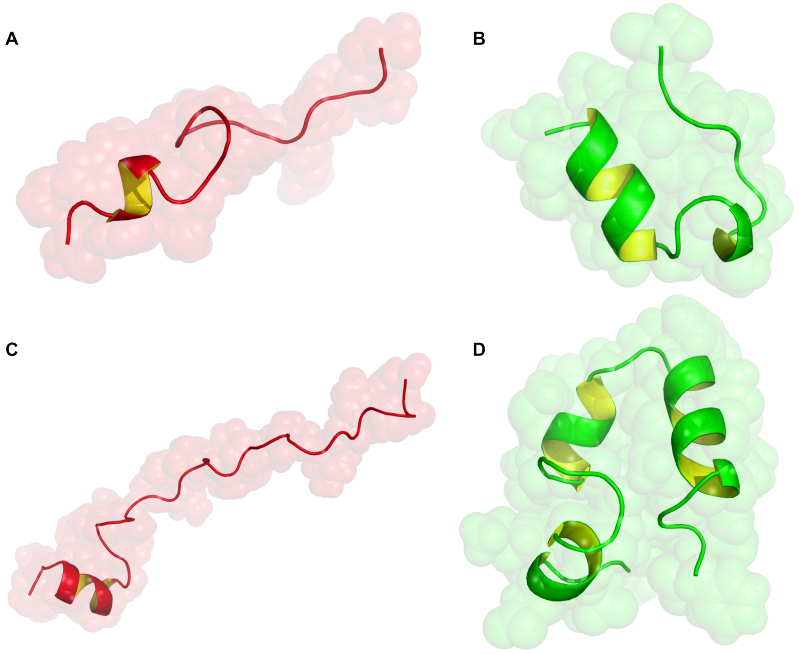

Figure 4.1. Protein structures of bias-quality study using contact-guided REX MD.
Initial (A) and native (B) conformation of Trp-Cage with a backbone RMSD difference of 8.8 Å. Analogously, initial (C) and native (D) conformation of VHP with a difference of 16.2 Å. Initial conformations were manually selected from short MD simulations at $T = 500\,\mathrm{K}$ due to their minimal amount of secondary structure motifs. Visualized in pyMOL[189,190]. Adapted from Ref.[1] under CC BY 4.0.

4.1.2 Trp-Cage Simulations

Using my modified REX temperature generator (see Eqs. A.5 and A.6), I was able to achieve nearly constant exchange rates of 16% using 60 replicas over the wide temperature range of 300 K to approximately 625 K. The difference between the starting conformation, which was used for all replicas, and the native conformation has a backbone RMSD of 8.8 Å.

A starting point of the discussion can be achieved by comparing the performance of MD and REX MD, especially under consideration of a bias integration. For this purpose, I looked at the following four generalized cases:

- MD without bias ("MD ref")
- MD with perfect bias at 100% TPR and 12 contact pairs ("MD 100% 12cp")
- REX MD without bias ("REX MD ref")
- REX MD with perfect bias at 100% TPR and 12 contact pairs ("REX MD 100% 12cp")

Fig. 4.2 summarizes the comparison of these cases and shows the backbone RMSD time evolution at the lowest-temperatures. The histograms on the right side of the figure display the RMSD range and how often each RMSD value occurs. The figure evidently captures the tendencies of the four generalized cases and indicates the benefits of both the integration of a contact bias and REX MD as an enhanced sampling method. In the case of the 500 ns long unbiased MD simulation, the RMSD curve shows a random behavior. As expected, the protein undergoes many conformational changes especially since the system temperature is close to the reported folding temperature (311-317 K[185]).

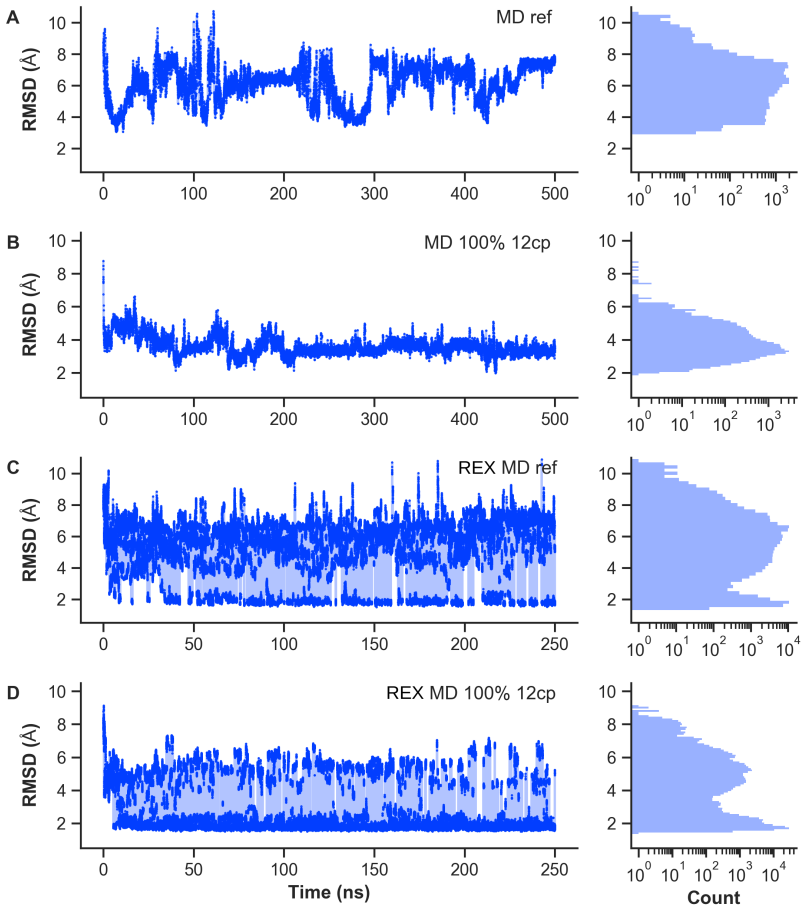

Figure 4.2. Comparison of Trp-Cage MD and REX MD simulations. Backbone RMSD time evolution at lowest-temperature replica and corresponding histogram with logarithmic count axis. Values were taken from 500 ns MD and 250 ns REX MD trajectories, respectively. **(A)** MD reference simulation without additional bias. **(B)** MD simulation with 12 native contact restraints. **(C)** REX MD reference simulation without additional bias. **(D)** REX MD simulation with 12 native contact restraints. Adapted from Ref.[1] under CC BY 4.0.

Due to the histogram we can verify that the simulation contains conformations with RMSDs between 3 and 10.5 Å. The highest observed frequency is approximately 7 Å, which is just 1.8 Å smaller than the starting value. As soon as the contact bias gets turned on ("MD 100% 12cp" simulation), we can see a drastic shift towards smaller RMSD conformations. However, due to the nature of MD we can see that the simulation gets trapped in a conformational state with approximately 4 Å for the majority of the trajectory. When looking at the unbiased REX MD simulation, we can observe a wide ensemble of structures. In contrast to MD, this simulation already shows many well-refined structures with RMSDs below 2 Å. Similar to before, as soon as the bias contacts are integrated ("REX MD 12% 12cp" simulation), the distribution of observed conformations shifts towards smaller RMSDs and native-like structures are significantly enriched. Even though the best observed structures are equally good among both REX scenarios, we can clearly see that contact-guided REX MD generates trajectories with much better RMSD statistics. The overall improvements over regular MD justifies the increased computational costs of contact-guided REX MD resulting from simulating multiple replicas in parallel. Especially when taking into account that REX MD should theoretically generate native structures in a single run, whereas regular MD usually requires multiple runs and cannot guarantee good results.

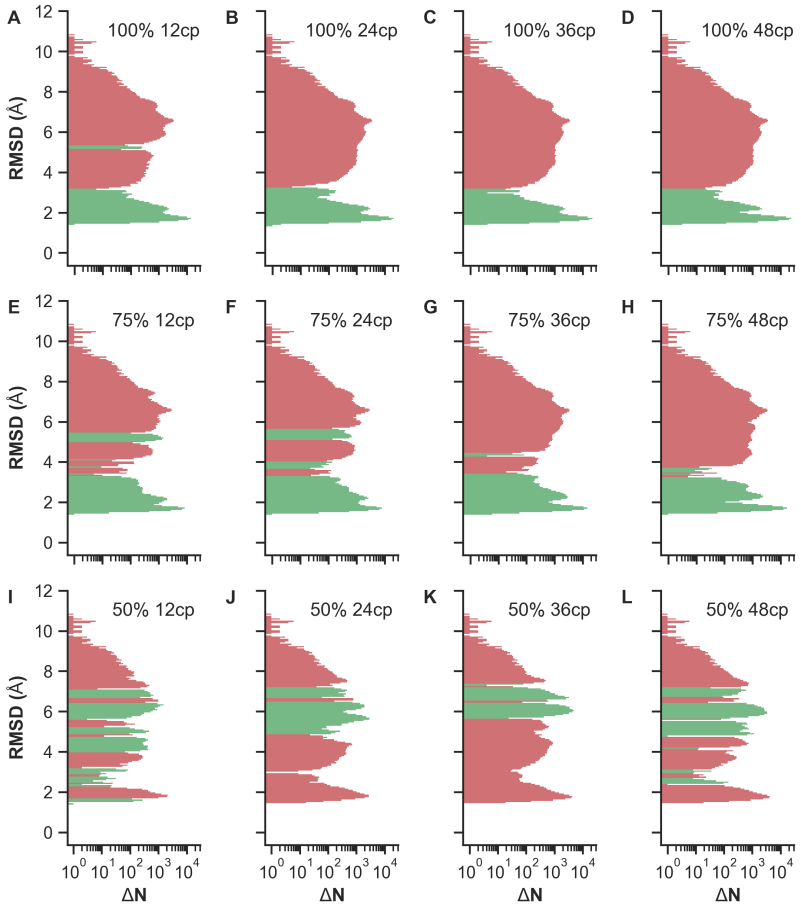

Figure 4.3. ΔN histograms of Trp-Cage REX MD simulations. Histograms show the enrichment and depletion of conformations with a particular backbone RMSD at $T_0 = 300\,\mathrm{K}$ as compared to the reference. Histogram bins are defined by the RMSD axis, while the logarithmic ΔN axis illustrates the count difference between the tested REX MD cases and the REX MD reference simulation with $\Delta N = N_{\mathrm{case}} - N_{\mathrm{ref}}$. Positive (negative) values corresponding to enrichment (depletion) are shown in green (red). **(A-D)** Simulations with 100% TPR and 12, 24, 36, 48 contact pairs. **(E-H)** Simulations with 75% TPR and 12, 24, 36, 48 contact pairs. **(I-L)** Simulations with 50% TPR and 12, 24, 36, 48 contact pairs. Reproduced from Ref.[1] under CC BY 4.0.

Until now the discussion only considered simulations with perfect bias which consisted only of native contacts. Theoretically or experimentally derived contact information is however error-prone and the TPR typically declines with increasing number of contacts (cf. Fig. 7.2). As previously mentioned, a critical question of this study is: *What is the required bias quality for an improved REX performance?* A quantitative and illustrative method to answer this can be achieved by so-called ΔN histograms. Such histograms display the count difference of backbone RMSDs between all tested REX cases with respect to the unbiased reference REX simulation, i.e.:

$$\Delta N = N_{\mathrm{case}} - N_{\mathrm{ref}}. \tag{4.3}$$

This means that conformations are occurring more often if ΔN is positive, i.e. such conformations get enriched. Vice versa, if ΔN is negative then corresponding conformations get depleted. Fig. 4.3 summarizes the performance of all Trp-Cage REX MD simulations by a grid of ΔN histograms, where each sub-figure corresponds to one case comparison. As evidently shown in Fig. 4.3(A-D), integration of a perfect bias with 100% TPR leads to a strong enrichment of low RMSD conformations between 1.6 and

3.0 Å, indicated by the green-colored bins. At the same rate, conformations with RMSDs between 3 and and 11 Å get depleted and appear less often. As expected, this enrichment-effect gets stronger when more contacts are used. Please note that in case of Trp-Cage, there is almost no difference in simulations using more than 24 native contacts, which is in the order of the protein length L. Besides, there is no large gain of native-like folds when the bias exceeds 12 restraints, corresponding to approximately $L/2$ contact pairs. The comparison of simulations using mixed bias contacts at 75% TPR shows similar results, as portrayed by Fig. 4.3(E-H). The main difference to previously discussed cases is, that by integrating 12 or 24 mixed bias contacts we can also observe an enrichment of conformations with around 5 Å RMSD. This unwanted conformation-guiding effect apparently results from the first few non-native contacts used for the REX simulations. As shown in Fig. B.2A, non-native contacts of the first two contact badges (visualized by cyan and green squares without grey border) are located far away from the main diagonal. Such long-range contacts typically are much more important and have a stronger effect for conformation-guiding effects. Furthermore, here the randomly-selected non-native contacts can also be grouped to a cluster of restraints. By design of my study, all bias contacts have the same coupling strength. However, if contact pairs are so close to each other within the contact map, the energetic bias does add up which increases the attractive force acting on the affected protein segments. In some cases this force can even become so strong that it traps the protein in specific conformations. Finally, looking at the performance of the last scenarios with bias restraints at only 50% TPR (cf. Fig. 4.3(I-L)), low-RMSD conformations get depleted which is contrary to the intention. Simulations with such a bad bias quality show a far worse RMSD statistics compared to the unbiased reference simulation. Therefore, it is safe to conclude that contact information with such low quality is inappropriate for the general use-case of contact-guided REX MD and should be avoided.

"A theoretical edge case occurs for equally contributing native and non-native contacts at 50% TPR. While half of the contacts (true-positives) would lower the global minimum, the other half (false-positives) would either lower existing local minima or introduce new unphysical ones. As the global minimum remains global under these circumstances, obtaining the native state is still possible with the help of enhanced sampling in REX MD. This edge case is, however, usually not met as the used contacts are not equally contributing due to the distance dependency"[1] of the sigmoidal bias potential. Besides, as previously mentioned, clusters of bias contacts can lead to protein structure entrapment in unfavorable conformations. To avoid such unwanted effects, it is necessary to analyze the locations of considered bias contacts, identify potential clusters and adjust the coupling strength of such bias contacts according to the cluster size.

So far the discussion and presented analyses were solely based on the RMSD metric. RMSD as a measurement to quantify the similarity of two different protein structures is very popular by researchers of this field. However, the disadvantage of RMSD-based evaluations is that it strongly correlates with the largest displacement of the two compared structures. E.g., consider two protein models with a 95% global alignment and 5% discrepancy resulting from the mobility of a short tail section. In such cases, the measured RMSD can reach disproportionately large values of multiple Å. Thus by representing the structure similarity as a single number, it is possible to get a false impression of the alignment quality even if the majority of both protein structures fits perfectly. One solution to this problem is given by the global distance test (GDT), which is another structure comparison method similar to RMSD. First, both considered protein models are spatially aligned. In order to estimate the relation of the structures, residual C_α atoms are mapped 1:1 and their calculated distances are compared to various cutoffs (0.5 Å, 1.0 Å, .., 10.0 Å). Finally, percentages of C_α distances below considered thresholds are calculated and used to calculate GDT scores via Eqs. 3.36 or 3.37. These GDT scores range from 0 to 100 and are less prone to local structural misalignments as compared to RMSD.

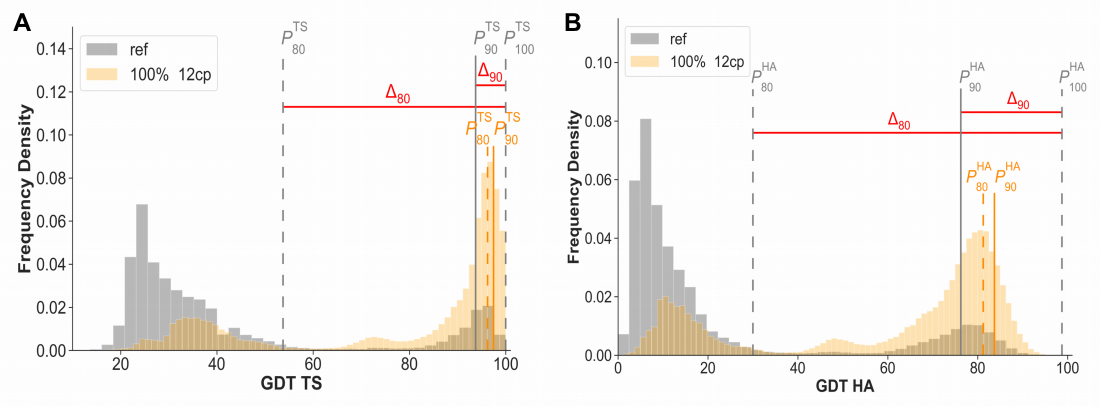

Figure 4.4. Comparison of global distance test (GDT) distributions of Trp-Cage simulations. Distributions are taken from REX MD and represent either the unbiased reference simulation (grey) or the biased scenario with 12 contact pairs at 100% TPR (orange). Figure also shows a selection of percentiles and their corresponding Δ_x difference (see Eq. 4.5), visualized by vertical or horizontal lines, respectively. **(A)** GDT total score (TS) distributions. **(B)** GDT high accuracy (HA) distributions.

For this reason, GDT is often applied in CASP[28,29] (Critical Assessment of Techniques for Protein Structure Prediction), a bi-annual world-wide experiment to rank and compare the accuracy of currently-applied methods for structure prediction and refinement. In most use-cases it is sufficient to apply the total score (TS) variant for structure analyses. However, in my case I intend to assess a bias quality threshold for an optimal performance using contact-guided REX MD. Therefore it is beneficial to consider the high accuracy (HA) variant as well, which can provide additional insight and unveil the limits of my method with regard to structure refinement precision. To answer this question, I analyzed the GDT_{TS} and GDT_{HA} distributions for all performed REX simulations at the lowest-temperature replica. Table 4.2 gives an overview of all performed Trp-Cage simulations and their individual structure refinement performance. This can be achieved by comparing representative GDT percentiles corresponding to native-like structures. To improve the readability of the table and capture its essence some table cells are visually modified[1]. For example, shaded table cells represent improved percentiles P_{x} compared to the reference REX simulation $P_{\mathrm{x|ref}}$. Such cells satisfy the condition

$$P_{\mathrm{x}} \geq P_{\mathrm{x|ref}}. \tag{4.4}$$

Additionally, a bold font highlights a significant improvement, which is defined by

$$P_{\mathrm{x}} \geq P_{\mathrm{x|ref}} + w \cdot \underbrace{(P_{100|\mathrm{ref}} - P_{\mathrm{x|ref}})}_{\Delta_x}. \tag{4.5}$$

Eq. 4.5 defines a relation of P_x and a percentile-dependant threshold indicating how much the considered percentile can improve relative to the reference simulation. The difference Δ_x (cf. Fig. 4.4) specifies the possible improvement range of the x-th percentile based on the reference simulation. Be choosing a weight factor w, we can define by how much the respective percentiles must increase to be considered *significant*. Here, I opted for $w = 0.5$ and defined a significant increase by 50%. Fig. 4.4 exemplarily shows the GDT distribution shift of Trp-Cage by comparing a biased REX simulation with the reference case. According to Table 4.2, observed GDT scores get drastically improved for REX simulations using contact information with at least 75% TPR. "Here, the TS distribution is clearly shifted from 53.75 to scores above 96 already at the 80th percentile. This means that 20% of the simulated structures in the trajectory already adopted conformations which are almost identical to the native fold.

Table 4.2. Total Score (TS) and High Accuracy (HA) percentiles of Trp-Cage simulations[1]. Overview of observed global distance test (GDT) percentiles. Statistics were taken from trajectories at $T = 300\,\mathrm{K}$ over 250 ns for REX MD and 500 ns for MD, respectively. Listed are the simulation method, the true-positive rate (TPR) in percent, used number of restraining contact pairs (CP), GDT total score percentiles (P^{TS}), and GDT high accuracy percentiles (P^{HA}). Values equal to or greater than the respective reference are shaded in gray. According to Eq. 4.5 significantly greater values are bold.

Method	TPR (%)	# CP	P^{TS}_{80}	P^{TS}_{85}	P^{TS}_{90}	P^{TS}_{95}	P^{TS}_{100}	P^{HA}_{80}	P^{HA}_{85}	P^{HA}_{90}	P^{HA}_{95}	P^{HA}_{100}
REX MD	ref	0	53.75	88.75	93.75	96.25	100.00	30.00	67.50	76.25	81.25	98.75
REX MD	100	6	**96.25**	**96.25**	**97.50**	97.50	**100.00**	**80.00**	81.25	82.50	85.00	96.25
REX MD	100	12	**96.25**	**97.50**	**97.50**	**98.75**	**100.00**	**81.25**	82.50	83.75	86.25	97.50
REX MD	100	24	**97.50**	**97.50**	**97.50**	**98.75**	**100.00**	**82.50**	**83.75**	85.00	86.25	97.50
REX MD	100	36	**97.50**	**97.50**	**97.50**	**98.75**	**100.00**	**82.50**	**83.75**	85.00	86.25	97.50
REX MD	100	48	**97.50**	**97.50**	**98.75**	**98.75**	**100.00**	**82.50**	**83.75**	85.00	86.25	97.50
REX MD	75	12	**95.00**	**96.25**	**97.50**	97.50	**100.00**	**78.75**	80.00	82.50	85.00	**98.75**
REX MD	75	24	**95.00**	**96.25**	96.25	97.50	**100.00**	**77.50**	80.00	81.25	83.75	96.25
REX MD	75	36	**96.25**	**96.25**	**97.50**	97.50	**100.00**	**80.00**	81.25	82.50	85.00	96.25
REX MD	75	48	**96.25**	**96.25**	**97.50**	**98.75**	**100.00**	**80.00**	81.25	83.75	85.00	97.50
REX MD	50	12	41.25	47.50	85.00	95.00	**100.00**	16.25	23.75	62.50	77.50	96.25
REX MD	50	24	40.00	42.50	47.50	91.25	**100.00**	17.50	20.00	23.75	71.25	96.25
REX MD	50	36	36.25	38.75	41.25	43.75	95.00	15.00	17.50	20.00	22.50	82.50
REX MD	50	48	37.50	38.75	42.50	46.25	93.75	15.00	17.50	18.75	23.75	76.25
MD	ref	0	33.75	36.25	40.00	43.75	56.25	11.25	13.75	16.25	20.00	32.50
MD	100	12	53.75	55.00	56.25	57.50	68.75	28.75	30.00	31.25	33.75	45.00

HA scores similarly show a significant improvement. It is particularly remarkable that the reference simulation yielded an exceptional HA score of 98.75."[1] REX simulations with 50% TPR show much worse GDT statistics compared to the reference simulations. In accordance with the previous RMSD-based analyses, we can see that such low-quality bias has a very negative influence for structure refinement. Instead of enriching native-like conformations this highly error-prone contact information either populates unfavorable conformations in a frustrated energy landscape or even traps the protein. All observations are equally true for both TS and HA distributions. According to Table 4.2, the best observed scenario was achieved with 12 contact pairs at 75% TPR when considering the HA percentiles. Note however, that there is no meaningful difference between the simulations with 75% or 100% TPR bias. In the case of Trp-Cage as a test system, these scenarios should be considered equally good as long as it is possible to confidently select well-refined structures out of the REX-generated ensembles. Besides, these simulations generated almost perfect native structures for more than 20% of the entire REX trajectory making an additional investigation of their local accuracy unnecessary.

4.1.3 Villin Headpiece Simulations

Analogously to the previous simulations, REX temperature distributions were obtained via the modified REX temperature generator according to Eqs. A.5 and A.6. However, due to the increased system size more replicas were necessary to obtain nearly constant exchange rates across all replicas. Using the distribution growth parameter $k = 0.0065$, it is able to generate a distribution ranging from 300 K to 625 K over 100 replicas while maintaining equally-good exchange rates as for Trp-Cage. This time, the backbone RMSD between initial and target structure was much higher at 16.2 Å. Looking at the RMSD time evolution of the regular MD simulation (cf. Fig. 4.5A), we can see that the structure quickly collapses to approximately 8 Å. Only minimal fluctuation around this value indicate that this protein conformation is fairly stable for VHP.

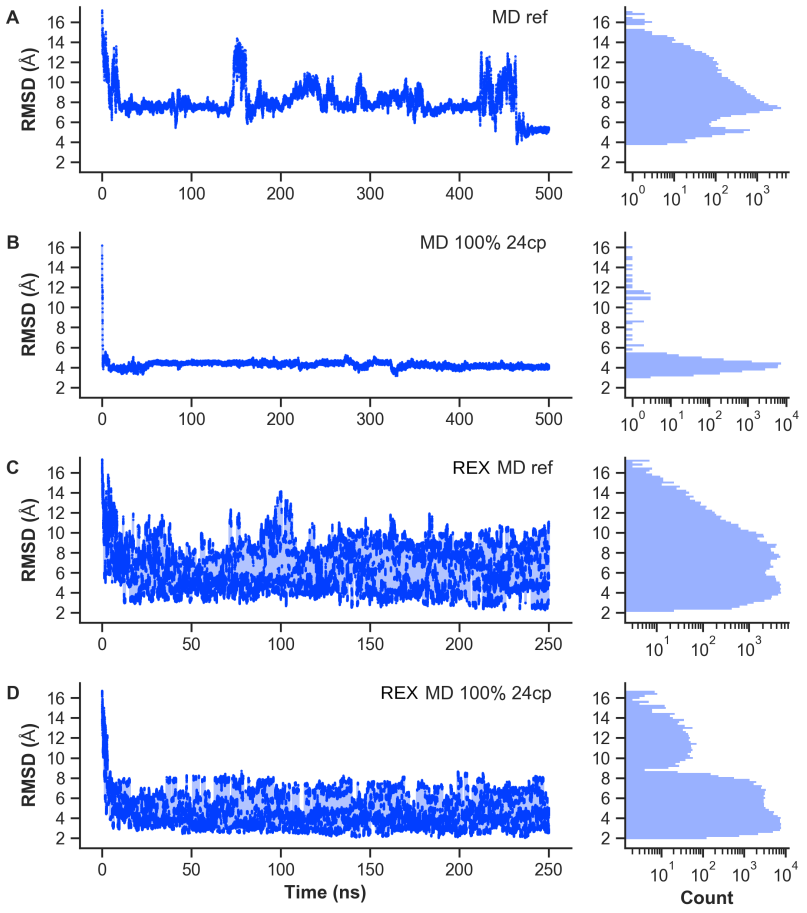

Figure 4.5. Comparison of VHP MD and REX MD simulations. Backbone RMSD time evolution at lowest-temperature replica and corresponding histogram with logarithmic count axis. Values were taken from 500 ns MD and 250 ns REX MD trajectories, respectively. **(A)** MD reference simulation without additional bias. **(B)** MD simulation with 24 native contact restraints. **(C)** REX MD reference simulation without additional bias. **(D)** REX MD simulation with 24 native contact restraints. Adapted from Ref.[1] under CC BY 4.0.

As expected, short conformational transitions do happen before the protein falls back to 8 Å. During the last 50 ns of ths 500 ns long MD simulation, the protein succeeds to adapt a lower energy state at around 4-5 Å. Next, to assess the importance of a contact-driven bias we can see that the MD simulation with 24 native contact pairs immediately guides the protein towards a 4 Å RMSD configuration, as displayed in Fig. 4.6B. One can observe that the protein gets trapped during the simulation and cannot overcome local energetic barriers at a fixed temperature of 300 K through the whole simulation. Here, the best observed RMSD value was 3 Å while the normal MD simulation only reached 4 Å. The comparison of both REX cases shows yet again a significant improvement of the RMSD statistics, as confirmed by the histograms of Fig. 4.5(C-D). Here, the first case corresponds to the REX simulation without bias. The application purpose of REX as an enhanced sampling technique is shown clearly, the conformational search space is broad but reaches conformations up to 2 Å RMSD. During the last scenario, a purely-native bias consisting of 24 contact pairs is effective and guides the protein as intended towards native-like folds. Now the majority of occurring structures are between 2 and 4 Å. This observation shows yet again how powerful REX is when coupled with a contact-driven bias potential. Similar to Trp-Cage, this great performance improvement is justifying the additional computational costs resulting from the parallel running replicas.

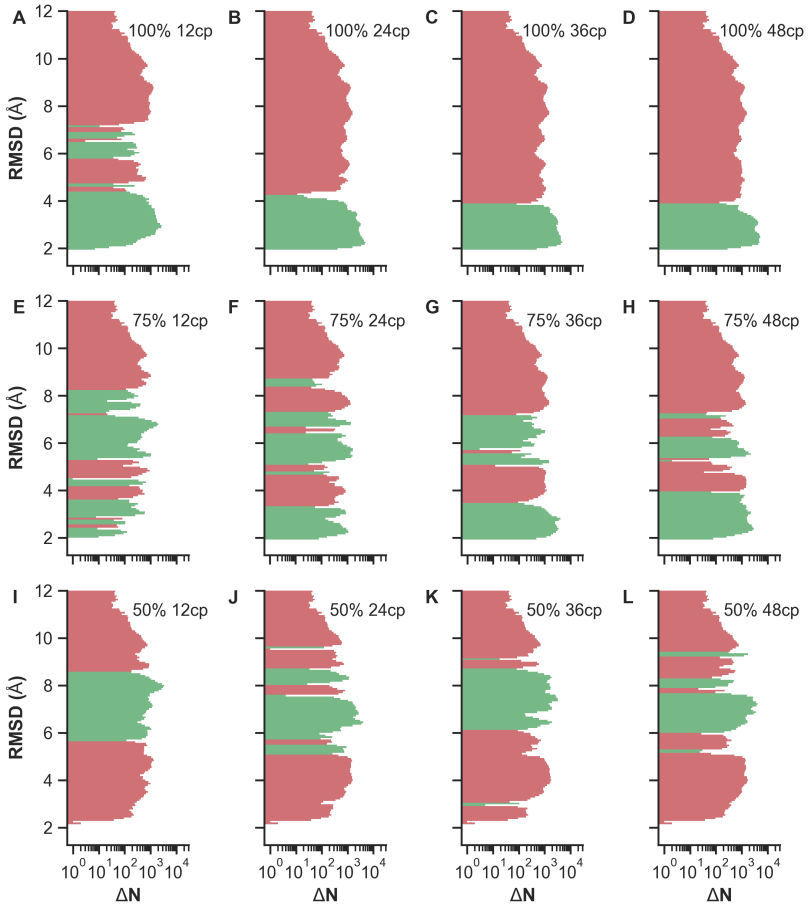

Figure 4.6. ΔN histograms of VHP REX MD simulations. Histograms show the enrichment and depletion of conformations with a particular backbone RMSD at $T_0 = 300\,\mathrm{K}$ as compared to the reference. Histogram bins are defined by the RMSD axis, while the logarithmic ΔN axis illustrates the count difference between the tested REX MD cases and the REX MD reference simulation with $\Delta N = N_{\mathrm{case}} - N_{\mathrm{ref}}$. Positive (negative) values corresponding to enrichment (depletion) are shown in green (red). **(A-D)** Simulations with 100% TPR and 12, 24, 36, 48 contact pairs. **(E-H)** Simulations with 75% TPR and 12, 24, 36, 48 contact pairs. **(I-L)** Simulations with 50% TPR and 12, 24, 36, 48 contact pairs. Reproduced from Ref.[1] under CC BY 4.0.

Following the same discussion path as previously, Fig. 4.6 shows a comparison of all tested VHP REX MD simulations via ΔN histograms. Test cases using perfect, i.e. purely native, contact bias are illustrated in Fig. 4.6(A-D). Here, significant improvements are achieved for all simulations. Conformations between approximately 2-4 Å are primarily strongly enriched. In case of only 12 bias contacts, we can additionally observe an increase of less-desired protein conformations at about 6-7 Å RMSD. Note that scenarios with at least 24 contact pairs show very similar results, meaning that additional restraints yield no benefit to the simulation. This means also, that this is the optimal threshold for VHP REX simulations when guided by a perfect bias at 100% TPR. Scenarios with mixed contacts at 75% TPR, indicate a good performance increase as well but less distinct as compared to its Trp-Cage counterparts (cf. Fig. 4.3(E-H) and Fig. 4.6(E-H)). Since VHP is larger and manifests into a more complicated native fold, the pathway leading into such conformations is not as smooth and erroneous restraints seem to have a relatively strong influence. Nevertheless, all mixed cases were able to enrich native-like conformations at a high rate. Simulations with 12, 24 and 36 contact pairs show qualitative meaningful step-wise improvements of the overall RMSD statistics.

Table 4.3. Total Score (TS) and High Accuracy (HA) percentiles of VHP simulations[1]. Overview of observed global distance test (GDT) percentiles. Statistics were taken from trajectories at $T = 300\,\mathrm{K}$ over 250 ns for REX MD and 500 ns for MD, respectively. Listed are the simulation method, the true-positive rate (TPR) in percent, used number of restraining contact pairs (CP), GDT total score percentiles (P_{TS}), and GDT high accuracy percentiles (P_{HA}). Values equal to or greater than the reference values are shaded in gray. According to Eq. 4.5 significantly greater values are bold.

Method	TPR (%)	# CP	P_{TS}^{80}	P_{TS}^{85}	P_{TS}^{90}	P_{TS}^{95}	P_{TS}^{100}	P_{HA}^{80}	P_{HA}^{85}	P_{HA}^{90}	P_{HA}^{95}	P_{HA}^{100}
REX MD	ref	0	50.00	53.47	57.64	63.89	79.17	27.08	30.56	34.72	40.98	58.34
REX MD	100	6	**66.67**	**68.75**	**71.53**	**75.00**	**87.50**	**43.06**	**45.14**	**47.92**	**51.39**	**68.06**
REX MD	100	12	61.11	63.19	65.97	69.44	**86.11**	37.50	39.58	42.36	45.84	**65.28**
REX MD	100	24	**71.53**	**73.61**	**75.00**	**77.08**	**88.89**	**47.92**	**49.30**	**51.39**	**53.47**	**68.75**
REX MD	100	36	**71.53**	**73.61**	**75.00**	**77.08**	**88.89**	**47.92**	**50.00**	**51.39**	**54.16**	**70.83**
REX MD	100	48	**72.22**	**73.61**	**75.00**	**77.08**	**87.50**	**48.61**	**50.00**	**51.39**	**53.47**	**68.06**
REX MD	75	12	47.92	50.70	54.17	59.02	**87.50**	24.30	27.08	30.56	34.72	**65.97**
REX MD	75	24	49.30	54.86	59.02	70.14	**84.72**	25.00	30.56	34.72	45.83	**64.58**
REX MD	75	36	**68.06**	**71.53**	**74.31**	**77.08**	**88.89**	**43.75**	**47.22**	**50.00**	**53.47**	**70.83**
REX MD	75	48	62.50	65.97	**69.44**	**73.61**	**85.42**	38.89	42.36	45.83	49.30	**63.89**
REX MD	50	12	34.03	38.89	44.44	50.70	**79.17**	13.20	17.36	21.53	27.08	55.56
REX MD	50	24	31.25	34.03	36.80	44.45	73.61	10.42	11.81	14.58	22.22	50.00
REX MD	50	36	28.47	31.94	36.11	40.28	70.83	9.03	11.11	14.58	18.06	49.30
REX MD	50	48	28.47	30.56	34.03	36.81	59.72	9.03	9.72	12.50	15.28	36.11
MD	ref	0	25.70	27.08	28.47	35.42	50.00	9.03	9.72	11.11	13.19	26.39
MD	100	24	41.66	42.36	43.06	44.44	57.64	17.36	18.06	18.75	20.14	32.64

While the first case primarily enriches folds between approximately 5 and 8 Å, instead the highest enrichment is observed for conformations between 2 and 3.5 Å when using 36 mixed restraints. By raising the number of used restraints to 48, the enrichment of low-RMSD conformations is further improved. This relative improvement however is negligible, meaning that the optimal number of considered bias contacts with 75% TPR is around 36 for VHP. Lastly, REX simulations using low-quality bias at only 50% TPR are summarized in Fig. 4.6(I-L). Similar to the TPR-Cage simulations, the high ratio of non-native contacts influences the conformation-guiding in a negative way. Here, both very low and very high RMSD conformations are depleted while structure ensembles between approximately 6 and 8.5 Å are enriched. This validates yet again that not all contacts are equally important and a 1:1 ratio of correct and incorrect bias contacts is disadvantageous for general use.

An overview of the GDT-based analysis including TS and HA percentiles above 80% is given by Table 4.3. While previously Trp-Cage managed to reach scores of $GDT_{\mathrm{TS}} = 100$ and $GDT_{\mathrm{HA}} = 98.75$ during the reference REX simulation, VHP only gets up to $GDT_{\mathrm{TS}} = 79.17$ and $GDT_{\mathrm{HA}} = 58.34$. This is already a drastic difference between the baseline performance of REX MD considering that both proteins are quite small and VHPs structure basically contains only one additional α-helix. In case of VHP, the 80th percentile starts with 50 for TS and 27.08 for HA. The indication of significantly improved values via shading and using bold font (see Eqs. 4.4 and 4.5) clearly shows that REX MD benefits from a restraints with 75% TPR or more. In such cases, the biased variant immensely outperforms the normal REX simulation. In compliance with the previous RMSD-based evaluation, application of highly erroneous restraints decreases the refinement probability which is undesired. Considering only realistic scenarios, i.e. simulations with 75% TPR, then the best performance was achieved for 36 contact pairs which is in the order of VHPs sequence length.

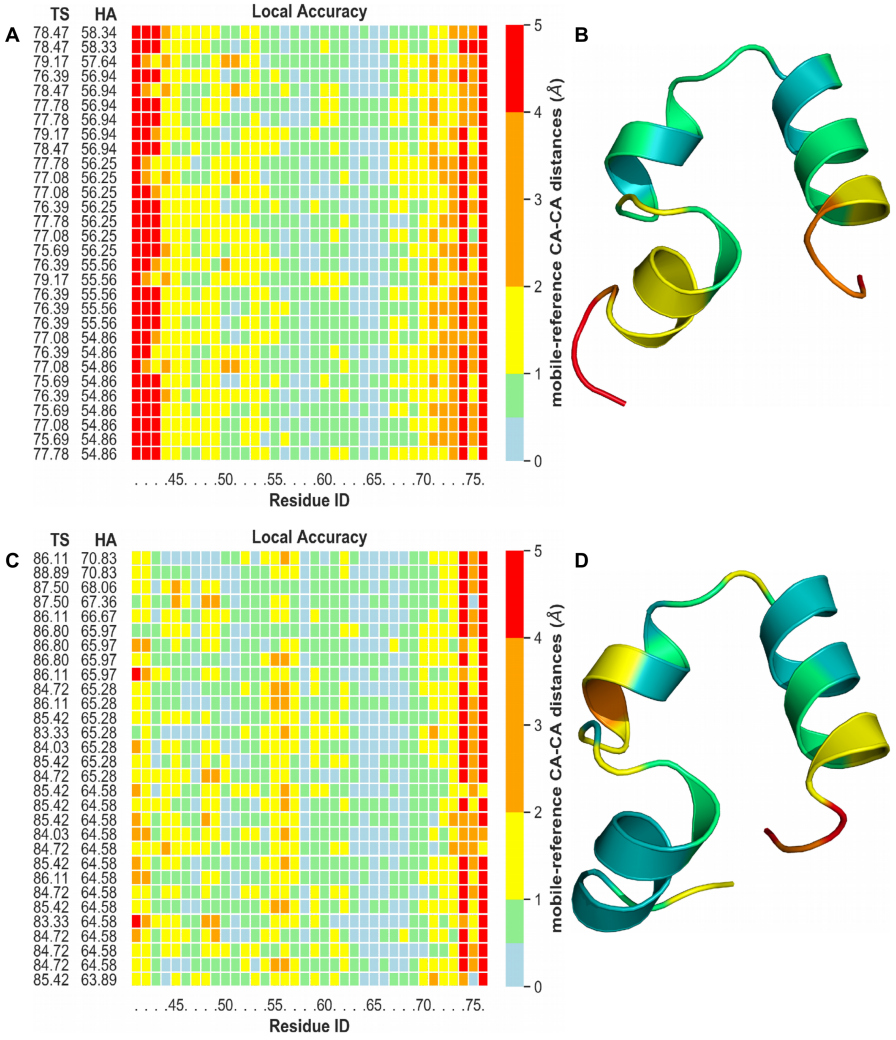

Figure 4.7. Local accuracy of VHP REX MD simulations (Part 1). The 30 best structures according to high accuracy (HA) score. Each matrix row represents one model and is color-coded based on pairwise C_α distances between the model and native fold. This visualization allows to clearly express the refinement-level of each individual protein section. Corresponding global distance test values, i.e. total score (TS) and high accuracy (HA), are shown on the left. **(A)** Reference simulation without additional bias. **(C)** Simulation with 36 contact pairs at 100% TPR. **(B+D)** Tertiary structure of highest-ranking model visualized in the same colors as the local accuracy matrix to the left. Reproduced from Ref.[1] under CC BY 4.0.

Finally, the investigation regarding the local accuracy of the best-refined VHP structures is depicted in Figs. 4.7 and 4.8. Here, each high-scoring model is fitted against the native reference structure and pairwise C_α distances are measured. By depicting the model as a color-coded row based on the distances, it is possible to quickly assess how well refined each individual protein section is. Each local accuracy figure consists of the 30 highest-scoring protein structures ranked by HA. As shown in Fig. 4.7A, we can see that the reference REX simulation can generate VHP structures with HA scores up to 58.34. These models are extremely well refined for the residues 55 to 65. Getting close to either terminus of the protein makes the displacement deviate more. According to our expectations the local accuracy is very poor for highly flexible regions, such as the protein tails. Nevertheless, this result should not be seen negative in any way because a protein's function is usually not related to its structural end sections. The integration of 36 native restraints into REX generates the 30 best models according to Fig. 4.7C. The highest observed HA value is now 70.83 and the majority of the protein is exceptionally well refined.

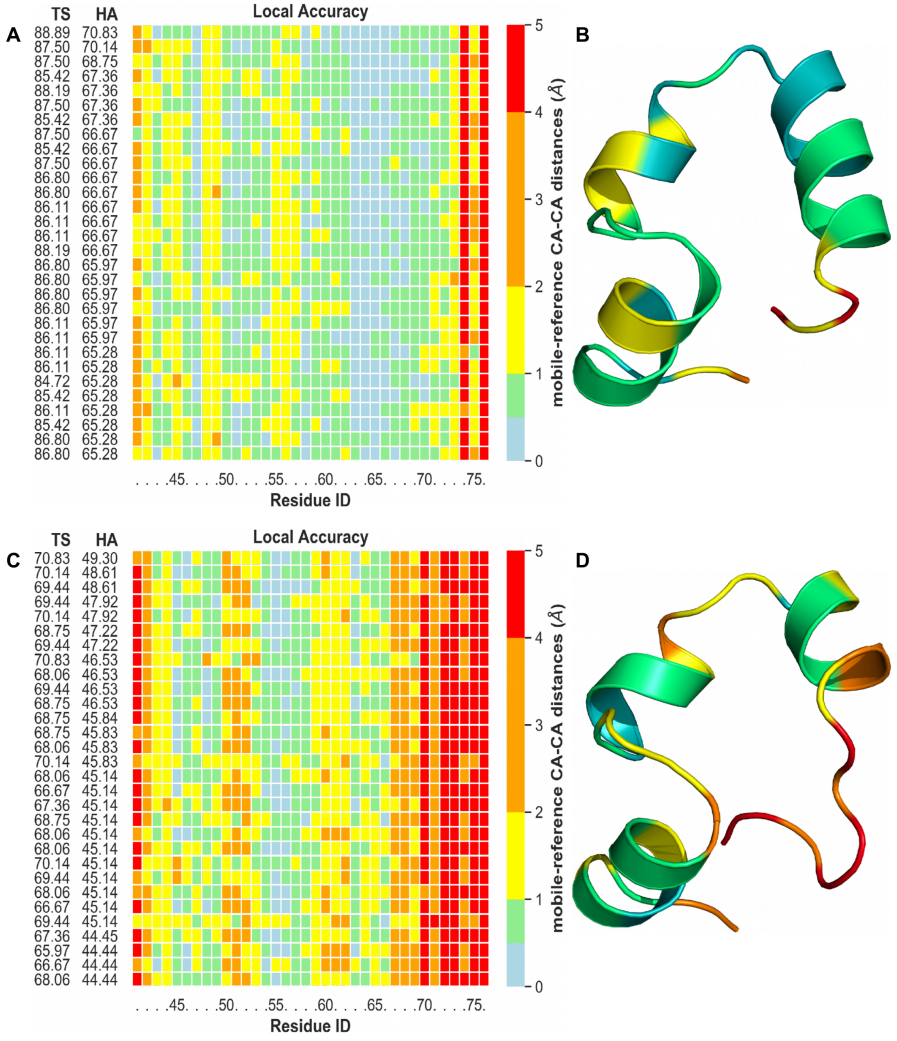

Figure 4.8. Local accuracy of VHP REX MD simulations (Part 2). The 30 best structures according to high accuracy (HA) score. Each matrix row represents one model and is color-coded based on pairwise C_α distances between the model and native fold. This visualization allows to clearly express the refinement-level of each individual protein section. Corresponding global distance test values, i.e. total score (TS) and high accuracy (HA), are shown on the left. **(A)** Simulation with 36 contact pairs at 50% TPR. **(C)** Simulation with 36 contact pairs at 50% TPR. **(B+D)** Tertiary structure of highest-ranking model visualized in the same colors as the local accuracy matrix. Reproduced from Ref.[1] under CC BY 4.0.

This time only one of the two protein ends still indicates a local misalignment above 4 Å, while the other gets improved to values between 1-2 Å. A lowered bias-quality of 36 restraints at 75% TPR does not affect the general quality of the highest-scoring structures. The reviewed structures, as shown in Fig. 4.8A, are as good as the structures with a 100% TPR bias. Lastly, Fig. 4.8C depicts the negative examples of the comparison as the C-terminal end section gets distorted. In this case, non-native long-range bias contacts are clustered within the contact map. Even though the sigmoid potential is designed in such a way that it should minimize guiding-effects of incorrect bias signals, the occurring pair distances are within the range where the energetic penalty results in a sufficiently large force. This combined with the contact clustering makes the attractive force so strong that the corresponding α-helix gets partially unwound.

4.1.4 Summary

To recap, contact information derived by theoretical or experimental methods is valuable for *in-silico* structure prediction and refinement protocols. On its own, it does not contain enough information to fully determine a protein's 3D structure. However, when combined with other methods such as replica exchange, it can drastically boost the overall performance under the right conditions. In this case, contact information was integrated as a bias via a sigmoidal potential. The additional restraints are applied only on C_α atoms of considered contacts and the resulting attractive force guides the protein towards native-like structures. REX as an enhanced sampling technique can generate large ensembles similar to a random walk in conformation space. The generated structures contain lots of valuable information and can be used for all sorts of physically-meaningful analyses.

The main motivation of my applied method is generate native-like folds within a single run. The integrated bias guides the folding process during REX cycles which reduces the sampling space, speeds up the whole process and lowers computational costs. As a proof of concept I compared the performance of four generalized cases, i.e. normal MD, biased MD, normal REX MD and biased REX MD. Contact-guided REX MD drastically outperforms all other cases. Not only did the method generate near-native structures in a shorter period of time, it also allowed the protein to occupy such states for the majority of the trajectory. This significantly increases the chance of observing and selecting such structures. Additionally, I compared different scenarios of varying bias-quality using up to 48 contacts at 100%, 75% and 50% true-positive rate. By additionally running one reference simulation, i.e. REX without bias, I was able to compare the individual performance changes with respect to the reference case. During the performance analyses I applied two different metrics, namely RMSD and GDT, to measure structural similarity with the already known protein structures. To summarize, all REX scenarios with 100% TPR and 75% TPR outperformed the reference case by a large margin. For both tested proteins, these simulations generated much more native-like structures during the same period of time. In case of 50% TPR however, the comparisons show a significant performance loss which makes such low-bias quality unsuitable for the applied use-case. This also proves that incorrect mapping of coevolutionary contacts impacts the performance of contact-guided REX MD extremely negative. Furthermore, bias contacts which are close located on the contact map can be considered as a cluster. Due to their spatial proximity the energetic penalty does add up, increasing the attractive force between the contact pairs and potentially leading to undesired protein trapping. By carefully comparing both RMSD and GDT statistics of the realistic scenarios, i.e. cases with TPRs unequal to 100%, the best performance gain was achieved for Trp-Cage with 12 contact pairs at 75% TPR, and for VHP with 24 contact pairs at 75% TPR, respectively. With a sequence length of $L = 20$ for Trp-Cage and $L = 35$ for VHP, I conclude that the best performance can be achieved for $N = \frac{L}{2}$ to L contacts. Also note that in this study all REX MD simulations outperformed the MD simulations regardless of the total number of used contacts or their TPR (cf. Tables 4.2 and 4.3).

Overall, this method can be directly applied, does not require fine-tuning of numerous parameters and yields high-quality results as long as the bias has a TPR $\geq 75\%$. Even with false contact information, this method greatly enriches native-like conformations and can be used as a refinement tool while generating large structural ensembles. Additionally, it is easy to combine different sources of contact information into REX, enabling it as a hybrid tool for joint data interpretation.

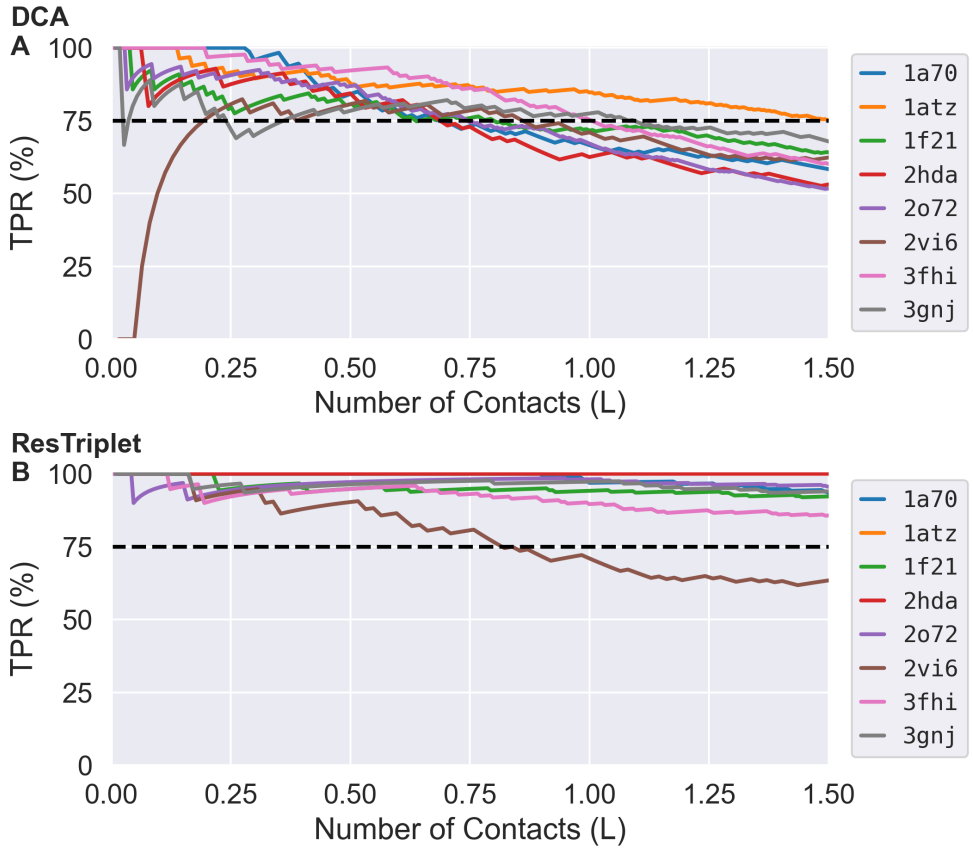

Figure 4.9. True-Positive Rate Analysis of two contact-deriving methods. Number of contacts are scaled to protein length L, which facilitates the comparison. PDB ids of selected proteins are shown in the legend box. Additionally, a 75% TPR threshold is visualized by a horizontal dashed line. Contact information was derived from **(A)** Direct coupling analysis[49–52] or **(B)** ResTriplet[97,98].

4.1.5 Learned Lessons: Bias Guidelines

As a general rule of thumb, the following guidelines should be considered to achieve optimal results with contact-guided REX MD:

1) Maximize the true-positive rate.
Use only contact information obtained by a reliable source or method. The true-positive rate is the biggest impact factor. For methods such as DCA, the quality of predicted contact pairs is highly correlated with the quality of the multiple sequence alignment, i.e. number of effective sequences[50] or used alignment algorithm[87,88].

2) For a safe approach apply $\mathbf{N \approx \frac{3}{4}L}$ contacts.
If the contact-deriving method is very accurate then more contacts can be used. Fig. 4.9 exemplarily compares the contact prediction of DCA (statistical method) vs. ResTriplet (ML method).

3) Illustrate considered bias contacts in a contact map.
Adjust the coupling strength of clustered bias contacts to prevent conformation trapping/enforcing.

4) Long-range contacts have a stronger influence than short-range contacts.
Sometimes the highest ranked bias contacts are primarily short-ranged. In such cases it may be advantageous to split contact information into two sets: short-ranged and long-ranged. Considered bias contacts can then be integrated with a specific ratio of short-ranged to long-ranged contacts.

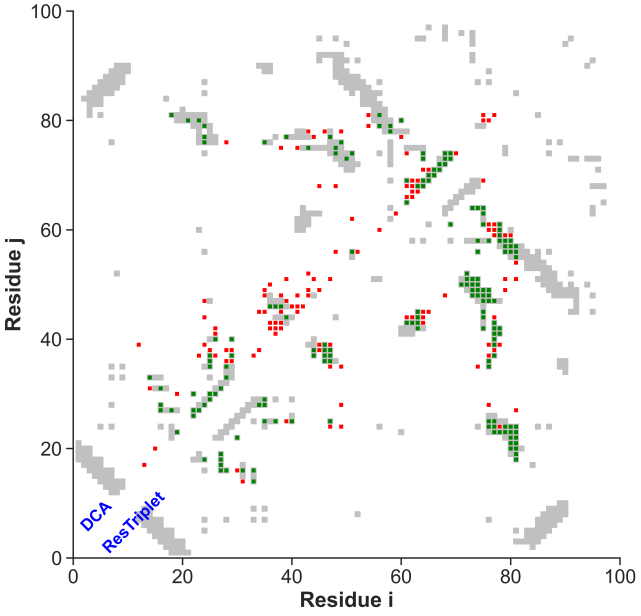

Figure 4.10. Comparison of contact predictions by DCA and ResTriplet (pdbid: 1a70, N = 1.5 L). Depicted are native contacts (grey), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

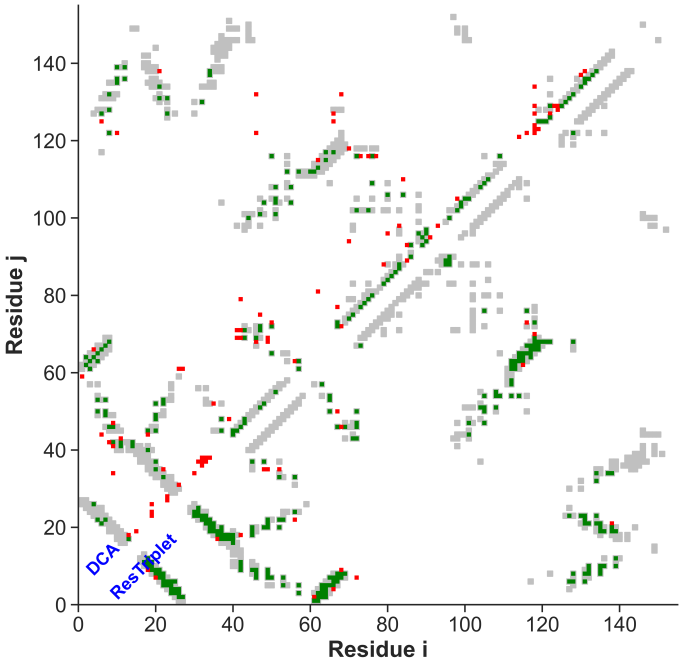

Figure 4.11. Comparison of contact predictions by DCA and ResTriplet (pdbid: 1f21, N = 1.5 L). Depicted are native contacts (grey), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

Lastly, I want to point out that contact prediction methods also have certain tendencies in their prediction pattern. To illustrate this I created contact maps containing both DCA and ResTriplet predictions for eight different proteins. Resulting comparisons are visualized in Figs. 4.10 and 4.11 for a total of 1.5 L contact predictions, with L being the protein's sequence length. As shown in these contact maps, DCA predictions tend to have a relatively high ratio of short-range contacts which are close to the main diagonal. Predicted contacts also seem to be spread out relatively even among the contact map. ResTriplet on the other hand seems to prioritize long-range contacts and manages to capture them correctly with a high accuracy. However, these predictions are also typically clustered. Additional comparison cases can be found in appendix Figs. B.9 to B.22. The application of either of these methods, i.e. DCA or ResTriplet, would require a different handling according to the presented bias guidelines. For example, the proposed *safe approach* is better suited for DCA-derived bias contacts, primarily because the true-positive rate of DCA is typically lower than for ResTriplet (cf. Fig. 4.9). Furthermore, DCA contacts can often be implemented with an equal coupling strength λ, since they are evenly spaced out. The handling of ResTriplet contacts would look quite different: Here it is safe to apply many more contacts to bias the simulation but it is absolutely necessary to reduce the coupling strength due to contact clustering. Based on these two application examples, it is best to first study the applied prediction method and its tendencies before following the presented guidelines for contact-guided REX MD.

4.2 Bias-Potential Optimization

This section covers a study to determine optimized parameters for the applied sigmoid bias potential, resulting in a distance-dependant attractive force between biased contact pairs. Section 4.2.1 introduces the five tested bias potentials and the used target structures, consisting of one α-helical and one β-sheet protein. Next I assess the potential's conformation-guiding effects of each scenario in section 4.2.2. More precisely, I analyze the resulting GDT distributions based on the parameter choices of λ (coupling strength) and r_0 (maximum force distance). Lastly, I summarize my findings in section 4.2.3 and infer optimal bias parameters that yield the best protein structures in contact-guided REX MD.

4.2.1 Study Concept

The shape of the underlying bias potential plays a major role in the performance of contact-guided REX MD. In order to find an optimal parameter range I investigated multiple potentials by varying either the coupling strength λ or the maximum force distance r_0. According to Eqs. 3.33 and 3.34, the sigmoid shape can also be adjusted by the α parameter. It affects the S-shape of the potential, i.e. how fast it transitions from low values to high values. High α values correspond to shapes similar to a step function, whereas low α values transform it into a linear ramp. However, according to my intention I only applied $\alpha = 25\,\text{Å}^{-1}$, as it resembles a smooth bias activation. Fig. 4.12A gives an overview of all tested potentials and Fig. 4.12B of their resulting forces. Note that λ defines the upper limit of the sigmoid function and thus the overall bias strength. r_0 on the other hand defines the position of the potential's inflection point, where the resulting force reaches its maximum strength. The choice of r_0 also affects the effective range, where contact restraints can experience a pulling force.

These potentials were tested on two mid-sized proteins, which are depicted in Fig. 4.13. The first test protein is Nanog homeodomain (PDB id: 2vi6[191]). It consists of three α-helices and has a length of 62 residues. The second test protein, Yes SH3 domain (PDB id: 2hda[192]), has a β-sheet structure and a length of 64 residues. Starting structures were obtained via *de novo* folding using `pyRosetta` (cf. chapter 5). To bias the simulations, I applied exactly 40 bias restraints at 80% TPR, which were derived with direct coupling analysis[49–51]. Integrated bias contacts are visualized in appendix Figs. C.1 and C.2. Each REX simulation yielded a trajectory of 500 ns.

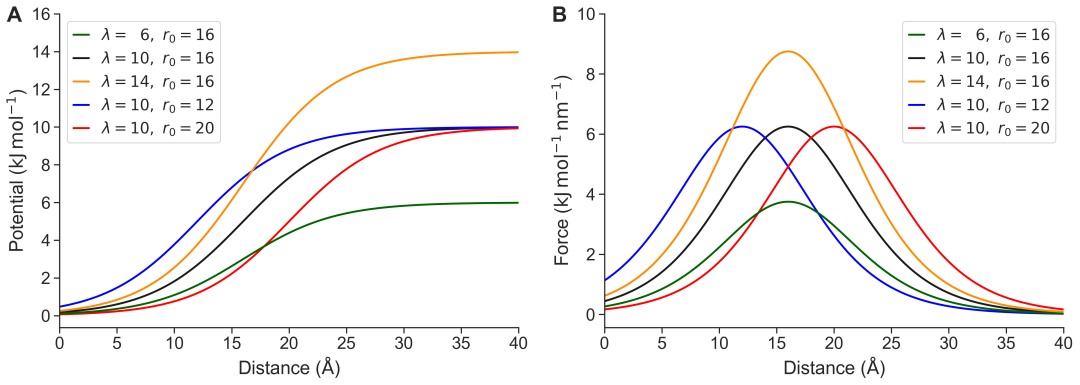

Figure 4.12. Shape of investigated bias potentials and their resulting force. Sigmoid bias potential for different coupling strengths λ (kJ mol^{-1}) and maximum force distances r_0 (Å), as defined by Eqs. 3.33 and 3.34. **(A)** Bias potential V(r). **(B)** Bias force dV(r)/dr.

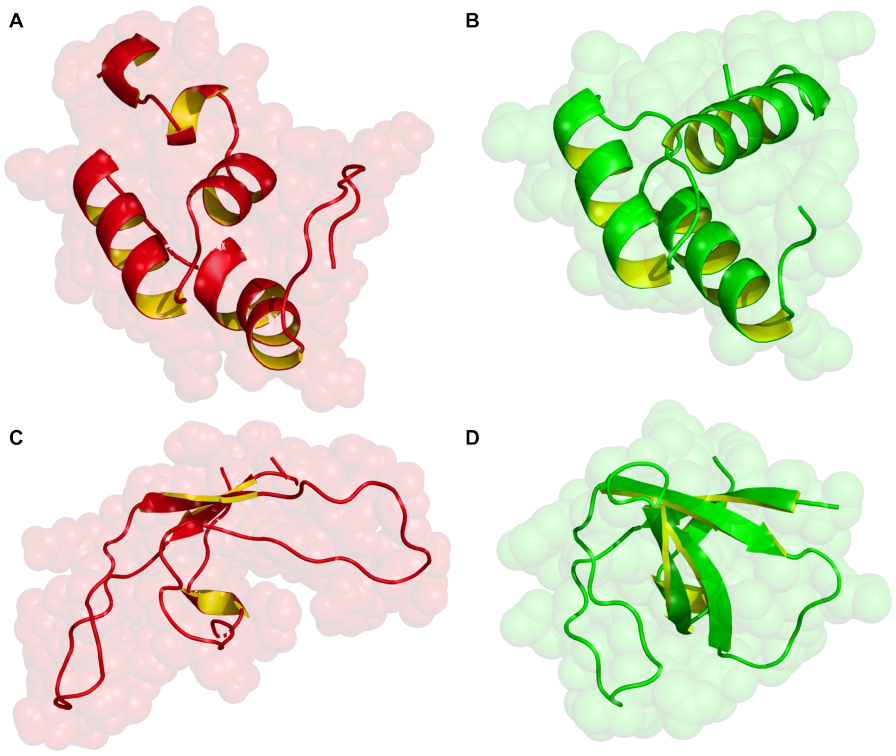

Figure 4.13. Protein structures used for bias-potential optimization. Initial (A) and native (B) conformation of Nanog homeodomain (PDB id: 2vi6[191]) with a difference of 7.5 Å. Analogously, initial (C) and native (D) conformation of Yes SH3 domain (PDB id: 2hda[192]) with a backbone RMSD difference of 6.7 Å. Initial conformations were folded *de novo* (cf. chapter 5) with `PyRosetta`[193]. Visualized in `pyMOL`[189,190].

4.2.2 GDT Distribution Analyses

In order to determine the best parameter choices, I decided to investigate the structure quality of each REX trajectory measured by GDT. My evaluation is straight forward and simply compares the GDT distributions that were obtained during the 500 ns long REX simulations. I primarily looked at the highest scores by calculating GDT percentiles between 80% and 100% in steps of 5%. The comparison of these values directly shows which distribution is more favorable, allowing me to infer optimized potential parameters for a general use-case.

Fig. 4.14 compares the GDT TS distributions based on different λ parameters. Similar to this, GDT HA distributions are shown in appendix Fig. C.3. As expected, it is very important to find an adequate balance of bias strength and number of used bias contacts in order to obtain a high ratio of well-refined structures. As exemplarily shown for Nanog homeodomain in Fig. 4.14A, a *weak coupling strength* of $\lambda = 6\,\mathrm{kJ\,mol^{-1}}$ translates into few high-scoring GDT structures. Here, the 80th GDT percentile corresponds to 61.8, the 90th percentile to 76.3 and the best achieved GDT score was 92.7. While the obtainable structure quality is expected to increase in longer simulations, this is not optimal for the intended use-case. Instead, increasing the coupling strength up to $\lambda = 10\,\mathrm{kJ\,mol^{-1}}$ (corresponds to a weak hydrogen bond) greatly improves the performance. The corresponding GDT distribution yields many more structures of high-quality which lowers the required computing costs of REX. In this case, the 80th percentile raises to 80.0, the 90th percentile to 85.0 and the 100th percentile to 95.0. When further increasing the coupling strength up to $\lambda = 14\,\mathrm{kJ\,mol^{-1}}$ (cf. Fig. 4.14B), we can observe a slight drop in performance. The 80th, 90th and 100th percentiles are now 77.3, 82.7 and 96.4, respectively.

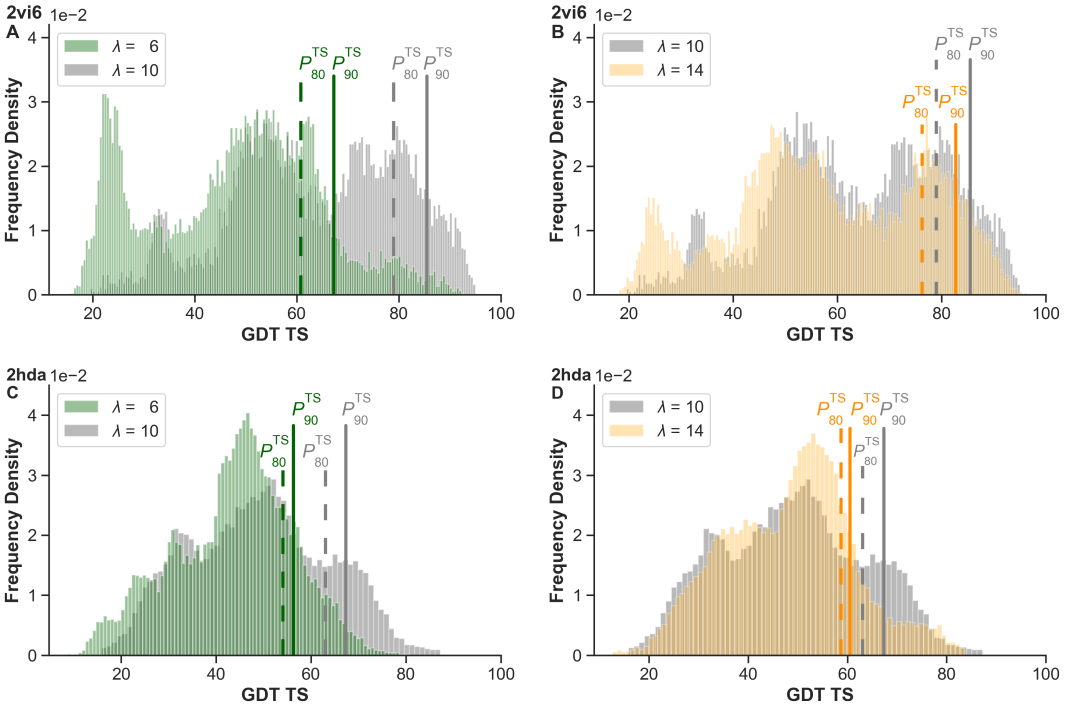

Figure 4.14. GDT TS distributions based on λ parameters. Vertical lines represent the 80th and 90th percentile. **(A+B)** Nanog homeodomain (PDB id: 2vi6[191]). **(C+D)** Yes SH3 domain (PDB id: 2hda[192]).

Note that the highest observed value got slightly improved but from a statistical point of view the underlying GDT distribution got worse. This indicates that the effective bias strength reached a limit where certain conformations get enforced. In other words, contact-guided REX is starting to impose conformations and not just guide towards them. Since Nanog homeodomain is an all α-helical structure, integrated bias contacts are typically spread out on the contact map, as shown in Fig. C.1. Therefore, the observed protein entrapment is not as strong.

Similar tendencies are observed for the second protein, as depicted by Figs. 4.14(C+D). The REX simulation with $\lambda = 6\,\mathrm{kJ\,mol^{-1}}$ yields an 80th percentile of 54.1 up to an 100th percentile of 80.9. In this case, the best results were also achieved with $\lambda = 10\,\mathrm{kJ\,mol^{-1}}$, which reached a GDT TS of 87.2. I want to emphasize that the distribution change from $\lambda = 10\,\mathrm{kJ\,mol^{-1}}$ to $\lambda = 14\,\mathrm{kJ\,mol^{-1}}$ indicates a stronger protein entrapment resulting from the bias potential as compared to the other protein. Since Yes SH3 domain is a β-sheet protein, integrated bias contacts are typically close to each other on the contact map (cf. Fig.C.2). These can be viewed as a cluster, meaning that the bias potential adds up which results in a stronger attractive force which in turn enforces protein conformations.

The r_0 parameter variation and its effect on GDT distributions are shown in Figs. 4.15 and C.4 for the TS and HA variant, respectively. Overall, the parameters $r_0 = 12\,\text{Å}$ and $r_0 = 16\,\text{Å}$ yield very similar structures during REX. The observed GDT percentiles show a minor improvement with $r_0 = 16\,\text{Å}$ for Nanog homeodomain (Fig. 4.15A) and a distinct improvement for Yes SH3 domain (Fig. 4.15C).

Increasing r_0 up to 20 Å significantly impacts the GDT distribution. As shown by Figs. 4.15(B+D), the obtained structure quality drops for both proteins, with a bigger change observed for Yes SH3 domain.

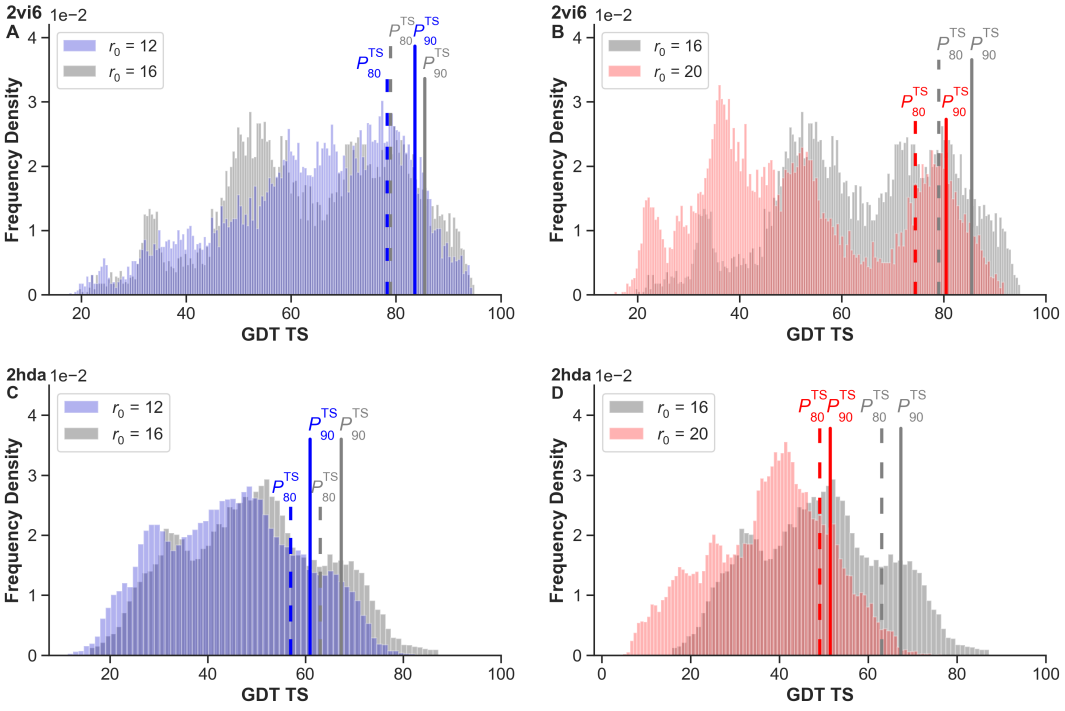

Figure 4.15. GDT TS distributions for different r_0 parameters. Vertical lines represent the 80th and 90th percentile. **(A+B)** Nanog homeodomain (PDB id: 2vi6[191]). **(C+D)** Yes SH3 domain (PDB id: 2hda[192]).

The larger r_0 parameter increases the range at which integrated bias contacts are pulled together as well as the distance of maximum force. However, this mainly influences the interaction with false-positive bias contacts. As previously shown in section 4.1, such contacts have a very negative impact on contact-guided REX, which is reflected in the observed distribution shifts.

4.2.3 Summary

In this section I performed various contact-guided REX simulations using different bias potentials. Meaningful GDT percentiles that were measured for each scenario are summarized by Table 4.4. Shaded cells indicate the highest values with respect to the variation type and protein. Based on all tested scenarios, the best results were achieved with the bias parameters $\lambda = 10\,\mathrm{kJ\,mol^{-1}}$ and $r_0 = 16\,\text{Å}$. The table also shows that it is important to perform a contact map analysis prior to the REX simulation. Depending on the location of considered bias contacts, the coupling strength λ should be adjusted to compensate clustered bias contacts. This was especially observed for the β-sheet protein, i.e. Yes SH3 domain. Furthermore, individual bias strengths should be in the order of a weak hydrogen bond, which corresponds to approximately $\lambda = 10\,\mathrm{kJ\,mol^{-1}}$ for a single bias contact. Large r_0 parameters should be avoided, as they primarily influence the attractive force between false-positive bias contacts due to the increased range. r_0 parameters below $r_0 = 16\,\text{Å}$ yield very similar results but slightly favor $r_0 = 16\,\text{Å}$, as shown for both proteins. The statistical comparison of GDT percentiles also indicates that α-helical structures yield better results as compared to β-sheet structures. Nevertheless, both systems could achieve highly native-like conformations with GDT TS values above 80 in all cases, except for Yes SH3 domain during the bias application with $r_0 = 20\,\text{Å}$.

Table 4.4. GDT percentiles during study to optimize the sigmoidal bias potential. Upper-half values correspond to λ-variation and lower-half values to r_0-variation. Shaded cells correspond to the highest values with respect to the variation type and protein. Listed are the PDB id, coupling strength λ, s-turn distance r_0, and selected GDT total score (TS) and high accuracy (HA) percentiles.

PDB id	λ (kJ mol^{-1})	r_0 (Å)	P_{TS}^{80}	P_{TS}^{85}	P_{TS}^{90}	P_{TS}^{95}	P_{TS}^{100}	P_{HA}^{80}	P_{HA}^{85}	P_{HA}^{90}	P_{HA}^{95}	P_{HA}^{100}
2vi6	6	16	61.82	64.09	67.28	76.36	92.73	38.18	40.46	44.54	53.64	80.00
2vi6	10	16	80.00	81.82	85.00	88.64	95.00	57.73	60.46	64.09	69.09	84.09
2vi6	14	16	77.28	80.00	82.73	86.36	96.36	54.55	57.72	61.36	66.36	86.82
2hda	6	16	54.09	56.31	58.55	63.01	80.86	33.81	36.29	38.77	42.90	65.24
2hda	10	16	63.07	66.36	69.66	72.96	87.24	41.81	44.19	46.57	50.14	71.60
2hda	14	16	58.69	60.53	64.22	71.49	83.56	39.93	42.55	45.19	50.46	68.02
2vi6	10	12	79.54	81.36	83.64	86.36	94.54	56.82	59.54	62.72	66.82	82.73
2vi6	10	16	80.00	81.82	85.00	88.64	95.00	57.73	60.46	64.09	69.09	84.09
2vi6	10	20	75.46	78.18	80.46	83.64	91.82	52.73	55.91	59.09	62.27	78.18
2hda	10	12	58.00	60.95	65.20	68.84	84.62	38.21	40.44	43.77	47.10	64.90
2hda	10	16	63.07	66.36	69.66	72.96	87.24	41.81	44.19	46.57	50.14	71.60
2hda	10	20	49.07	51.46	54.40	58.61	76.53	33.52	35.83	38.86	42.78	60.51

5

Starting-Structure Generation

This chapter covers how to generate and optimize the starting conformations of each individual replica for contact-guided REX MD. In section 5.1 I briefly discuss why replicas should start with different starting conformations and how it provides additional pathways towards the native state. Furthermore, I show a method to quickly generate a broad spectrum of unique starting structures ("decoys[†]"). In section 5.2 I investigate correlations between applied energy mappings and obtained refinement levels of generated decoys. I compare the overall performance of the applied de novo folding algorithm and test how reliable the energy mapping is based on a set of seven different protein models. In section 5.3 I explain what should be considered during the final decoy selection and present two different approaches. Additionally, I perform some detailed analyses on the resulting decoy selections and compare their quality with regard to the intended use-case. Lastly, in section 5.4 I summarize my findings and draw a conclusion on the entire topic of starting-structure generation.

5.1 De Novo Folding

The systematic REX study in section 4.1 used the same starting conformation for all replicas. I obtained these conformations by heating up the proteins in normal MD simulations with explicit water at 500 K. The final selection was made based on high RMSD values with respect to the native fold while prioritizing minimal remains of secondary structure motifs. This choice was done intentionally, as I wanted to assess the influence of used bias restraints and find an estimated bias-quality threshold for optimal REX performance. However, it is reasonable that the performance of REX is dependant on the starting conformations and their similarity with respect to the native state. Generally speaking, typical applications of contact-guided REX MD should strongly benefit from varying starting conformations.

[†]decoy: lowest-energy structure of a MC trajectory[194]

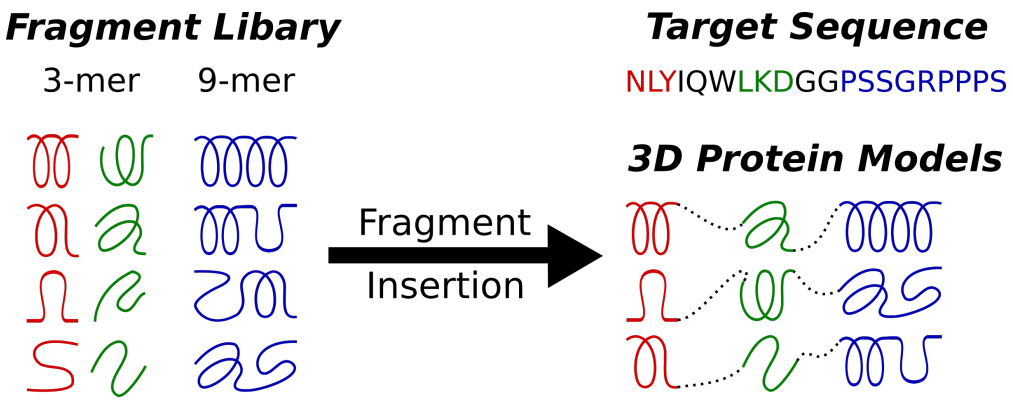

Figure 5.1. Concept of fragment insertion during *de novo* folding algorithm. The fragment library consists of fragments, which correspond to tertiary structure segments of experimentally determined structure models. Therefore, each fragment library depends on multiple sequence alignments and knowledge of sequentially similar structures. During fragment insertion, according sections (red, green and blue) of the newly generated protein model get transformed into the shape of library fragments.

This diversification alleviates starting conditions and increases the accessible sampling space at the beginning, which opens up additional pathways to the native fold. Otherwise it would require hundreds of nanoseconds of trajectory time and multiple turnaround cycles to get access to the same broad spectrum of protein structures. Additionally, starting-conformation bias gets reduced and false-positive contacts do not affect each replica in the same way. This increases the chance that some replicas may adapt conformations by partially negating erroneous bias restraints. In the end, best results will be achieved for simulations which suppose a good balance of diversified starting conditions and a sufficiently strong bias with a high true-positive rate.

One of the best methods to generate unique starting structures (*decoys*) is *de novo* folding, i.e. by starting from sequence. MD simulations with linear protein models however are not reasonable, since these would require a huge water box and the majority of calculations would be wasted on water interactions. This would make such simulations computationally extremely demanding. Instead, it is better to perform Monte-Carlo (MC) simulations. MD simulations rely on physical force fields and generate meaningful trajectories with femtosecond time steps, which can be interpreted as a slow-motion movie capturing the atomic movement. MC relies on random moves and allows very large conformational changes to be condensed into one single move. To some extent, MC trajectories can still be physically meaningful but this mainly depends on the used protocol and if movement restrictions are applied. MC algorithms are mainly driven by the principle of energy minimization of biomolecular structures. Each iteration compares assigned energy values of previous and new configuration and accepts the changes if they are favored, according to the Metropolis criterion [195]. Nevertheless, many algorithms still apply movement restrictions, e.g. by limiting dihedral angle changes to a specified threshold. One of the most famous MC-based software is `Rosetta` [196,197], which offers many algorithms for computational modeling and analyses of proteins or RNAs.

During my studies I utilized `PyRosetta` [193,194], which allows an interactive application of `Rosetta` via Python-wrapper functions. To quickly generate unique starting structures of desired protein targets, I designed a MC *de novo* folding algorithm with additional fragment insertion, which constructs a new protein model within approximately 10 to 20 seconds (based on protein length) using a single CPU core[†]. Fragment insertion [198] is the process where entire pieces of experimentally determined structures, i.e. the 3D atom locations and their bond orientations, are inserted into the model. Such fragments can be obtained from different tools that construct a fragment library for the specific target protein.

[†]Intel Core i7-8700 (6 cores, 12 threads, 3.2 GHz)

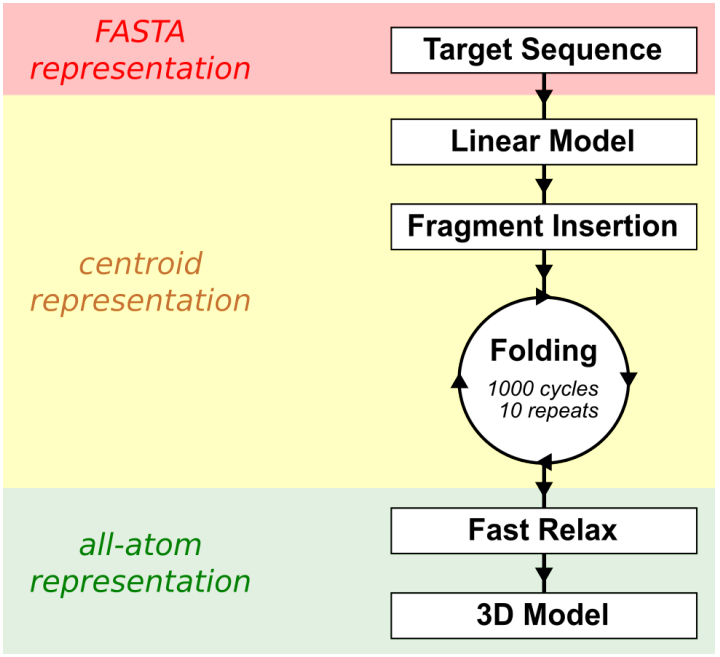

Figure 5.2. Overview of the applied *de novo* folding algorithm during decoy creation with PyRosetta[193,194]. The algorithm can be classified by three stages based on the representation type: **I)** FASTA representation (red), **II)** centroid representation (yellow), **III)** all-atom representation (green). Licensed by Arthur Voronin under CC BY-SA 4.0.

These tools heavily rely on multiple sequence alignments of the protein family and additionally consider already resolved structures. In my case, I obtained fragments with the lengths $L = 3$ (*3-mer*) or $L = 9$ (*9-mer*) using the `Robetta Fragment Server`[25,199] for my designed MC algorithm. Fig. 5.1 illustrates the concept of fragment insertion and Fig. 5.2 summarizes the workflow of the *de novo* folding algorithm. Starting with a straight protein model based on the `FASTA`[200,201] sequence, the MC algorithm first constructs the model with a lower resolution using the so-called *centroid* representation. In the following steps, fragment insertions and roughly 10000 folding moves are applied. Afterwards the representation is switched to a higher resolution, i.e. *all-atom*, and a fast relaxation protocol[202] is applied. Finally, the generated structures, which are usually referred to as *decoys*, are saved as .pdb files and additional log files containing the high-resolution Rosetta scores (*REF2015*)[203,204] are created. As an extra feature, it is possible to stream the decoys directly to an opened `PyMOL`[189,190] interface during the execution of the MC algorithm[205], providing a first impression of the generated structure ensemble.

The applied *de novo* folding algorithm code is attached to appendix D. It performs by design very short folding cycles. It is not supposed to generate highly native structures but instead should create as many decoys as possible within a short period of time. The primary goal of its application is to quickly create a sufficiently large ensemble of varying structures in order to select N_{rex} unique decoys to populate each individual replica. By doing so, the contact-guided REX MD protocol should become much more efficient because additional pathways towards the native fold are provided from the very beginning, as illustrated in Fig. 5.3. In my case, I found it sufficient to generate approximately 5000 decoys with $N_{\mathrm{c}} = 1000$ folding cycles and $N_{\mathrm{r}} = 10$ folding repeats.

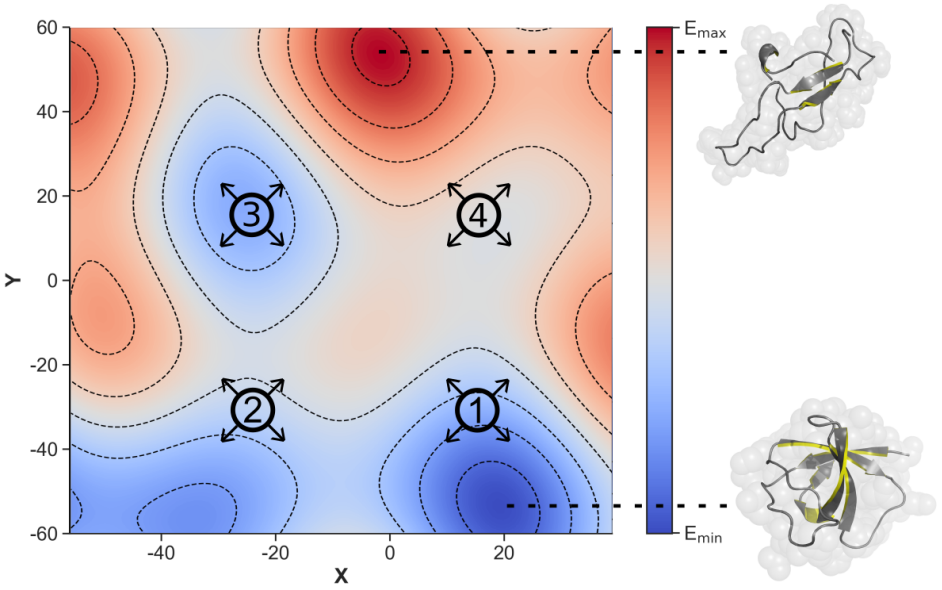

Figure 5.3. Concept of enhanced sampling with unique REX starting conformations. Exemplary visualization of a protein's conformation space. Protein conformations are represented in 2D via dimension reduction of C_α distance matrices onto (X,Y) coordinates. Corresponding energies are illustrated in different colors and contour levels. Unique starting conformations are depicted by numbered circles and their possible sampling direction by black arrows. Additionally, low- and high-energy states are indicated by the native and unfolded conformation, respectively. Licensed by Arthur Voronin under CC BY-SA 4.0.

5.2 Decoy Analyses

In order to assess the performance of the *de novo* folding algorithm, I selected seven different test proteins with varying structure complexity. These proteins were also used in two of my major studies, namely for the bias-potential optimization discussed in section 4.2 and to evaluate the ensemble-selection algorithms discussed later in chapter 6. A summary of the used proteins, including their size and the occurring secondary structure motifs, is provided by Table 5.1. These proteins have sequence lengths between 39 and 92 residues. Four proteins have purely α-helical structures, two consist only of β-sheets and NTL9 is the only test protein with a mixed structure. I decided to order the proteins based on the secondary structure motifs and also the protein size during the upcoming analysis evaluation. This makes it easier to compare the different cases and to recognize structure-related patterns or performance differences.

Utilizing the MC folding algorithm, I created a data set of 5000 decoys for each protein target. Each model was generated with exactly four fragment insertions (3x 3-mer, 1x 9-mer), followed by 10000 folding moves ($N_\mathrm{c} = 1000$ folding cycles, $N_\mathrm{r} = 10$ folding repeats). With these settings it requires approximately 10 to 20 seconds, based on the sequence length, to generate exactly one decoy. A typical modern desktop PC with 12 CPU cores and 2 threads per core requires therefore only between 0.6 and 1.2 hours for 5000 decoys if multithreading is utilized. The elapsed wall-time is relatively short considering that the main intention is to select about 50 to 100 structures out of 5000 generated decoys and to use them as starting conformations for individual replicas.

Table 5.1. Overview of proteins used for decoy creation. Table contains the protein name, PDB id, sequence length and occurring secondary structure (ss) motifs. Ordered by ss motifs and length.

name / description	PDB id	length	ss motifs
BBL	`2wxc`	47	α
Albumin-binding domain	`1prb`	53	α
Nanog homeodomain	`2vi6`	62	α
Lambda repressor	`1lmb`	92	α
NTL9 (N-terminal of L9 protein)	`2hba`	52	α, β
WW domain of human Pin1 Fip mutant	`2f21`	39	β
Yes SH3 domain	`2hda`	64	β

I started by investigating the correlations between the Rosetta scoring functions and the achieved model refinement, measured by the global distance test. More precisely, I wanted to validate how reliable the Rosetta scores are and if this information alone is sufficient to pinpoint high-quality structures. Furthermore, I was interested in learning if the results are somehow related with occurring secondary structure motifs of the models. For example, if the applied Rosetta scoring works better for purely α-helical structures. The correlation of two variables X, Y can be measured by the Pearson correlation coefficient $\rho_{X,Y}$ [206]. It is defined by

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \in [-1, 1] \tag{5.1}$$

where σ_X and σ_Y are the standard deviations of X and Y, respectively, and with the covariance [207]

$$\mathrm{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y}) \tag{5.2}$$

$$= \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)(y_i - y_j). \tag{5.3}$$

This correlation coefficient indicates the linear relation of two variables and ranges between -1 and +1 based on the correlation strength, with the meaning of

$\rho = \ \ 0$: no linear correlation,
$\rho = -1$: `negative` correlation, i.e. if X increases then Y decreases,
$\rho = +1$: `positive` correlation, i.e. if X increases then Y increases.

Fig. 5.4 compares the GDT vs. Rosetta score correlations on the left side and GDT vs. RMSD correlations on the right side for α-helical protein targets. As expected, we can see a strong negative correlation between GDT and RMSD values. At first glance, their linear relation appears to get stronger with growing protein size. Starting with $\rho = -0.91$ for BBL ($L = 47$ residues) the correlation coefficient rises according to amount up to $\rho = -0.96$ for Nanog homeodomain ($L = 62$ residues). The only exception is given by Lambda repressor, which is 92 residues long and shows a correlation of only $\rho = -0.89$. The main reason for this observation is that RMSD and GDT are both used to quantify the global alignment of such a large protein. Here, the values deviate much more since GDT scales better with local misalignments whereas RMSD can get disproportionately large, which affects the correlation. Nevertheless, the highly linear relation between GDT and RMSD confirms that they are nearly equally good for structure comparison, such that generalized observations based on one metric can be transferred to the other. Because GDT-based evaluations are more robust I solely focus on them to estimate the structure-quality or refinement level of generated decoys.

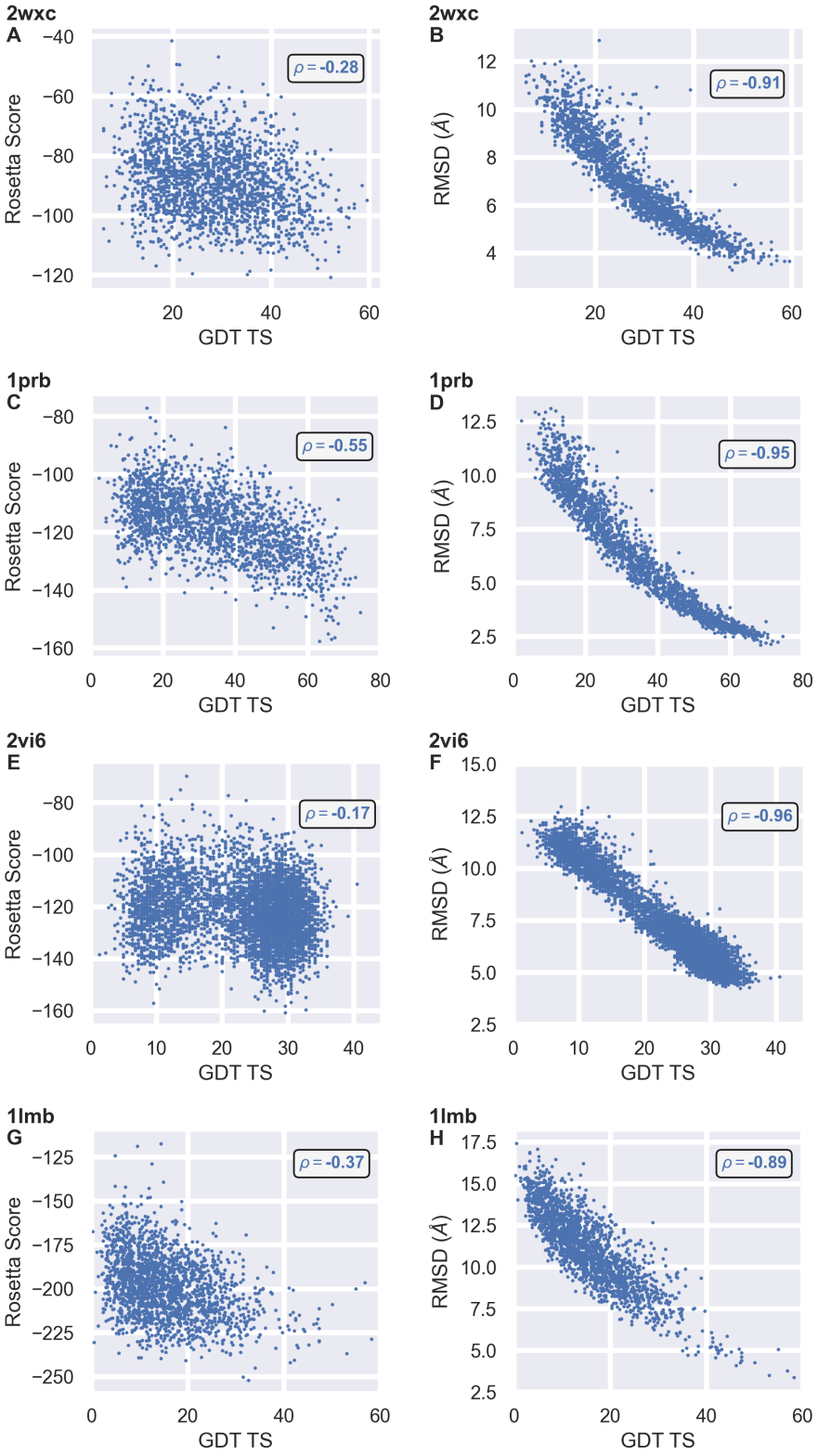

Figure 5.4. Correlation analyses of generated decoys (α-helical proteins). Scatter plots visualize the relation of the GDT TS vs. Rosetta score or backbone RMSD. The linear dependency of two variables is measured by the Pearson correlation coefficient ρ. **(A+B)** BBL (PDB id: 2wxc[208]). **(C+D)** Albumin-binding domain (PDB id: 1prb[209]). **(E+F)** Nanog homeodomain (PDB id: 2vi6[191]). **(G+H)** Lambda repressor (PDB id: 1lmb[210]).

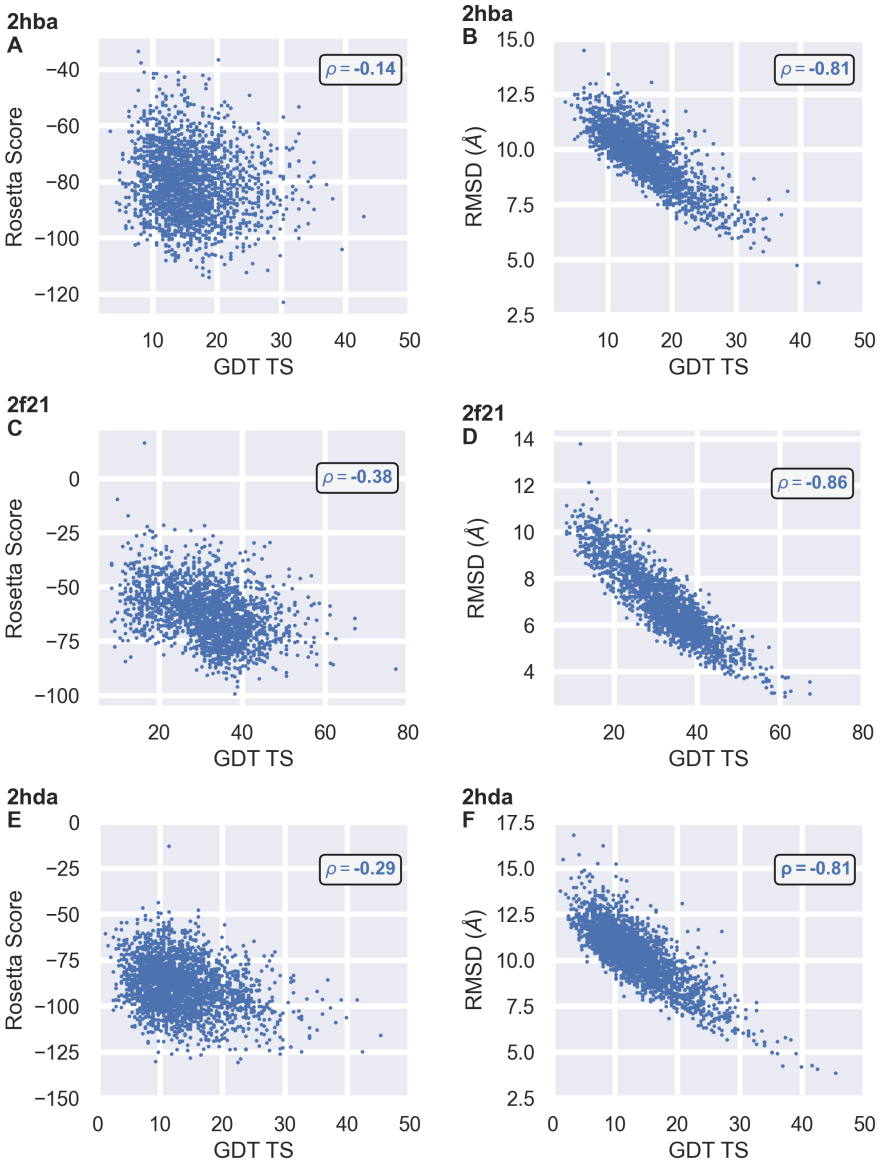

Figure 5.5. Correlation analyses of generated decoys (β-sheet proteins). Scatter plots visualize the relation of the GDT TS vs. Rosetta score or backbone RMSD. The linear dependency of two variables is measured by the Pearson correlation coefficient ρ. **(A+B)** N-terminal of L9 protein (PDB id: 2hba[211]). **(C+D)** WW domain of human Pin1 Fip mutant (PDB id: 2f21[212]). **(E+F)** Yes SH3 domain (PDB id: 2hda[192]).

Observed correlations between GDT and the used Rosetta scoring function (*REF2015*)[203,204] are less significant. The scatter plots on the left side of Fig. 5.4 are very bloated, i.e. their correlations are typically non-linear due to very high standard deviations. In case of Nanog homeodomain (cf. Fig. 5.4E) we can observe a correlation of only $\rho = -0.17$. Such decoys are basically uncorrelated with regard to GDT and the according Rosetta score mappings. However, one exception is given by the albumin-binding domain, where the correlation is much stronger with $\rho = -0.55$. Most notably, GDT and Rosetta scores are always negatively correlated for all tested proteins. This means that from a statistical point of view low-scoring structures are more likely to have higher GDT values assigned and vice versa. Such behavior is in accordance with a physical meaningful energy function, where a pathway of energetically favored structures can be seen as protein folding leading into a global minimum representing the native fold.

Correlation analyses of the decoys with β-sheet motifs yield very similar observations, as displayed in Fig. 5.5. The right-sided scatter plots with GDT vs. RMSD show ρ values between -0.81 and -0.86. This indicates that α-helical structures can be converted between GDT- and RMSD-based mappings more accurately as compared to β-sheet structures. Additionally, Rosetta score correlations are comparably low as previously observed for α-helical structures, which is shown on the left of Figs. 5.4 and 5.5. The observed values range from $\rho = -0.14$ to $\rho = -0.38$. Based on this fact, we can conclude that Rosetta scoring by itself does not provide sufficient information to reliably select high-quality structures out of the entire structure ensemble. It can be used as an additional indicator to select structures, but on its own it cannot be used with great confidence to estimate the refinement level of a protein model.

Next, I investigated how to classify the generated decoys and also how to better represent the (dis)-agreement between decoy energies and their refinement levels. For this purpose, I decided to visualize each feature via 2D surface plots, which makes it especially easy to compare them. Starting with the atom positions taken from the decoys, I computed C_α-distance matrices to represent the individual structures. In a following step I concatenated all matrices and applied a dimension reduction algorithm called multidimensional scaling (MDS) [149]. This method converts the input coordinates, in my case the $L \times L$ matrices with L being the sequence length, into (X,Y) coordinates. Additionally, the algorithm is designed to conserve distance information. In other words, the distance between two structures and therefore the difference between C_α distance matrices will be represented as the distance between two MDS points. This representation is well-suited for structure comparison and to grasp the overall variety of the generated decoy ensemble. By additionally using either the Rosetta scores or GDT scores as height information it is possible to generate the intended 2D surface plots. Note that Rosetta scores do scale with protein length, which is why I intentionally kept their color-scale relative, i.e. it can vary from figure to figure. GDT scores on the other hand are always mapped between 0 and 100. For this reason, the applied color-scale for GDT values is fixed and therefore the same for all figures.

Fig. 5.6 shows the energy surface (left side) and the refinement levels (right side) for α-helical decoys. The energy surfaces of all presented proteins are heterogeneous rugged with many little hills and valleys. Given the MDS-representation of the decoy structures, I expected to see a much smoother energy surface with clear separations of low- or high-energy areas. However, this is not the case and the energy landscapes appear more or less random. As displayed in Fig. 5.6(A+G), BBL and Lambda repressor have extremely small but distinct global minima located in the middle of the MDS plane. The best separation of energy states is achieved for the albumin-binding domain (see Fig. 5.6C). Here, the energy surface is much smoother as compared to the other cases and has a local valley positioned in the left centered region. Keep in mind that native-like structures are supposed to be located within minima of the energy surface. Besides, the absolute energy values are not that important for this kind of analysis but rather if it is possible to locate a distinct local or global minimum.

The surface representing the refinement-quality of the decoys, which is displayed on the right side of Fig. 5.6, is in line with my expectations. As mentioned previously, MDS conserves difference information of two structures, i.e. the difference between two structures is proportional to their MDS distance. By mapping the GDT values to the (X,Y) coordinates representing the individual decoys, we obtain a generally smooth landscape. According to this we can also clearly locate regions separating low- and high-refined structures. The *de novo* folding algorithm was able to generate decoys with GDT values of 80 for albumin-binding domain, approx. 64 for BBL and Lambda repressor and approx. 48 for Nanog homeodomain. The location of optimal structures can be precisely located for the first two proteins. The opposite is observed for Nanog homeodomain, where the valley containing the best structures is very broad and flat, as visualized in Fig. 5.6F. Here, the majority of structures reach GDTs of around 20

to 30. Lastly, the GDT surface of Lambda repressor indicates an overall flat surface as well but contains a few small and deep hot-spots where decoys reach GDTs between 48 and 64.

Note that both Rosetta scores and GDT scores use the same color schemes but with an inverse scaling. This allows us to directly compare relative changes of both surfaces and comprehend how reliable the applied Rosetta scoring function really is. A relatively good agreement can be observed for the albumin-binding domain, as depicted in Fig.5.6(C+D). The energy surface does resemble the GDT surface to some extent but not with the same level of detail. The local minimum region does overlap in both figures but its boundaries are much clearer on the GDT surface. In case of BBL and Nanog homeodomain, we can observe a mixed disagreement between the energy mappings and the according refinement levels. The visualized color patterns do still resemble each other to some extent. However, the energy surface is much more random as compared to the relatively smooth GDT surface. BBL's energy surface also shows multiple very small local minima, located in the centered and left centered regions. Since these minima have the same depth, they cannot be used to infer the location of the best decoy structures. The energy surface of Lambda repressor is also very rugged and random but does have a clear global minimum at the center of the MDS plane. This position is in agreement with the GDT mapping. In this case, it is possible to correctly locate the high-refined decoys with the Rosetta score function.

Following the same discussion for the β-sheet structures, Fig.5.7 summarizes the surface plots for NTL9, WW domain of human Pin1 Fip mutant and Yes SH3 domain. Similar to before, the overall shapes of the energy landscapes do not meet the expectations. The first protein contains many equally deep local minima spread across a large portion of the MDS plane, making it impossible to pinpoint any good structures. The WW domain shows an overall broad and flat valley with a few small spots going even deeper. A similar-looking energy surface is also shown for Yes SH3 domain. Just looking at such energy surfaces indicates that the Rosetta scoring function does not provide sufficient information to make a decisive decoy selection for these proteins.

When looking at the GDT surfaces, we can see that the folding algorithm performs much worse for β-structures as compared to α-helical structures. NTL9 and Yes SH3 decoys reached GDTs of 48 but the majority of generated decoys have GDTs below 24. Contrary to that, WW domain decoys reached extremely high GDT values up to 80 and the majority being at around 40 to 50. However, note that WW domain is the smallest test protein with a size of only 39 residues. During the setup of this study I did not adjust the number of fragment insertions, i.e. all proteins were constructed with exactly 3x 3-mer and 1x 9-mer insertions followed by the normal folding cycles. Consequently, WW domain should be refined for about 50% of the entire structure due to the fragment insertions alone. This is reflected in the high counts of well-refined decoy structures, as illustrated in Fig. 5.7D.

The agreement comparison between energy surface and refinement levels are especially interesting for the β-sheet structures. Surfaces which belong to the same protein do actually have a relatively strong resemblance, as long as only relative topology changes are compared to each other. The color patterns of all three proteins are very similar but have naturally different color-mappings since GDT is fixed and Rosetta scores are not. However, the general performance of the *de novo* folding algorithm for β-sheet structures was rather poor. The performance for such proteins is mainly dependant on fragment insertions and not on the folding cycles itself, as exemplary shown by WW domain. This makes sense since α-helices can change in size with minimal conformational changes whereas β-sheets must undergo large conformational changes. To compensate this it is necessary to either increase the number of fragment insertions or the folding cycles according to the protein size.

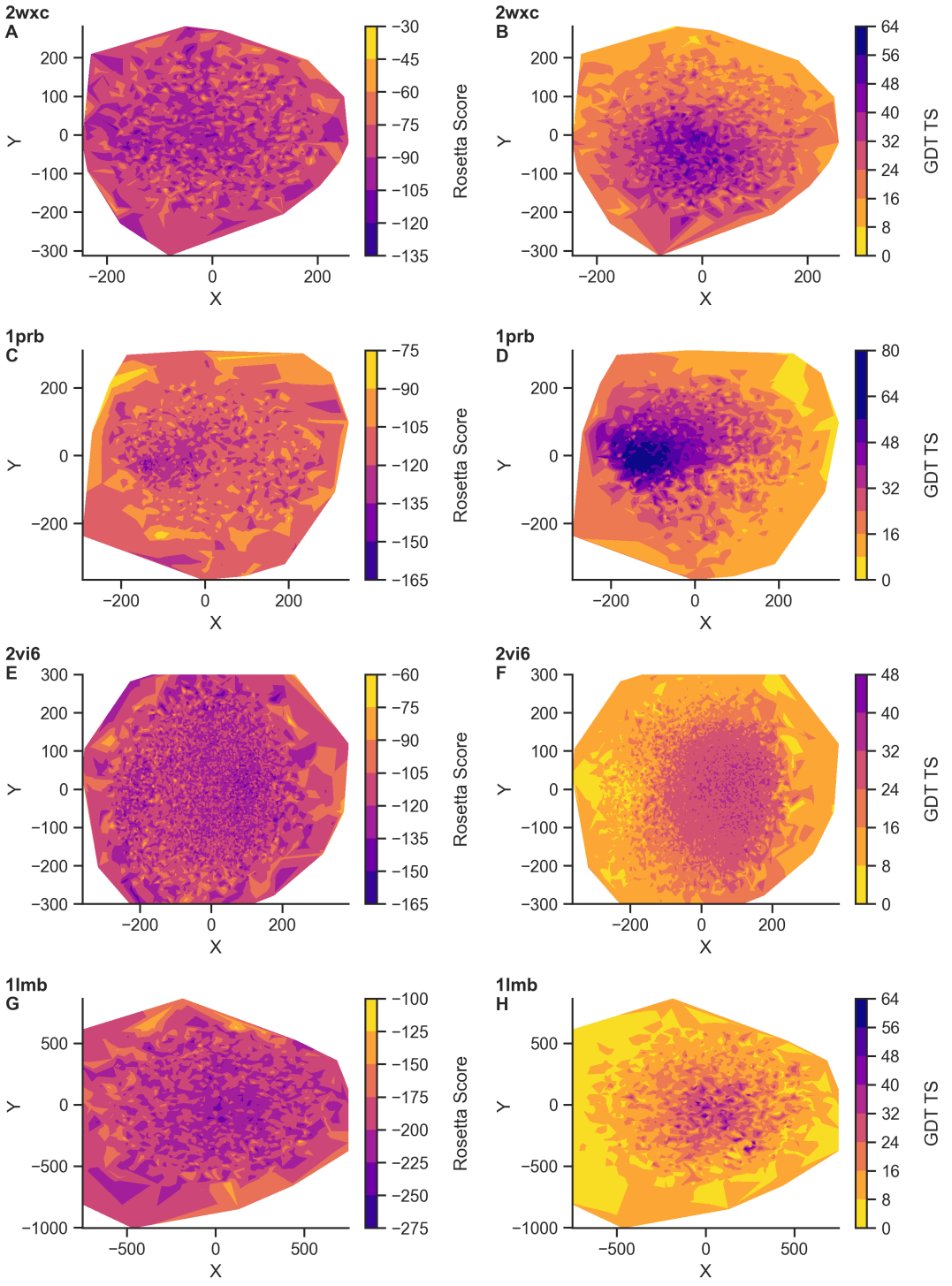

Figure 5.6. Surface plots of generated decoys (α-helical proteins). Figures visualize the interpolated three-dimensional energy surface (Rosetta score) or refinement levels (GDT TS) of generated decoys. Decoys are represented by (X,Y) coordinates, which were obtained via multidimensional scaling of C_α distance matrices. **(A+B)** BBL (PDB id: 2wxc[208]). **(C+D)** Albumin-binding domain (PDB id: 1prb[209]). **(E+F)** Nanog homeodomain (PDB id: 2vi6[191]). **(G+H)** Lambda repressor (PDB id: 1lmb[210]).

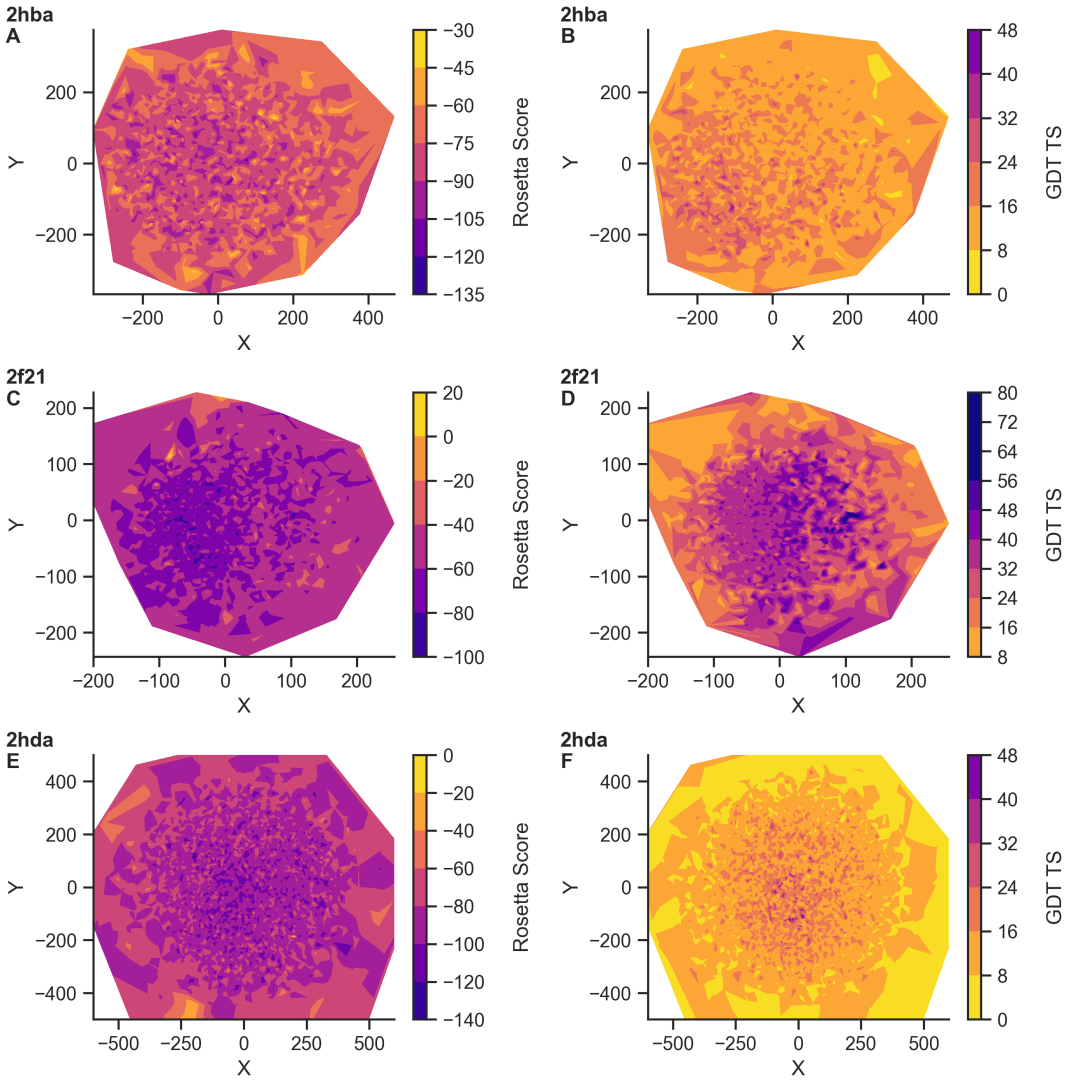

Figure 5.7. Surface plots of generated decoys (β-sheet proteins). Figures visualize the interpolated three-dimensional energy surface (Rosetta score) or refinement levels (GDT TS) of generated decoys. Decoys are represented by (X,Y) coordinates, which were obtained via multidimensional scaling of C_α distance matrices. **(A+B)** N-terminal of L9 protein (PDB id: 2hba[211]). **(C+D)** WW domain of human Pin1 Fip mutant (PDB id: 2f21[212]). **(E+F)** Yes SH3 domain (PDB id: 2hda[192]).

5.3 Decoy Selection

After generating sufficient structures to populate replicas with unique starting conformations it is necessary to discuss how to actually select them. The main motivation is to select different structures in order to maximize the variety from the very beginning. This way, the REX simulation can probe a wider conformation space and attempt multiple pathways towards the native state. At the same time, the starting structure ensemble should not be spread out too much and have a high ratio of well-refined structures. The primary goal is still to reach native-like conformations within a single REX run. The smaller the distance between starting structures and native fold, the sooner the convergence due to the integrated bias potential. In the following section i will discuss two straight forward decoy-selection methods, i.e. the *direct* and the *KMEANS* method, and compare them to each other.

The direct method consists of simply ranking the decoys in ascending order based on the Rosetta scores and then select N_{rex} decoys. As previously discussed in section 5.2, Rosetta and GDT scores have been shown to be always negative correlated to each other. In most cases the correlation is rather low but it is still significant enough that more native structures tend to get mapped with lower Rosetta scores. From this point of view, simply selecting the lowest-scoring decoys appears as a valid approach. But the analysis also showed that the standard deviations are extremely high which is not ideal. A selection solely based on Rosetta scores is very likely to contain many low-refined structures as well. The question remains: *How much variety can we expect from this method? Can we see any patterns, i.e. are selected decoys primarily located within the high-refinement region or are they randomly distributed?*

The other method is very similar but requires additional pre-analysis of the decoy data. Here, we utilize the MDS representation and the fact that the distance between two data points is an important source of information. After projecting all generated decoys onto the MDS plane, it is possible to systematically select structures from different regions and thus guarantee a high variety of structures. For example, one can split the decoys into groups organized as a grid or a tessellation. Without a doubt the cell sizes must be adjusted based on the overall size or shape of the MDS plane made up by the structure locations. This can be easily accomplished with algorithms such as natural neighbor interpolation (Voronoi tessellation) [172,173] or KMEANS clustering [163,213]. Whenever possible the natural neighbor interpolation splits the data into regions with an equal member count. KMEANS can achieve this too but typically does not. Instead, KMEANS initiates with k randomly positioned cluster centers. During its execution the algorithm optimizes the cluster center locations and assigns data points towards their closest centers. If the data is equally distributed this would result in equally sized clusters. Most of the time, however, this is not the case and the data contains both dense and thin population regions resulting in differently sized clusters. This property is advantageous for the intended use-case and can be applied to increase the variety of selected decoys. Since dense regions on the MDS plane have very similar structures it is sufficient to select only one of the corresponding decoys and obtain more structure variety from other clusters. The KMEANS-selection method basically classifies decoys into clusters with a radius of approximately

$$r_{\text{cluster}} \approx \frac{N_{\text{decoys}}}{N_{\text{clusters}}}, \tag{5.4}$$

where N_{decoys} is the total number of decoys and $N_{\text{clusters}} = k$ is the number of KMEANS cluster centers. This time the ranking is performed on entire clusters and their average Rosetta scores. However, the final selection is still based on the lowest-scoring decoys but with the condition that only one decoy from each cluster is allowed. This concept combines the positive aspects of MDS and the observed correlation coefficients between Rosetta scores vs. GDT scores. The resulting decoy selection is therefore guaranteed to be more diverse and has a statistically higher chance of containing well-refined structures.

Note that the performance of the KMEANS method is heavily dependant on the cluster radius defined by Eq. 5.4. Given that the number of generated decoys is fixed, a proper selection of cluster centers is essential. Fig. 5.8 exemplarily visualizes the final selection of 80 decoys for different cluster counts k for BBL and Nanog homeodomain. As evidently shown, low values of k are associated with large cluster radii, i.e. the selected decoys are spread out over the entire MDS plane. When k is increased the cluster radii are reduced accordingly, which in turn shrinks the occupied decoy area and concentrates it towards the high-refinement regions. This effect can be explained by the count ratio of selected decoys with respect to the KMEANS clusters. For example, with $k = 400$ the MDS plane gets split up into 400 approximately evenly large regions. But with 80 selected decoys only a small fraction of the entire MDS plane contributes towards the selection ensemble.

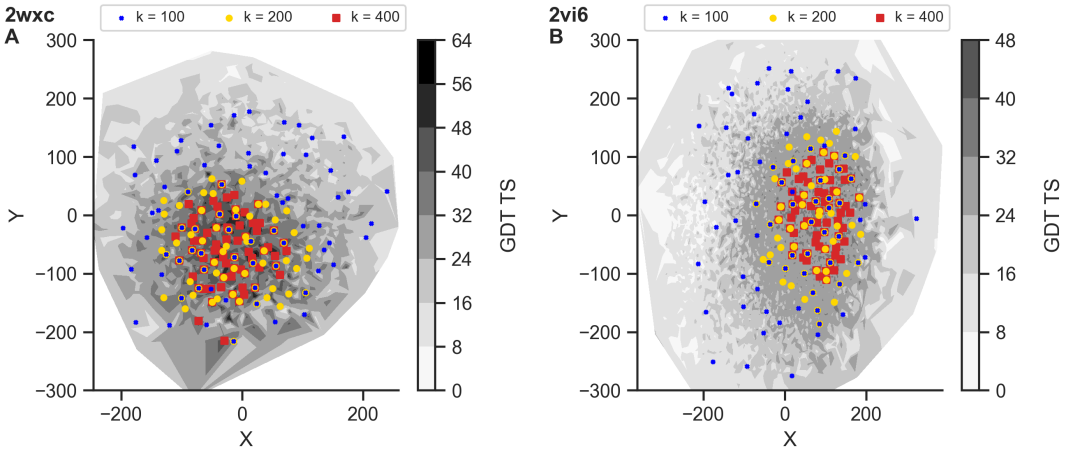

Figure 5.8. KMEANS-selection method performance for different cluster radii. Figures visualize the dependency of the final decoy selection in relation to different cluster radii which are defined by the choice of cluster centers k (blue, green, red) according to Eq. 5.4. Decoys are represented by (X,Y) coordinates which were obtained via multidimensional scaling of C_α distance matrices. Additionally, the interpolated surface of refinement levels (GDT TS) is indicated by the gray scale. **(A)** BBL (PDB id: 2wxc[208]). **(B)** Nanog homeodomain (PDB id: 2vi6[191]).

Since the selection order is based on average Rosetta score rankings of entire clusters, low-scoring decoys from these regions are singled out and all others are ignored. This is clearly shown in Fig. 5.8, where the area that is spanned by uniquely colored markers is inverse proportional to the cluster count k.

In order to compare the performance of the two presented selection methods, i.e. direct vs. KMEANS method, I choose exactly 80 decoys from a set of 5000 generated decoys and apply $k = 400$ KMEANS clusters. Fig. 5.9 gives an overview of the final decoy selections based on the applied method for α-helical proteins. Starting the comparison with the direct method, we can observe that simply choosing the lowest-scoring decoys results in a high variety of structures with respect to their GDT refinement. As illustrated in blue in Figs. 5.9(A,D,G,J), measured GDT values range from 15 to 55 for BBL, 15 to 75 for albumin-binding domain, 5 to 35 for Nanog homeodomain and 0 to 55 for Lambda repressor. According decoy mappings on the MDS plane are shown in Figs. 5.9(B,E,H,K). We can see that in three cases the decoys represent a very large conformational space and are spread out over the majority of the MDS plane. Contrary to that, selected decoys of the albumin-binding domain are primarily focused on a very small MDS region representing high-quality structures. In this specific case, the decoy selection is the result of an especially good mapping between Rosetta scores and GDT. The region is of high interest for the intended application but unfortunately corresponding decoys do not cover a large conformational space, offering only minimally different pathways towards the native state. For the albumin-binding domain, we can observe a mixed spread of decoys, i.e. many decoys are clustered around the high-refinement area while others are located very far away. Such unrefined structures are also sub-optimal for contact-guided REX MD because they will most likely not fold within the relatively short simulation times.

Investigation of the KMEANS method, which is illustrated in red in Fig. 5.9, indicates much more promising results. The scatter plot analyses of Figs. 5.9(A,D,G,J) display a diverse spread of decoys similar to the direct method. This time, however, the selections are mainly located in the scatter plot regions corresponding to higher GDT values. The associated GDT values range from 24 to 55 for BBL, from 36 to 76 for albumin-binding domain, from 24 to 36 for Nanog homeodomain, and lastly from 10 to 59 for Lambda repressor.

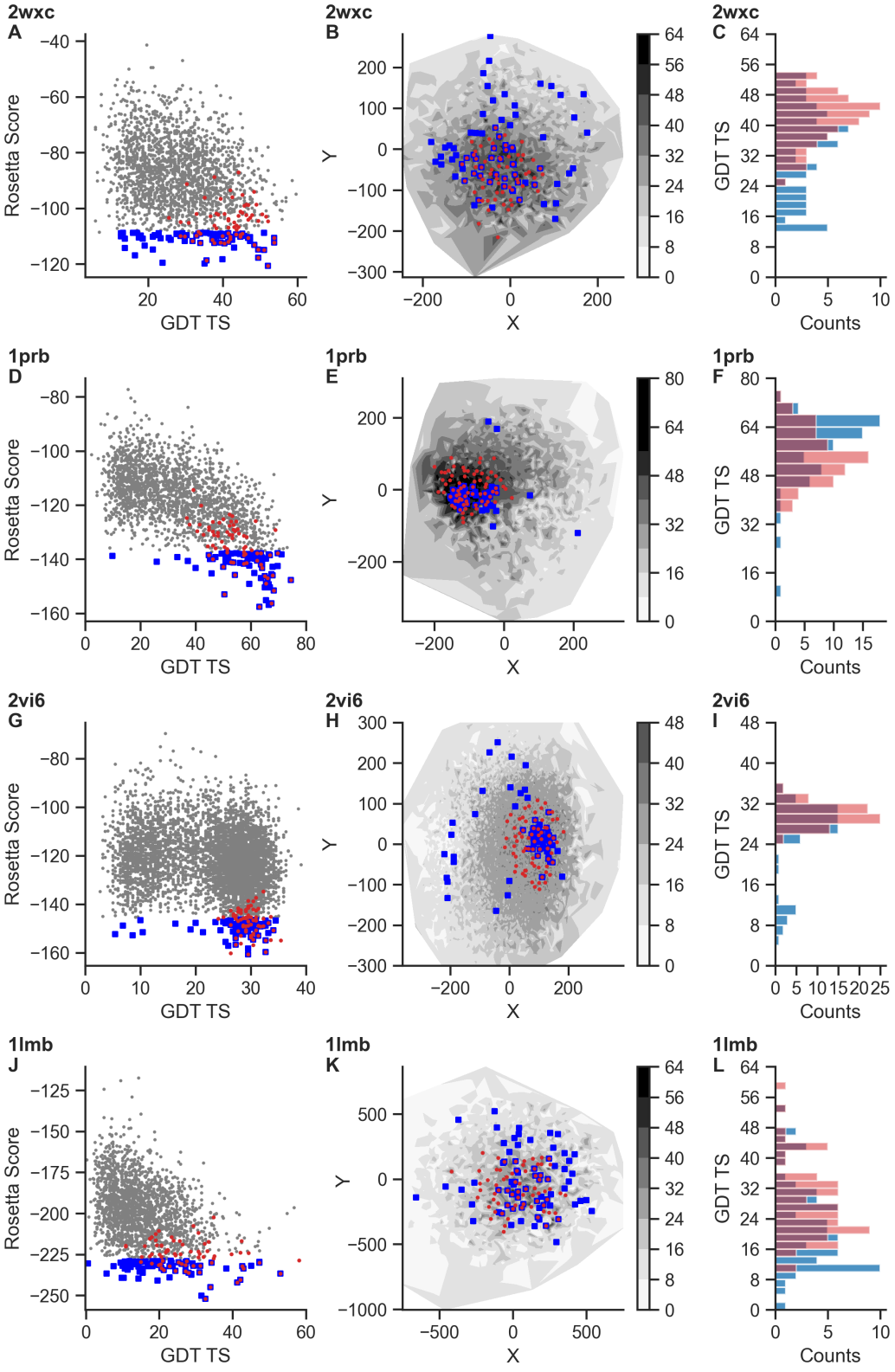

Figure 5.9. Comparison of different selection methods (α-helical proteins). Overview of different representations to display and compare the final decoy selections based on the applied method, i.e. direct method highlighted in blue and KMEANS method highlighted in red. **(left: A,D,G,J)** Scatter plots of Rosetta scores vs. GDT TS. **(middle: B,E,H,K)** Interpolated surfaces with refinement levels. Decoys are represented by (X,Y) coordinates which were obtained via multidimensional scaling of C_α distance matrices. **(right: C,F,I,J)** GDT distributions of selected decoys. **(A-C)** BBL (PDB id: 2wxc[208]). **(D-F)** Albumin-binding domain (PDB id: 1prb[209]). **(G-I)** Nanog homeodomain (PDB id: 2vi6[191]). **(J-L)** Lambda repressor (PDB id: 1lmb[210]).

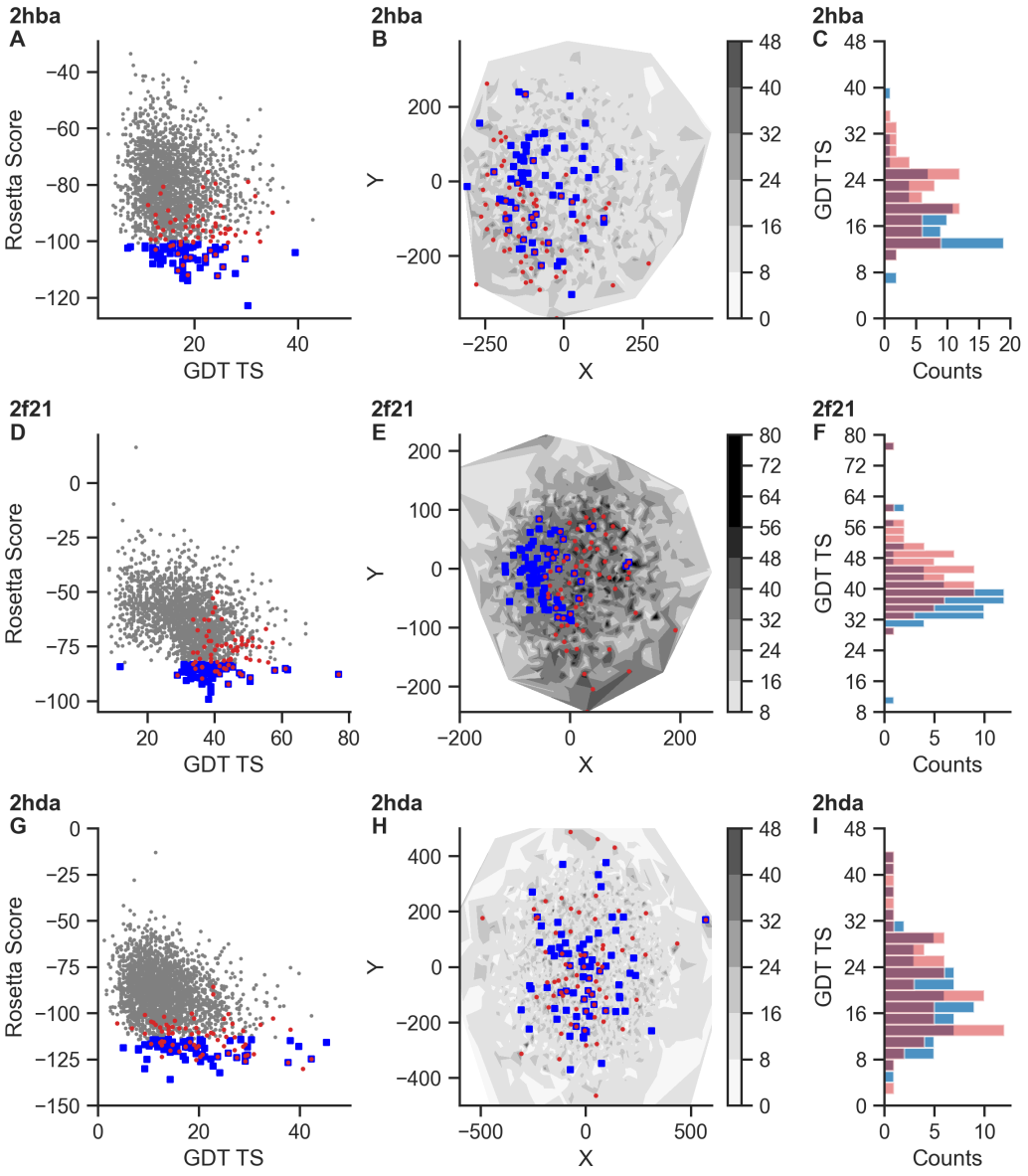

Figure 5.10. Comparison of different selection methods (β-sheet proteins). Overview of different representations to display and compare the final decoy selections based on the applied method, i.e. direct method highlighted in blue and KMEANS method highlighted in red. **(left: A,D,G)** Scatter plots of Rosetta scores vs. GDT TS. **(middle: B,E,H)** Interpolated surfaces with refinement levels. Decoys are represented by (X,Y) coordinates which were obtained via multidimensional scaling of C_α distance matrices. **(right: C,F,I)** GDT distributions of selected decoys. **(A-C)** N-terminal of L9 protein (PDB id: 2hba[211]). **(D-F)** WW domain of human Pin1 Fip mutant (PDB id: 2f21[212]). **(G-I)** Yes SH3 domain (PDB id: 2hda[192]).

Analogously, the MDS representations (with highlighted decoy selections) show a significant improvement as well. As evident to Figs. 5.9(B,E,H,K), we can see that selected decoys yield structures with overall good refinement quality while simultaneously providing sufficient structure variety. Due to the large count difference between KMEANS clusters and selected decoys, the majority of unwanted structures are filtered out, leaving only good candidates for the final selection. Within the cluster selections only the lowest-scoring decoy of each cluster is selected, which in turn increases the conformation variety as intended. Note that similar to before, all selected decoys of albumin-binding domain have extremely high GDT refinements compared to the other tested proteins.

But contrary to the direct method where the decoys were very dense located, the KMEANS method achieves a much wider spread on the conformational space. Based on the observed patterns for α-helical proteins, this method indicates a good balance between straightway structure refinement and variety, which should improve the expected performance of contact-guided REX MD.

Analogous comparison figures related to β-sheet structures are summarized in Fig. 5.10. As mentioned previously, the overall achieved refinement quality for the proteins NTL9 and Yes SH3 domain were rather low. GDT values reached only up to 40, with the majority being below 30. Closely inspecting the direct method selections, we can see that NTL9 contains decoys with GDTs between 6 and 40, WW domain between 10 and 78, and Yes SH3 domain GDT values between 6 and 45. Note that WW domain typically shows GDTs below 60. Only a single decoy out of 5000 was able to reach a GDT of almost 80 and also got selected. Further inspecting the decoy locations within the MDS representation, as visualized in Fig. 5.10(B,E,H), we can see that the direct method results in a wide spread of conformations for NTL9 and Yes SH3 domain. Given that the refinement surfaces for these proteins have no definite high-refinement regions, this wide spread of conformations is definitely useful to be applied as initiating REX structures. Contrary to that, the selected decoys of the WW domain are more densely packed around the left centered region on their according MDS plane, as shown in Fig. 5.10E. Their occupied MDS area is rather small in comparison to the other two proteins. Further comparing with the full-colored MDS representation of Fig. 5.7D, we can see that the majority of selected decoys are not located around the high-refinement region which center has the MDS coordinates of approx. (100,0). The according GDT distribution of Fig.5.10F verifies that as well, showing that most picked conformations have GDTs slightly below 40.

In order to finalize the comparison, it is now time to investigate the results of the KMEANS method for the same β-sheet structures. The scatter plots with Rosetta score vs. GDT of Figs. 5.10(A,D,G) indicate the same patterns as previously observed for α-helical structures. The selected decoys indicate a wide spread on the scatter plot, but unlike before they are not so strongly focused on the high GDT regions. More precisely, only the WW domain of human Pin1 Fip mutant shows a strong tendency towards higher quality structures. The other two cases are of comparable quality as during the direct method but with slightly better GDT averages. A closer inspection of the decoy locations with respect to their refinement levels, as visualized on the MDS planes of Figs. 5.10(B,E,H), reflect similar tendencies. The expected structure variety for NTL9 is on par with the other methods selection. Their occupied area is about equally large but the decoy structures belong to different MDS locations. Very similar results are observed for Yes SH3 domain, but this time the KMEANS method spans a much wider area over the MDS plane. Given that this highly increases the variety, it is still not optimal as it picks up some structures which are basically on the edge of the MDS plane with GDTs close to zero. Such poor quality structures are unlikely to be useful for contact-guided REX MD. Lastly, the decoy selections of the WW domain look much more promising as compared to the direct method. Not only is the decoy area larger than before containing more varying structures, now it is also shifted towards the optimum around (100,0) as indicated in Fig. 5.7D.

To finalize the comparison between the two presented decoy-selection methods, I obtained the GDT statistics corresponding to the final decoy selections and summarized them in Table 5.2. Additionally, I computed an approximation of the area which is occupied by the clusters belonging to the selected decoys. This allows me to estimate the structural variety of the decoy selection. More precisely, the overlap between the selection area with the entire MDS plane can be used as a measure reflecting the decoy variety. As shown in Table 5.2 and in Figs. 5.10(C,I,L), observed GDT statistics for the proteins BBL, Nanog homeodomain and Lambda repressor indicate that the KMEANS method performs much better.

Table 5.2. Comparison of decoy-selection statistics based on selection method. Table contains the protein's PDB id and statistics of the corresponding decoy selections (cf. Figs. 5.9 and 5.10 for colored representations of selection methods). Listed are the mean (μ), standard deviation (σ), minimum and maximum value of the method-based GDT distributions. Additionally, the overlapping area, which is defined by the selected cluster regions in relation to the entire MDS plane, is computationally approximated and listed in percent. Analogously to previous analyses, the table is split up into α-helical proteins (BBL, albumin-binding domain, Nanog homeodomain, Lambda repressor) and β-sheet proteins (NTL9, WW domain, Yes SH3 domain).

	direct method (blue)					KMEANS method (red)				
PDB id	area (%)	GDT_μ	GDT_σ	GDT_{min}	GDT_{max}	area (%)	GDT_μ	GDT_σ	GDT_{min}	GDT_{max}
2wxc	36.4	34.1	11.4	12.8	53.7	19.1	41.9	6.5	25.5	53.7
1prb	12.9	57.7	10.9	9.9	74.5	15.8	54.2	8.4	36.8	74.5
2vi6	22.4	25.5	7.7	5.5	35.5	12.5	29.6	2.1	24.1	35.5
1lmb	33.0	22.6	11.2	0.5	53.0	21.6	27.7	9.6	10.9	58.2
2hba	29.1	17.8	5.4	7.2	39.4	30.9	20.8	5.6	11.1	35.1
2f21	19.4	38.8	8.5	11.8	77.0	28.6	43.7	7.7	28.9	77.0
2hda	34.2	19.5	7.8	5.1	42.4	39.0	20.5	8.1	3.8	42.4

The biggest impact is that extremely low GDT structures are filtered out, thus shifting the distribution towards higher GDT values. The only exception is observed for albumin-binding domain. But as mentioned previously, here the direct method yields only very little structure variety. On the other hand, by using the KMEANS method the spanned area is enlarged. In this case, it is even concentrated around the high-quality region indicating an overall better performance yet again. The comparison of the occupied MDS area in relation to the entire MDS plane typically yields much higher values for the direct method as compared to the KMEANS method. We observe, e.g., for BBL 36.4% vs. 19.1%, for Nanog Homeo domain 22.4% vs. 12.5%, and for Lambda repressor 33.0% vs. 21.6%. As already discussed, albumin-binding domain shows the opposite, namely 12.9% vs. 15.8%.

However, these numbers alone are not sufficient to reflect the expected structure variety. I want to emphasize that it is also important to consider how dense the spanned area is and where it is located. For example, the albumin-binding domain yields a smaller region with the direct method. At the same time most of the decoys are located very close to each other, while others are separated very far away (cf. Fig.5.10E). For the intended application as starting conformations, we do not obtain much structure variety close to the region of interested but we also get a few decoys with low-quality refinement. Contrary to that, the KMEANS method yields a decoy selection which is primarily focused around the region of interest. At the same time it also provides good structure variety due to the spacing of adjacent decoys. Therefore, the KMEANS method significantly outperforms the direct method, even though the GDT statistics may indicate the opposite.

The comparison of β-sheet structures and their corresponding decoy selection areas shows an opposite trend. This time, KMEANS-method derived decoys occupy a larger area. The approximated area overlaps of direct method vs. KMEANS method are 29.1% vs. 30.9% for NTL9, 19.4% vs. 28.6% for WW domain, and lastly 34.2% vs. 39.0% for Yes SH3 domain. The MDS representations displayed in Figs. 5.10(B,E,H) indicate equally good structure variations for NTL9 and Yes Sh3 domain. Only the WW domain can achieve slightly better results with the KMEANS method, mainly due to the distance between selected decoys and the region of interest around (100,0). The analyses of the GDT distributions clearly favor the KMEANS method over the direct method. In all cases the distributions are slightly shifted towards higher GDT values, as reflected by the mean values and shown in Figs. 5.10(C,F,I).

5.4 Summary

I introduced the concept of starting-structure generation based on its application purpose, i.e. to generate unique and diverse starting structures for contact-guided REX MD. In section 5.1 I explained the theoretical background and how additional structure variety is beneficial for the overall REX performance due to the introduction of new pathways towards the native fold. Additionally, starting-conformation bias gets reduced and erroneous bias signals are not equally contributing to each replica. I also introduced a method which can generate structure ensembles within short periods of time. More precisely, I explained how fragment insertion works and how it is integrated into my *de novo* folding algorithm to improve the decoy generation process.

In section 5.2 I followed up by systematically analyzing and comparing the generated decoy ensembles. I investigated correlations such as Rosetta score vs. GDT TS and analyzed how reliable the energy mappings are. It was shown that the correlations were negative for all test proteins, which does indeed reflect a physically meaningful energy function where low energy states are expected to be more stable and better refined. Furthermore, I obtained interpolated surfaces reflecting the energy and refinement levels by projecting the generated decoy structures onto 2D via multidimensional scaling. The comparison of both surfaces gave additional insights into how accurate Rosetta scores are. Overall the energy surfaces were often heterogeneous rugged. However, for some proteins it was possible to obtain energy surfaces which were relatively smooth and indicated clear separations between high- and low-energy states. The observed patterns also showed that the scoring function performs much better for α-helical structures as compared to β-sheet structures. A further investigation of the refinement levels in terms of GDT mappings showed that the *de novo* folding algorithm performs really well for α-helical proteins and could achieve high-quality conformations during the generation process. Although β-sheet proteins also contained a few highly refined structures, the majority was of rather low quality with GDTs below 30.

Finally, in section 5.3 I presented two valid approaches, i.e direct method and KMEANS method, in order to select decoys as starting conformations for contact-guided REX MD. I compared the methods based on their resulting decoy selections while focusing on two aspects: 1) the structure quality and 2) the structure variety of the obtained decoy selections. To recap, both methods are highly depending on the underlying quality of the generated decoy ensemble, i.e. on the performance of the *de novo* folding algorithm. Whenever clear separations between high- and low-refinement regions are available, the KMEANS method drastically outperforms the direct method, as shown for e.g. α-helical structures. When this is not the case, both methods are about equally good but KMEANS is still slightly favored. The main reason for this is that KMEANS method does enforce a structural variance as a consequence of the cluster radii and the condition that only one decoy is allowed from each cluster. Additionally, the discrepancy between the number of selected decoys and the number of KMEANS clusters does affect the area belonging to selected clusters. By finding a good ratio it is possible to focus the decoy selection onto regions related to higher refined structures while maintaining sufficient structure variety. This shows that the KMEANS method is in general a much better approach for the intended use-case.

6

Ensemble Selection

This chapter covers one of the most critical topics, namely how to select a representative ensemble out of the vast number of computationally generated structures with REX MD. Section 6.1 describes my ensemble-selection study, where I investigate various methods to make such a selection. I give an overview of the used protein systems and introduce four very robust ensemble-selection methods. In section 6.2 I briefly discuss the achieved model accuracy of the structures which I obtained via REX simulations. In the following section 6.3 I outline the steps of my investigated ensemble-selection methods and introduce relevant techniques. I then continue with a detailed comparison of the different selection methods and state their strengths and weaknesses in section 6.4. Furthermore I introduce a numerical rating to objectively compare each method's performance and to indicate their reliability to select high-quality structures. Lastly in section 6.5 I conclude my findings and recap the most important aspects of the performed study. This Chapter is based on my article named "Selection of representative structures from large biomolecular ensembles" (2022)[3], published by the Journal of Chemical Physics.

6.1 Study Concept

There are many different *in-silico* approaches to obtain highly accurate protein structure models, be it either *de novo* or, e.g., via refinement in a physical force field. As previously shown in chapter 4, contact-guided REX MD is a suitable method to generate large structure ensembles which is also capable to enrich native-like conformations. One of the most challenging tasks is to select a representative member out of the vast variety of generated structures. Mimicking a *blind-prediction scenario*, I present here a combination of different techniques to reliably select highly native-like structures. All eligible structures were generated in contact-guided REX MD over 500 ns using five mid-sized proteins.

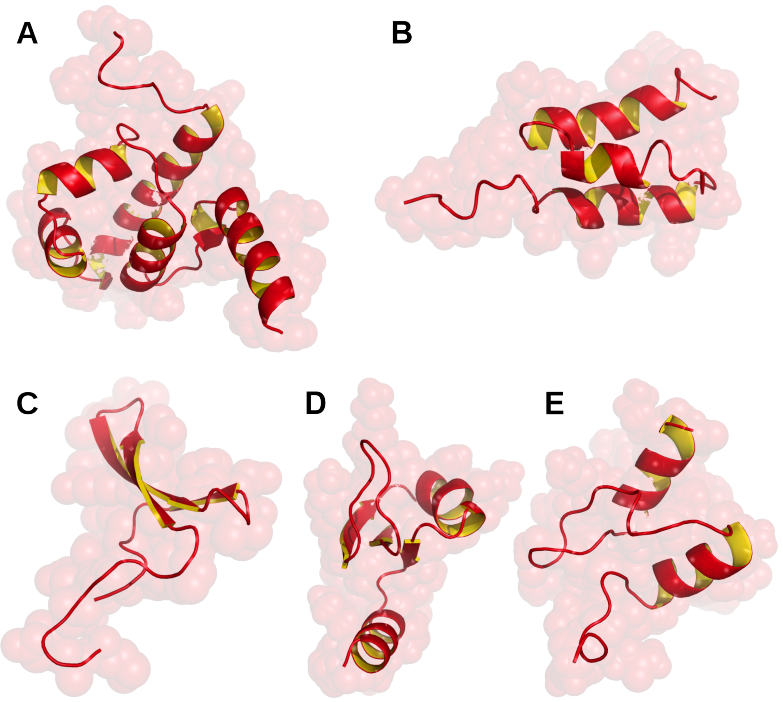

Figure 6.1. Native structures of proteins used for ensemble-selection study. (A) Lambda repressor (PDB id: 1lmb[210]). **(B)** Albumin-binding domain (PDB id: 1prb[209]). **(C)** WW domain of human Pin1 Fip mutant (PDB id: 2f21[212]). **(D)** N-terminal of L9 protein (PDB id: 2hba[211]). **(E)** BBL (PDB id: 2wxc[208]). Visualized in PyMol[189,190]. Adapted from Ref.[3] under CC BY 4.0.

To minimize the correlation between initial replica structures and the generated REX ensembles, I populated each replica with unique conformations as described in chapter 5. Appendix tables E.1 and E.2 give an overview of the starting decoy accuracy measured by GDT and RMSD, respectively. Integrated REX bias contacts were obtained from a deep residual neural network called ResTriplet[97,98]. Following the bias guidelines of section 4.1.4, I selected approximately $\frac{3}{4}L$ contact pairs and adjusted individual coupling strengths for bias contacts that were clustered within the contact map. A detailed listing of the used bias contacts is given in appendix table E.3, and visualized in Figs. E.1-E.5 as contact maps. Note that the 500 ns long REX trajectories can also be used to assess the accuracy and limitations of REX and the underlying `AMBER99SB-ILDN`[110] force field when applied to very large protein systems.

The tested proteins have lengths between 39 and 92 residues and cover multiple unique folds with varying structure complexity and secondary structure motifs. The largest test protein is the Lambda repressor (PDB id: 1lmb[210]). In this case I only used the second dimer chain, which is composed of six α-helices in different orientations and has a folding time of approximately 49 μs. The second test protein is given by the albumin-binding domain (PDB id: 1prb[209]) with an extremely short folding time in the order of 3.9 μs. Its structure is 53 residues long and consists of three α-helices orientated as a helical bundle.

Table 6.1. Overview of proteins used for ensemble-selection study[3]. Left side of the table contains structure-related information. The right side lists the number of used bias contacts during contact-guided REX MD and the corresponding true-positive rate (TPR).

name / description	PDB id	folding time	length	bias contacts	bias TPR (%)
Lambda repressor	`1lmb`	49 μs	92	70	87
Albumin-binding domain	`1prb`	3.9 μs	53	40	82
WW domain of human Pin1 Fip mutant	`2f21`	21 μs	39	30	96
NTL9 (N-terminal of L9 protein)	`2hba`	29 μs	52	40	95
BBL	`2wxc`	49 μs	47	35	91

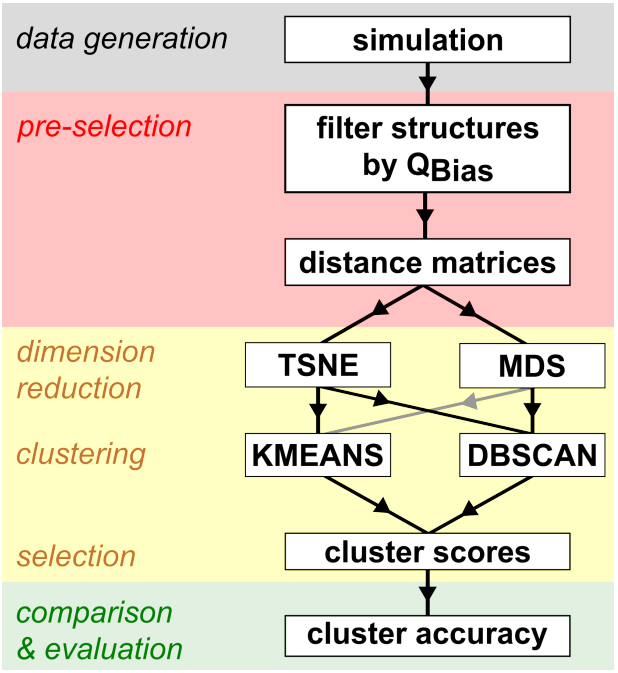

Figure 6.2. Overview of investigated ensemble-selection methods. Flowchart shows all steps which were performed during the study to investigate different ensemble-selection methods. Each stage is highlighted in another color: **I)** data generation (gray), **II)** pre-selection of structures (red), **III)** ensemble-selection algorithm chains (yellow), **IV)** comparison and evaluation (green) of algorithm chains. Reproduced from Ref.[3] under CC BY 4.0.

The third test structure is represented by a small segment of an entire protein, more precisely the WW domain of human Pin1 Fip mutant (PDB id: 2f21[212]). This β-sheet structure is only 39 residues long and has a folding time of 21 μs. The test system representing a mixed structure of α-helices and β-sheets is given by the N-terminal of L9 protein (NTL9, PDB id: 2hba[211]). Its protein size is 52 with a folding time around 29 μs. Lastly is BBL (PDB id: 2wxc[208]), which α-helical structure is 47 residues long and has a folding time of approximately 49 μs. Fig. 6.1 displays the native conformations of the simulated proteins, which also correspond to the target structures of the ensemble selections investigated during this study. Table 6.1 gives an overview of the performed REX simulations by listing the mentioned proteins including their sizes, folding times and the number of used bias contacts. Reported folding times were obtained as average lifetime in the unfolded state observed in MD simulations[34].

The main objective of this study was to obtain a method to reliably select highly native-like conformations from the generated REX trajectory. For this purpose I investigated a total of four ensemble-selection methods and compared their performance in achieving this task. Each method applies a combination of different algorithms and techniques, such as dimension reduction or clustering. To get a better overview of the investigated methods, I divided them into meaningful stages based on their application purpose. As summarized in Fig. 6.2, the stages are I) data generation, II) pre-selection of structures, III) ensemble-selection algorithm chain, and IV) comparison and evaluation. Note that the four investigated ensemble-selection methods only deviate at stage III), where I apply different methods for dimension reduction or clustering. A detailed explanation of the four possible pipelines, i.e.

1) TSNE → KMEANS,
2) MDS → KMEANS,
3) TSNE → DBSCAN,
3) MDS → DBSCAN,

will be done later in section 6.3.

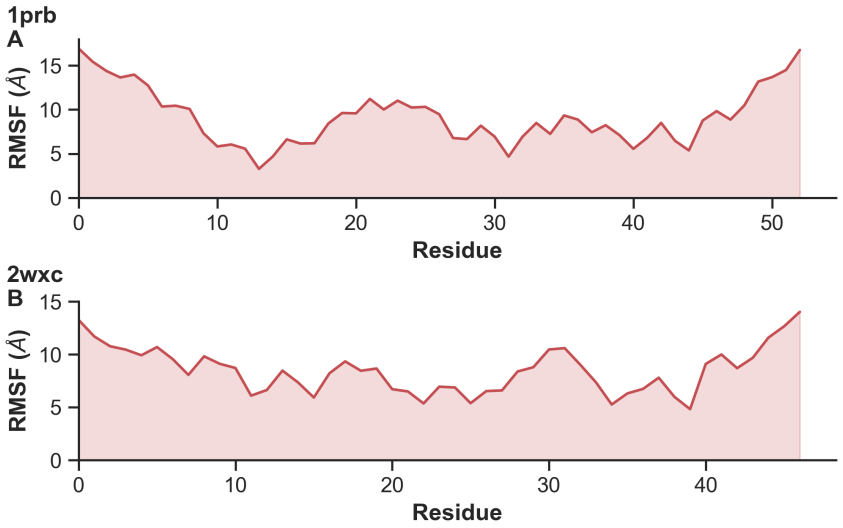

Figure 6.3. Flexibility of protein segments measured by root-mean-square fluctuation (RMSF). **(A)** Albumin-binding domain (PDB id: 1prb[209]). **(B)** BBL (PDB id: 2wxc[208]).

6.2 Achieved Model Accuracy

As previously mentioned, my intention is to select a few representative structures out of the entire REX simulation. However, it is necessary to first assess the similarity of generated REX structures to their native fold. Note that the best achieved model accuracy during REX does technically correspond to the upper limit of each ensemble selection. Table 6.2 summarizes the highest-quality structures that were achieved for each test protein. All REX simulations of this study were capable to generate highly native-like structures. GDTs reached two times values around 80, two times around 90 and one time a nearly perfect score of 97. Surprisingly, the best results were achieved for the largest test protein, i.e. the Lambda repressor with a length of 92 residues. Its best achieved structure model measures a GDT score of 97, which corresponds to a backbone RMSD of approximately 1.0 Å. The high similarity with the native fold might be resulting from the high count of α-helices within the structure, since helices tend to be very stabilizing during MD. The worst results were achieved for Albumin-binding domain and BBL, with GDTS up to 78 and 81, respectively. I want to emphasize that in both cases the "relatively low" GDT scores yield from highly flexible tail segments, as shown in Fig. 6.3. If they are ignored during the GDT calculation then the new GDT scores reach values of approx. 90. But I kept using the entire protein model in order to maintain a global measurement, as intended by the global distance test.

Table 6.2. Best achieved model accuracy during study using contact-guided REX MD[3]. Table contains the protein's PDB id, the occurring secondary structure (ss) motifs, the system size (approx. atom count), the global distance test total score (GDT TS), and the backbone root-mean-square-deviation (RMSD) relative to the known protein structure.

PDB id	ss motifs	system size	GDT TS	RMSD (Å)
1lmb	α	$54 \cdot 10^3$	97	1.0
1prb	α	$47 \cdot 10^3$	78	1.9
2f21	β	$36 \cdot 10^3$	89	1.8
2hba	α, β	$41 \cdot 10^3$	88	1.8
2wxc	α	$34 \cdot 10^3$	81	1.9

6.3 Method Introduction

All structures generated during REX were considered as valid candidates for the ensemble selection, which should include only highly native-like structures. Starting with the entire REX trajectory of the lowest-temperature replica, the first important step is to reduce the data set by pre-selecting structures using a meaningful quantity. Staying close to the origin of my data set I decided to use Q_{Bias}, i.e. the fraction of realized bias contacts in a structural model, as a metric to filter the REX structures. Mathematically, this is described by

$$Q_{\mathrm{Bias}}(t) = \frac{N_{\mathrm{Bias}}(r(t) \leq r_{\mathrm{nc}})}{N_{\mathrm{Bias}}} \in [0,1], \tag{6.1}$$

with the number of integrated bias contacts N_{Bias} and the native contact distance $r_{\mathrm{nc}} = 6\,\text{Å}$. Note that Q_{Bias} does not differentiate between true-positive or false-positive bias contacts. It may even be impossible to achieve $Q_{\mathrm{Bias}} = 1$, especially if this is structurally impossible for example due to competing bias contacts. Typically observed correlations between Q_{Bias} and GDT are highly positive making it a good measurement for the intended structure reduction, as exemplarily shown by Fig. 6.4A. The REX simulation with WW domain (c.f. appendix Fig. E.8A.) was the only exception, in which case no correlation was observed. I decided to pre-select exactly 2000 structures with the highest Q_{Bias} values, which corresponds to 4% of the entire REX trajectory composed of 50000 structures. These pre-selected structures are highlighted in blue in the corresponding scatter plots such as Fig. 6.4A. Next, I calculated the C_α distance matrices of the pre-selected structures which are intended for a structural representation. More precisely, these matrices are used as input for the following dimension reduction (cf. yellow section of Fig. 6.2) which projects the LxL matrices onto (X,Y) coordinates indicating the different structures. This dimension reduction step also marks the first possible deviation between the four investigated ensemble-selection methods.

During this study I investigated two different methods for dimension reduction: t-distributed stochastic neighbor embedding [137] (TSNE) and multidimensional scaling [149] (MDS). Using either of these methods, it is possible to visualize all 2000 pre-selected structures and comprehend their structural (dis)similarities, as exemplified in Figs. 6.5A or 6.6A. It is important to understand that locally adjacent points correspond to structures of high similarity. However based on the applied dimension reduction algorithm the distance interpretation is slightly different. For example, "TSNE can visualize small structural differences better by creating many separated point clusters due to the t-distributed push-pull moves of samples during the algorithm. In other words, this algorithm aims to separate different structures from each other. MDS on the other hand visualizes structural differences better. That is because the distance between MDS points is always proportional to the difference of the corresponding distance matrices." [3] The projection onto lower dimensions puts a focus on structural differences and illustrates them human-readable. This allows me to directly compare the final selections that result from the different pipelines and to evaluate each method's performance. Additionally, by using less dimensions in the upcoming clustering step, the entire algorithm chain becomes much more robust as it reduces deviations between separate clustering executions (mainly relevant for KMEANS).

I also investigated two different clustering algorithms: KMEANS [163] and DBSCAN [167,168] (density-based spatial clustering of applications with noise). Note that KMEANS, which is the most famous clustering algorithm, has multiple variations that differ, e.g., in their initialization process or the used objective function to assign cluster labels [164,175]. In my work I only applied Lloyd's KMEANS algorithm with k-means++ initialization [178,179]. This method requires two important parameter specifications: 1) the number of cluster centers k and 2) the number of KMEANS runs with independent initializations n_{init}.

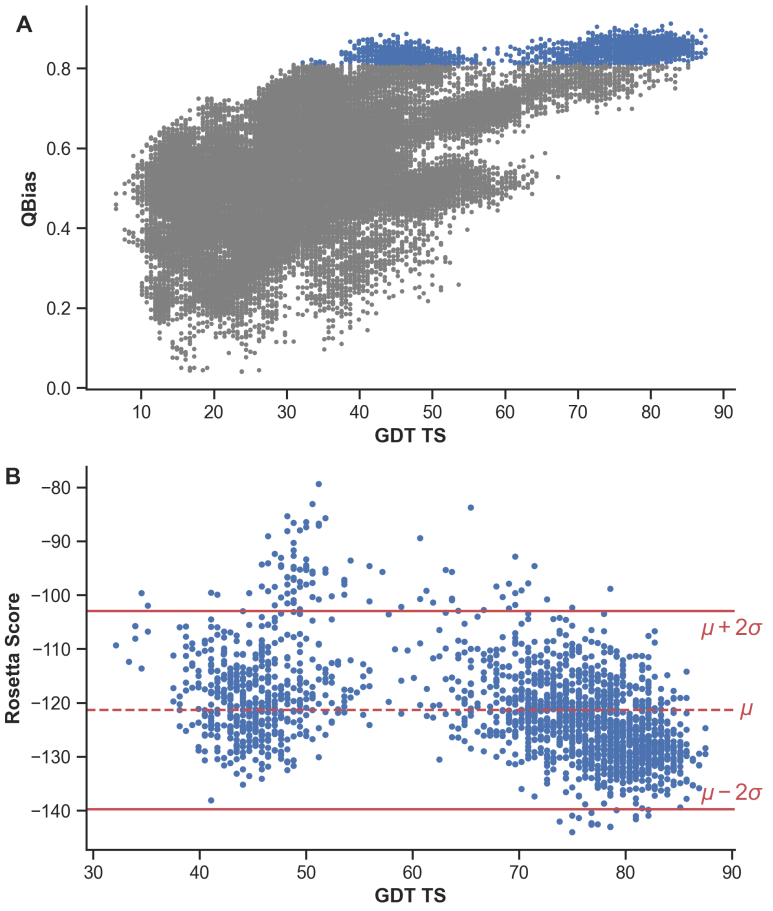

Figure 6.4. Correlations of NTL9 (PDB id: 2hba[211]) simulation. (A) Relation between Q_{Bias} (fraction of realized bias contacts) and GDT TS. Gray and blue colored dots represent the entire REX MD trajectory composed of 50000 structures. Blue dots highlight the 2000 structures with the highest Q_{Bias} values, which were pre-selected for the ensemble selection. **(B)** Relation between Rosetta score and GDT TS of the 2000 pre-selected structures. Figure also depicts the mean score μ (red dashed line) and $\mu \pm 2\sigma$ (red solid lines) which were used as thresholds to filter outliers during the cluster score calculations. Reproduced from Ref.[3] under CC BY 4.0.

"In each run, initial cluster centers are randomly chosen from the data set and data points are assigned to the nearest cluster center. Next, cluster centers are shifted to the mean of all points belonging to a cluster and previous steps are repeated until convergence where no further changes occur. After n_{init} independent runs, KMEANS selects the best result based on the smallest sum of cluster variances."[3] Due to the random selection of initial cluster centers independent executions do typically not produce identical clustering results. The difference primarily depends on the underlying data, or more precisely on the density of data points. A large amount of randomness can be eliminated, however, if a dimension reduction is performed prior to the clustering. The combination of these techniques make the whole ensemble-selection method in general more robust and reliable. The other investigated clustering method, DBSCAN, is density-based and can automatically differentiate between cluster points and noise. This algorithm also requires two important parameter specifications: 1) the neighborhood distance ε and 2) the core density min_{pts}. "ε describes the maximum distance between two samples, which are considered in the neighborhood. min_{pts} specifies how many data points within ε around sample X are required to consider X as a core sample and part of a cluster. If a core sample is identified, the cluster grows by including points within the ε neighborhood, which can also be core points or just simply reachable

neighbors. Lastly, all points which are not within the neighborhood of core points are considered noise." [3] This density-based algorithm makes DBSCAN deterministic, meaning that individual executions yield the same results for the same parameters. For comparison reasons, I decided to keep the total cluster count fixed based on the applied clustering method. Hence all pipelines using KMEANS generated exactly $k = 30$ clusters, whereas DBSCAN parameters were tweaked until I obtained exactly 21 clusters (i.e. 20 cluster labels and 1 noise label).

After the clustering process is over it is finally time to select the wanted target structures. Since my goal is to select highly native-like structures, I aim to maximize the GDT. My selection criterion is based on Rosetta scores because this energy function is always accessible and requires solely the atom positions and their type for its calculation. After obtaining the Rosetta scores of the 2000 pre-selected structures I map them onto the corresponding TSNE/MDS representations. These can now be interpreted as an energy landscape of protein structures where native-like folds should be located in the deepest valleys. I want to emphasize that Rosetta scores alone are usually not accurate enough to discriminate native from native-like or even non-native folds. This was already shown during the decoy analyses in section 5.2 (cf. Figs. 5.4+5.5) and can also be seen here. As exemplified in Fig. 6.4, the 2000 pre-selected structures of NTL9 have low Rosetta scores for structures with GDT scores primarily between 70-90 but also between 40-50. Similar scatter plots for the other proteins can be found in appendix E. Performing a selection only based on Rosetta scores is therefore not ideal, as it cannot guarantee high-quality structures. But similar to the observations in section 5.2, correlations of Rosetta score vs. GDT are always negative. Hence, low Rosetta scores are statistically favored to represent high GDT structures. For this reason, instead of making a selection based on individual structures, it is more reliable to select a bunch of similar structures and take their collective Rosetta scores into account. In my case, this means that I can obtain high-quality structures by identifying four clusters with the lowest mean scores and pick them in increasing order.

Finally, the comparison and evaluation regarding the performance of the four presented algorithm chains can be achieved by looking at the cluster accuracy statistics. Analogously to the energy landscape, I am able to assign GDT scores to the TSNE/MDS representations and obtain a landscape indicating the structure's refinement levels. To assess each cluster's accuracy I assigned each cluster their corresponding mean GDT value. Afterwards I relabeled the clusters based on their GDT ranking which makes the interpretation of the selected clusters much easier and allows to quickly assess each method's performance. The valid cluster indices are 0-29 (0: best, 29: worst) for KMEANS or 0-20 (0: best, 19: worst, 20: noise) for DBSCAN. Please note that the calculation of GDT scores as well as the ranking of cluster labels are only possible as I have access to the experimentally determined native structures of the used proteins. However, this information is only used for evaluation purposes and is not required to select the structure ensembles themselves.

6.4 Method Comparison

From now on each time cluster labels are mentioned they have been already sorted according to their GDT accuracy. Tables 6.3 and 6.4 give a performance overview by listing the selected clusters of each algorithm chain using KMEANS or DBSCAN, respectively. "Additionally, the tables state a performance rating, which indicates the importance of selected clusters. The rating is given by the weighted sum of selected clusters. However, meaningful weights are only assigned to clusters with labels 0-3 representing the highest GDT ensembles. Mathematically, this is provided by

$$rating = \sum_i w_i(cluster), \tag{6.2}$$

with the weights $w_0 = 4, w_1 = 3, w_2 = 2, w_3 = 1, w_{i>3} = 0$.

In general, all compared algorithm chains yield very good results regarding the final ensemble selections. Note that it was always possible to select the two highest GDT ensembles (labels 0 and 1) and in some cases even up to the four highest GDT ensembles. Algorithms using TSNE for dimension reduction were exceptional stable and produced ratings with 9/10 or higher for TNSE → KMEANS and 8/10 or higher for TNSE → DBSCAN. The direct rating comparison slightly favors the TNSE → KMEANS algorithm. Additionally, this procedure is very straight forward and does not require any case-specific parameter tuning, as compared to the TSNE → DBSCAN pipeline. The selected ensembles resulting from algorithms using MDS are promising as well. MDS → KMEANS tends to produce ratings with approximately 9/10. However, the test case with NTL9 (PDB id: 2hba) yielded a rating of only 7/10. MDS → DBSCAN pipelines produced ratings between 8/10 and 10/10."[3] Please refer to appendix tables E.4 and E.5 for a detailed listing of the selected clusters and their accuracy.

Table 6.3. Performance of algorithm chains with KMEANS clustering[3]. Selection order of clusters is based on mean Rosetta scores, while cluster labels are ranked by accuracy (mean GDT scores, 0: best cluster). Rating is calculated according to Eq. 6.2 and indicates the importance of selected clusters by weighing only clusters 0-3.

	TSNE → KMEANS		**MDS → KMEANS**	
PDB id	selected clusters	rating	selected clusters	rating
1lmb	1-0-6-2	9/10	2-0-1-6	9/10
1prb	2-0-1-3	10/10	0-2-1-10	9/10
2f21	1-0-2-3	10/10	1-0-3-2	10/10
2hba	3-2-0-1	10/10	0-1-7-8	7/10
2wxc	5-0-2-1	9/10	1-10-0-2	9/10

Table 6.4. Performance of algorithm chains with DBSCAN clustering[3]. Selection order of clusters is based on mean Rosetta scores, while cluster labels are ranked by accuracy (mean GDT scores, 0: best cluster). Rating is calculated according to Eq. 6.2 and indicates the importance of selected clusters by weighing only clusters 0-3.

	TSNE → DBSCAN		**MDS → DBSCAN**	
PDB id	selected clusters	rating	selected clusters	rating
1lmb	1-0-2-4	9/10	2-0-3-1	10/10
1prb	1-0-2-14	9/10	2-0-1-12	9/10
2f21	0-1-6-2	9/10	0-1-8-2	9/10
2hba	3-0-1-7	8/10	1-4-0-3	8/10
2wxc	1-0-3-9	8/10	2-3-0-1	10/10

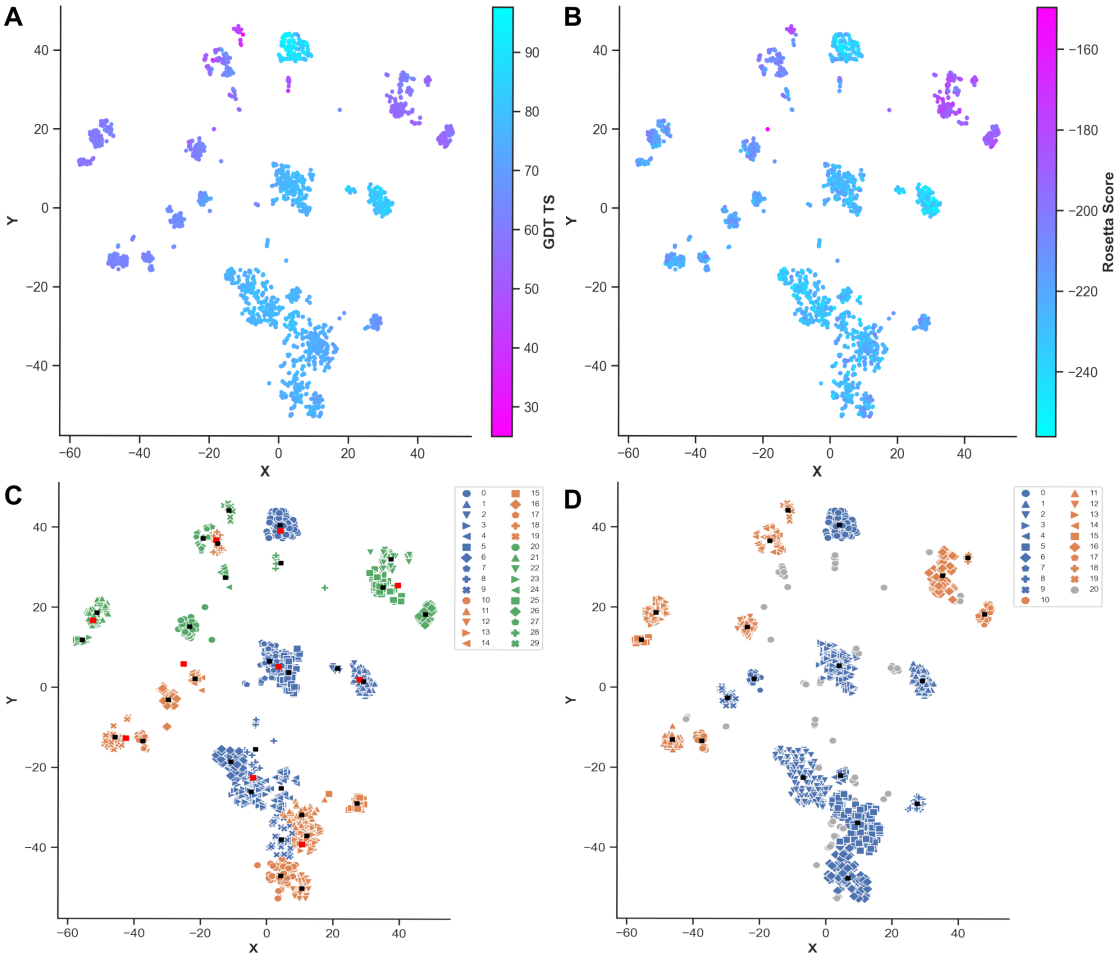

Figure 6.5. TSNE representation of selected Lambda repressor (PDB id: 1lmb[210]) structures. **(A)** Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 2.55$, $min_{\text{pts}} = 20$). Adapted from Ref.[3] under CC BY 4.0.

Next I want to discuss the different ensemble-selection methods in great detail, pointing out their individual strengths and weaknesses. By doing so I will highlight specific aspects that should be considered when evaluating each method's performance. The rating, which was introduced in Eq. 6.2, indicates the achieved performance of each algorithm pipeline. Nonetheless, to fully understand the differences between the selection methods, it is important to investigate the location of selected clusters or why exactly they were selected. For example, Fig. 6.5 illustrates many different relations which can be inferred by comparing the respective energy, GDT, or cluster mappings. In this case, Fig. 6.5A shows the TSNE representation of the 2000 pre-selected Lambda repressor structures and displays their structure quality measured by GDT. It can be interpreted as the *ground truth* in terms of the similarity of individual structures compared to the native fold, which is not always accessible. Fig. 6.5B on the other hand shows the Rosetta score mapping instead, which is always accessible as it requires only the atom position and their type for calculation. A high similarity between Figs. 6.5A and 6.5B means that the underlying Rosetta score mapping is accurate and can be used to reliably deduce high-quality structures. Note, that GDT scores and Rosetta scores have inverse scaling, i.e. high-quality structures have high GDT scores but statistically low Rosetta scores.

To facilitate the direct comparison I adjusted the applied color-schemes accordingly. I want to point out again that Rosetta scores alone are usually not sufficient to accurately infer the structure quality, as shown in Figs. 6.4A or Figs. E.7A to E.9A. In all these cases, low Rosetta scores are mapped to both low- and high-quality structures. Lastly, Figs. 6.5(C+D) show the final cluster mapping for KMEANS or DBSCAN, respectively. Depicted cluster labels are already sorted by accuracy, where cluster 0 has the highest mean GDT scores and corresponds to the best selection. To fully comprehend each methods performance, it is necessary to compare the Figs. 6.5(A-D) while focusing on the location of the four selected clusters (cf. Tables 6.3 and 6.4). A comparison of all test proteins shows that Lambda repressor has the most accurate energy landscape, which is why Figs. 6.5(A+B) look almost identical. Of course the same similarity is observed for the MDS representation, as shown in Fig. 6.6, due to the fixed structure-to-score mapping while their (X,Y) positions differ based on the dimension-reduced representation. Note that the derived energy landscapes of other test proteins are generally bad at inferring low-quality structures, which can be seen in Figs. E.11(A+B) for albumin-binding domain or Figs. E.15(A+B) for NTL9. In these cases structural regions that contain the lowest GDT scores are not mapped to the lowest Rosetta scores.

"When comparing Figs. 6.5 with 6.6 we can observe the main difference between TSNE and MDS representations. TSNE plots tend to have more distinct sample groups which results from the t-distributed push-pull projection. In both representations, highly similar structures are located very close to each other. However, in TSNE the distance information is not conserved to the same degree as for MDS. This means, that both GDT and Rosetta score landscapes are much easier to understand for MDS, as compared to TSNE. E.g., Fig. 6.6A has exactly one distinct local minimum (left centered region), whereas Fig. 6.5A has multiple local minima spread around. This feature can be utilized to guess bad ensemble selections for algorithms using MDS. For example, if three out of four clusters are close to each other but one is far away during MDS $\rightarrow$ KMEANS, the one isolated cluster has a high probability to be a bad choice due to inaccurate Rosetta score mappings. Although TSNE does not have such a reliable way to tell false-positives, I found a realizable workaround. As seen in Fig. 6.5C, KMEANS cluster centers for $k = 30$ and $k = 10$ are shown additionally as black and red squares, respectively. By clustering all samples with a high and low number of cluster centers, we can probe the associated cluster scores on different scales."[3] By carefully interpreting the positions and distances of all cluster centers, it is possible to obtain some sort of confidence boost and apply this information to deduce an estimated cluster ranking. In my case it was helpful to identify the three nearest $k = 30$ clusters to the lowest scoring $k = 10$ cluster center. Most of the time at least two of these three clusters had cluster labels 0-2 and were part of the best selections based on their associated GDT accuracy.

Another important aspect of the evaluation is the robustness of each investigated algorithm chain. In general, using TSNE for dimension reduction makes the ensemble selection very robust. Application of either KMEANS or DBSCAN on a TSNE representation yield very similar results. Additionally, independent executions of KMEANS reproduce almost identical results. This means that TSNE greatly reduces the variance related to KMEANS' random initialization. MDS on the other hand cannot eliminate the stochasticity of KMEANS, which is mainly observed if the MDS representation contains very dense regions. This is however a typical feature of MDS, since the distance conservation leads to an accumulation of structures into few but dense regions. In some extreme cases structures can even accumulate mainly into a single stack, as exemplarily shown for WW domain in Fig. E.13 or for BBL in Fig. E.17. Here the structural differences are not large enough to indicate a clear separation of structures. When KMEANS is applied on such 2D representations, independent runs do generate slightly varying results which can be observed by minor movement of cluster borders. Consequently this can influence the final ensemble selection which depends on the cluster ranking determined by mean Rosetta scores.

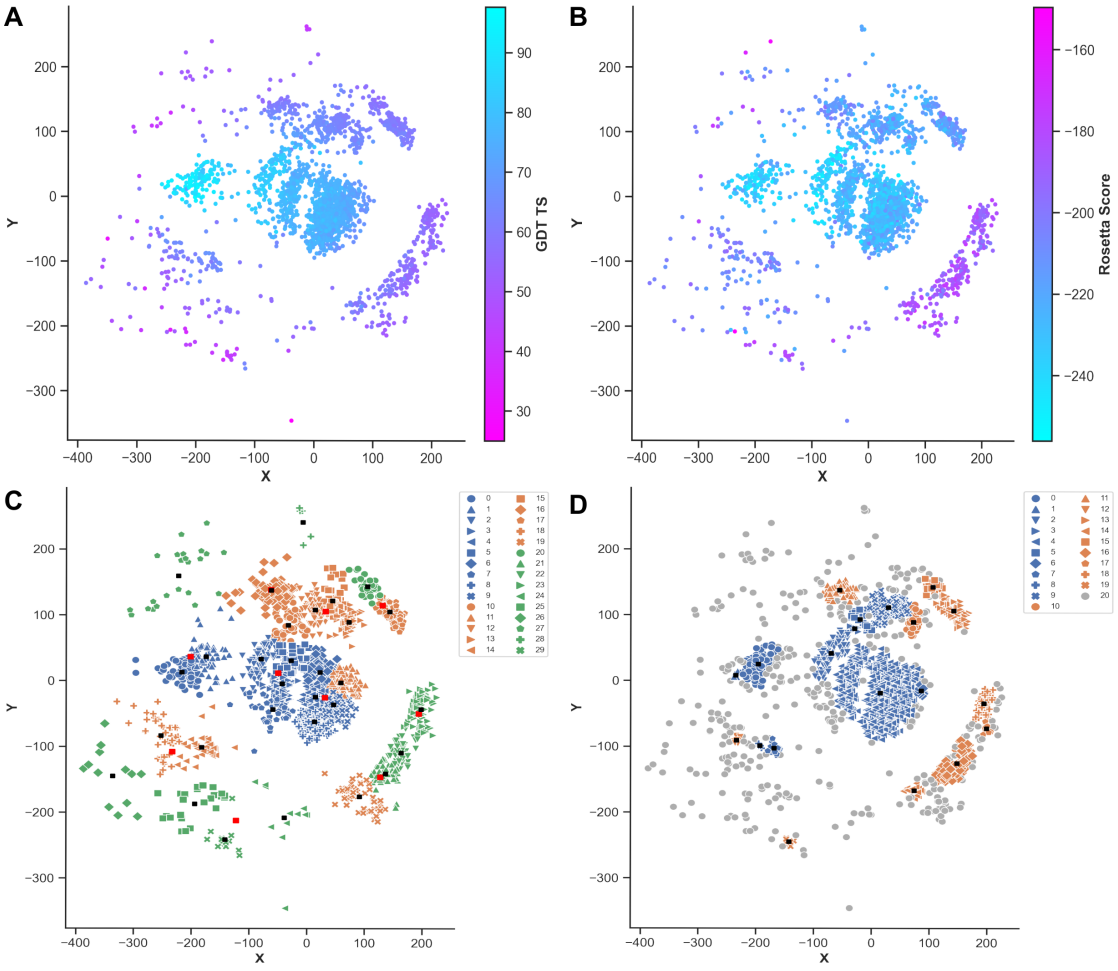

Figure 6.6. MDS representation of selected Lambda repressor (PDB id: 1lmb[210]) structures. **(A)** Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 8.31$, $min_{\text{pts}} = 5$). Adapted from Ref.[3] under CC BY 4.0.

I want to emphasize that these variations can also be observed with different KMEANS initialization methods (e.g. Forgy[175,176]), since it primarily depends on the density of data points. To further analyze the robustness of this pipeline, I performed 100 independent MDS → KMEANS executions and calculated the individual ratings according to Eq.6.2. In the course of this the rating stayed nearly constant and observed changes were in the order of ±1.

Lastly, when comparing the fourth pipeline, namely MDS → DBSCAN, we can observe one negative aspect that stands out. As previously mentioned, this density-based clustering method is deterministic and its outcome depends on two things: 1) the parameter choices of ε and min_{pts} and 2) the 2D representation of the underlying data set. Note that both parameters have a correlation with distance, whereas ε is clearly dominating and affects the results of DBSCAN the most. With this in mind, MDS → DBSCAN becomes very case-specific and can be seen as unreliable, especially because different parameters may lead to very different cluster sizes and also ensemble selections. Negative examples can be given by, e.g., WW domain and BBL as illustrated in Figs. E.13D and E.17D, respectively. However, during this study I forced a fixed cluster count for algorithm chains using DBSCAN, which allowed me to objectively compare the results of different proteins.

To fulfill this condition, I had to manually tweak the parameters until I got exactly 21 clusters, highlighting the case-dependency of DBSCAN. The resulting ensemble selections and the overall performance were comparable to the other investigated alternatives. The biggest difference compared to other pipelines is that MDS → DBSCAN manages to select very small ensembles with extremely high structure accuracy. For example, the final selection for Lambda repressor contained an ensemble with only 5 structures and a maximum GDT of 94.64 (cf. Table E.4). Similar to this, the final selection for NTL9 yielded an ensemble with 8 structures and a maximum GDT of 85.71 (cf. Table E.5).

6.5 Summary

As shown at the beginning of this chapter, contact-guided REX MD is capable to achieve relatively good structure refinement for medium-sized proteins up to approximately 90 residues. Generated structures reached GDT values above 80 in all 500 ns long simulations. In case of WW domain and NTL9 it was possible to obtain highly native-like structures with GDTs close to 90. However the best structure accuracy was observed for Lambda repressor, which was the largest test protein with a size of 92 residues. Its best model had an outstanding GDT score of 97 and was acquired after only 250 ns using REX. The observed performance and best-achieved model precision was more dependent on the true-positive rate of the bias contacts as compared to the secondary structure motifs or size of the protein. High-quality structures of α-helical proteins were obtained even after very short simulation times. Larger proteins or β-sheet proteins generally required more time before achieving good results. Based on the size and complexity of the used protein, REX trajectories above 1 μs might be required to reach the best-possible refinement results due to additional replica turnarounds.

My main objective of this study was to find a robust and reliable solution to select a few representative structures out of the large pool generated by REX. In this case, I targeted to reproduce the native state and validated the selection method by mimicking a *blind-prediction scenario* with unknown target structure. In general, such a task is very challenging. There exist many different measurements or scoring formulas, which can be used to assess the quality of a protein structure. However, each on their own is typically not sufficient enough to guarantee outstanding structure selections. This was exemplarily shown for the Rosetta scoring function, which was applied to estimate the structure's quality. However, this energy function could not reliably differentiate between individual low- and high-quality structures as shown for four out of five proteins. Nevertheless, the underlying correlation of Rosetta score vs. GDT was always negative. Utilizing this statistic feature to my advantage, I showed that I can successfully obtain highly native-like structures when executing a specific order of algorithms. I introduced the design of the applied ensemble-selection method and explained the four different algorithm pipelines. In the course of my study, I investigated each pipeline in great detail and objectively compared their performance.

Starting with the structures taken from the simulation, each chain requires a pre-selection of trajectory frames to reduce the frame count to a manageable amount. I showed that the fraction of realized bias contacts Q_{Bias} is a suitable quantity to reduce the frame count due to the primarily positive correlation with GDT scores. The next important step is the dimension reduction of pre-selected structures and their C_α distance matrices. The projection of structural features onto lower dimensions improves the overall robustness of the algorithm chain and minimizes cluster-related stochasticity. It also provides human-readable 2D representations of structures which can be extended via Rosetta and GDT score mappings to evaluate the algorithm performance. I compared two variants of dimension-reduction (TSNE vs. MDS) and two variants of clustering (KMEANS vs. DBSCAN).

The four possible algorithm chains were

1) TSNE → KMEANS,
2) MDS → KMEANS,
3) TSNE → DBSCAN,
3) MDS → DBSCAN.

I showed that I can successfully extract high-quality structure when taking many similar structures and their collective Rosetta scores into account. Hence I calculated mean Rosetta scores for each cluster and selected the four lowest-scoring clusters as final picks. Note that while REX leads to thermodynamically correct ensembles, the clustering does not maintain this property. As I explicitly used proteins with an already determined native structure, I was able to evaluate the performance of each algorithm chain by comparing the selected ensembles with their corresponding cluster accuracy. For this purpose, I introduced a numerical rating which makes it easier to compare the performance by weighing only the four ensembles corresponding to the most-refined structures. Lastly, I compared each algorithm chain in great detail by talking about differences of TNSE/MDS representations and highlighting relevant pipeline features. A comparison of the different methods including their pros and cons given by Table 6.5. On the other hand, Fig. 6.7 gives an overview of selected representative structures.

The presented algorithmic workflows performed very well in all test cases. I want to emphasize that I was always able to obtain the two most native-like structure ensembles (i.e. clusters with label 0 and 1). However, it is still not possible to perfectly rank the selected ensembles based on accuracy if the target structure is truly unknown. The final ensemble selections primarily depend on the accuracy of the underlying energy function. As shown for four of the five test proteins, Rosetta scores alone are not accurate enough to reliably distinguish between low and high GDT conformations. Still, it is possible to apply small tricks to differentiate between particular good and bad picks. For instance, it is possible to use the distance preservation of MDS to eliminate bad picks, if one of the selections is located far away from the others. Another example was given by performing KMEANS on two separate scales, e.g. with $k = 10$ and $k = 30$. By changing the number of cluster centers k, comparing of individual distances of cluster centers and their energy rankings, one can deduce the real accuracy ranking in some cases.

Although I exemplarily aimed for native-like ensemble selections during this study, the key aspects of this methodology should be applicable to other ensemble-selection objectives as well. This would require an alteration of only two steps, namely the pre-selection (here: filter structures by Q_{Bias}) and the scoring function during the final ensemble selection (here: mean Rosetta scores of clusters). By doing so, one should achieve similar results for other ensemble targets. For example, the application of an energy function that favors β-sheets could detect and select structure ensembles with high amounts of β-sheets.

Table 6.5. Comparison overview of the four investigated algorithm chains[3]. Clusters were selected by calculating mean Rosetta scores and picking the four lowest-scoring clusters. The total cluster count varies based on the used clustering method (KMEANS or DBSCAN). Average rating was calculated using Eq. 6.2 and normalized across all five test proteins.

algorithm chain	cluster selection	average rating	positive features	negative features
TSNE → KMEANS	top 4/30	9.6/10	straight forward / no parameter tuning	selection can include noise data
MDS → KMEANS	top 4/30	8.8/10	distance preservation allows to guess false-positives	dense sample regions increase randomness of cluster borders
TSNE → DBSCAN	top 4/21	8.6/10	reduced noise	slightly parameter dependent
MDS → DBSCAN	top 4/21	9.2/10	possible to identify small ensembles with extremely high structure accuracy	DBSCAN parameters correlate with distance, i.e. heavily depend on case-specific MDS representation

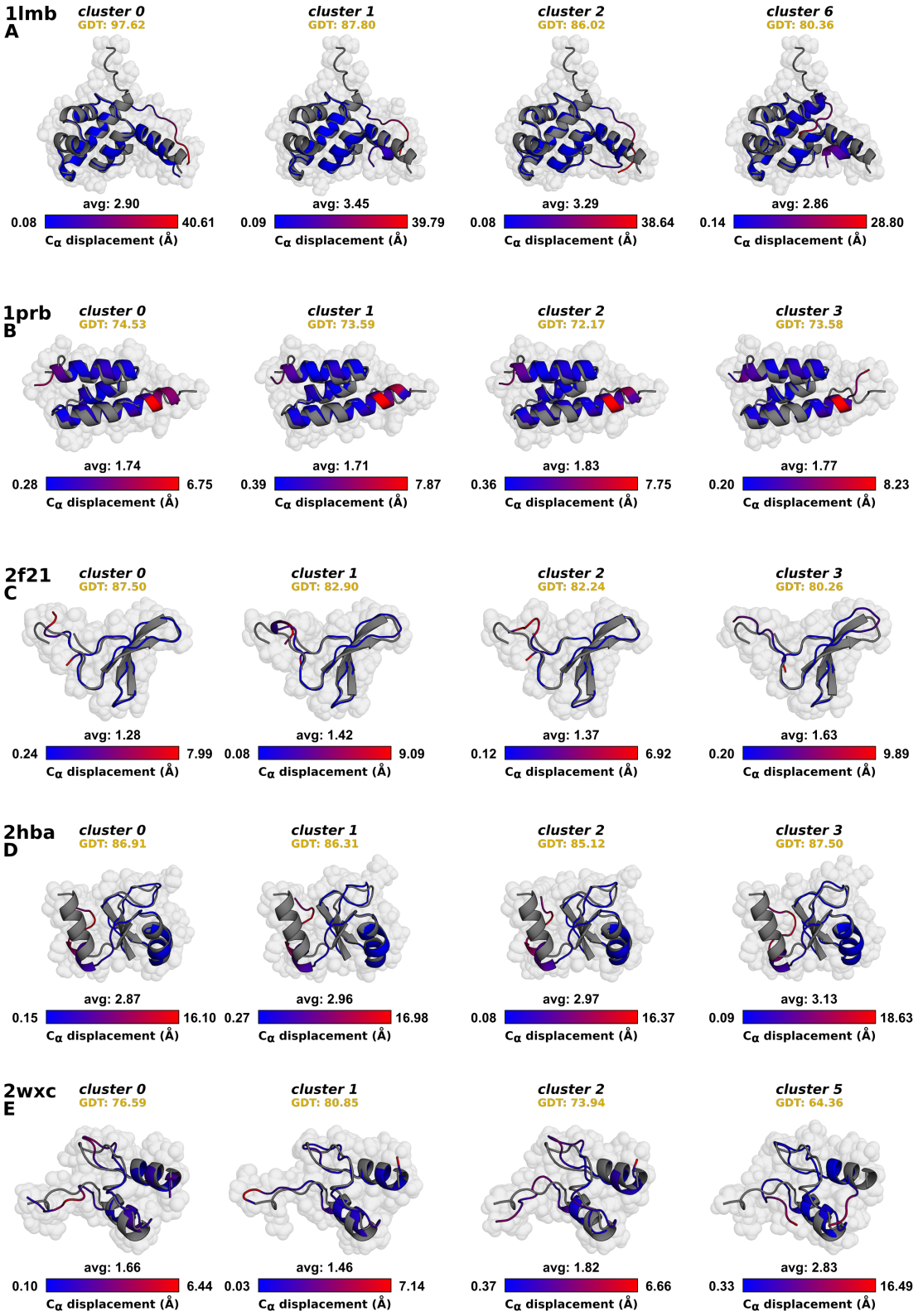

Figure 6.7. Representative structures obtained via TNSE → KMEANS algorithm chain. Best structures of each selected cluster. Cluster labels are ranked according to GDT mean statistics. Color-coding highlights the C_α displacement with respect to the reference structure. **(A)** Lambda repressor (PDB id: 1lmb[210]). **(B)** Albumin-binding domain (PDB id: 1prb[209]). **(C)** WW domain of human Pin1 Fip mutant (PDB id: 2f21[212]). **(D)** N-terminal of L9 protein (PDB id: 2hba[211]). **(E)** BBL (PDB id: 2wxc[208]). Visualized in PyMol[189,190].

7

pyrexMD: Workflow-Orientated Python Package

This chapter covers `pyrexMD`*, which is the Python package I developed during the course of my work. Section 7.1 outlines both motivation and intention of the software. By combining the experience and insights from my conducted REX studies, I provide a solution to automate and facilitate (contact-guided) REX MD.* `pyrexMD` *offers an interactive "all-purpose" environment and combines critical aspects of each biomolecular study, i.e. design, simulation, analysis and visualization. Section 7.2 provides a brief overview of* `pyrexMD's` *functionality based on the module architecture. Finally, section 7.3 acts as a quick guide to* `pyrexMD` *and contains multiple short code examples for different use-cases. Some parts of this chapter are reproduced from my article called "pyrexMD: Workflow-Orientated Python Package" (2021)[2], published by the Journal of Open Source Software.*

7.1 Motivation

Molecular dynamics (MD) is already a well-established method for *in-silico* studies of biomolecular systems. There exist various software solutions which are specialized for specific tasks, such as:

- simulation software: `GROMACS`[107–109], `NAMD`[214,215], `LAMMPS`[216], etc.
- molecule/trajectory viewer: `PyMOL`[189,190], `VMD`[217], `Chimera`[218,219], etc.
- data analysis and visualization: `MDAnalysis`[220,221], `MDTraj`[222], etc.

Researchers often have to select and mix different software solutions to conduct their work. Initially, it requires an intensive time investment just to learn the unique cmd-line syntaxes, get familiar with provided functions, etc. Besides, each software uses input and output files with specific data formats, such that mixing of different software can become quite arduous when setting up a newly developed workflow.

My own studies primarily utilize the replica exchange (REX) method during MD simulations to generate physically-meaningful structure ensembles. Each REX simulation can generate terabytes of data and thus requires a well-organized file system and streamlined workflow operations to successfully analyze and visualize the simulated data. Additionally, each REX setup consists of various small and system-specific tasks, since I apply contact-based bias potentials to guide my simulated systems towards specific conformations. Manually performing these steps is not only very time consuming but also a source for errors, which could drastically change the outcome of any REX study and should therefore be eliminated.

For this purpose, I decided to develop a Python package named `pyrexMD`[2] to automate many REX-related tasks and make the setup of (contact-guided) REX MD simulations as easy as possible. Gained insights from my conducted studies are integrated in provided functions and used to optimize workflows or protocols. Additionally, I wanted to create an interactive "*all-purpose*" environment, which combines the three critical aspects of each biomolecular study: simulation, analysis and visualization.

`pyrexMD` efficiently integrates and extends the following popular MD-related Python packages:

- `GromacsWrapper`[223]: used for interactions with `GROMACS` to setup and run MD simulations
- `MDAnalysis`[220,221]: used to parse MD-related data and provides most-basic analysis functions
- `nglview`[224]: used as trajectory viewer.

My software package was build around the use of `GROMACS`, one of the most popular and efficient MD simulation software packages available. This open-source solution provides many different force fields such as AMBER[105], CHARMM[106], GROMOS[225], or OPLS[226]. The core functionality of `GROMACS` can be further extended with plug-ins such as `PLUMED`[227,228] or `SSAGES`[229]. These plug-ins implement additional algorithms and enhanced-sampling methods which interact during the MD simulation itself or can give access to user-defined collective variables for new types of analyses.

"`pyrexMD` on the other hand focuses on facilitating, assisting, and automating the simulation setup and post-simulation analyses. It provides efficient and robust methods for setting up optimized (contact-guided) REX MD or MD simulations. Furthermore, it offers many intuitive and user-friendly structure analyses and comparison functions to explore the large I/O sets generated by REX."[2]

During the software development, I focused on providing functions with a consistent but easy-to-use syntax. Note that some functions offer interactive features which can only be utilized when used in `jupyter` notebooks[230,231]. The application within `jupyter` provides even more benefits. Most notably, it allows the user to execute `pyrexMD` remotely via ssh on any browser to utilize other hardware systems providing more resources and storage. For example, it is possible to access an office computer (or even HPC) with better specs and more local storage as compared to a home office PC/laptop. Results can be inspected and new analyses started, while the software runs with full speed on the remote machine.

7.2 Package Overview

"`pyrexMD` is a self-developed Python package that is mainly designed for research projects which:

- use (contact-guided) Replica Exchange Molecular Dynamics or (contact-guided) Molecular Dynamics
- or focus on structure analyses and comparison.

It has three main goals:

1) Interactive '*all-purpose*' environment:
By including various modified `GROMACS` and `MDAnalysis` Python bindings, this package provides a comprehensive `Jupyter` notebooks based environment to design, run, and analyze entire MD projects.

2) Data visualization is important:
Most analysis functions for calculating useful quantities, such as RMSD, Q values, contact distances, etc., can generate specialized figures in the same step by passing the keyword argument $plot = True$.

3) User-friendly and simple application:
Where possible, the provided functions combine individual steps into comprehensive workflows with additional automation features. It is possible to rapidly create whole setup or structure-analysis workflows within a few commands, thereby significantly enhancing productivity and reducing the time spent on various stages of the project.

`pyrexMD` makes it straightforward to create, share, and reproduce research results or transfer the work to other biomolecular structures of interest. Furthermore, it lowers the technical barrier for non-specialists who want to use Replica Exchange for enhanced sampling. `pyrexMD` should be used with `Jupyter` notebooks and requires `GROMACS` to run MD simulations."[2]

Provided functions are combined into modules based on their application purpose. A short overview of the module content is given in Table 7.1. Additional information such as install instructions, quick guide, or detailed API documentations are accessible via `https://kit-mbs.github.io/pyrexMD/`.

Table 7.1. Module overview of `pyrexMD`.

module name	module content
pyrexMD.core	functions enabling interactive analyses. Its main parts are the `iPlayer` and `iPlot` classes, which allow the use of a trajectory viewer or a dynamic linking of the trajectory viewer and any 2D graph.
pyrexMD.gmx	modified `GromacsWrapper` functions for streamlining the interaction with `GROMACS` for system setups etc.
pyrexMD.rex	functions related to (contact-guided) Replica Exchange Molecular Dynamics, mainly for automating and speeding up the simulation setup.
pyrexMD.topology	functions for modifying universe topologies, e.g., align atoms /residues of two universes, get matching selection strings, include bias contacts.
pyrexMD.analysis.analyze	various functions for basic trajectory analyses, e.g., calculating RMSDs, distances, etc.
pyrexMD.analysis.cluster	functions for decoy clustering and post-REX cluster analyses.
pyrexMD.analysis.contacts	functions for native-contact and bias-contact analyses.
pyrexMD.analysis.dihedrals	functions for dihedral-angle analyses.
pyrexMD.analysis.gdt	functions for global distance test (GDT) analyses.
pyrexMD.misc	Consists of pyrexMD.misc.classes, pyrexMD.misc.func, and pyrexMD.misc.plot. This sub-package is a collection of miscellaneous and frequently used functions and classes. These functions may contain modified versions of small existing functions to extend their default behavior in order to streamline `pyrexMD`.

7.3 Application Overview

7.3.1 Setup of Normal MD Simulation

Using GROMACS in pyrexMD is very similar to the known command-line syntax. Commands such as

```
gmx function -p parameter
```

simply become

```
gmx.function(p=parameter)
```

Additionally to the expected GROMACS behavior, each gmx module function creates by default a unique log file with a meaningful name which is stored in the logs folder. The code example below shows a complete setup of a normal MD simulation.

```
import pyrexMD.gmx as gmx
import pyrexMD.misc as misc

# create ref pdb:
pdb = "path/to/pdb"
ref = gmx.get_ref_structure(pdb, ff='amber99sb-ildn', water='tip3p', ignh=True)

# generate topology & box
gmx.pdb2gmx(f=ref, o="protein.gro", ff='amber99sb-ildn', water='tip3p', ignh=True)
gmx.editconf(f="protein.gro", o="box.gro", d=2.0, c=True, bt="cubic")

# copy mdp files (ions.mdp, min.mdp, nvt.mdp, npt.mdp, md.mdp) into working directory
misc.cp("path/to/mdp/files", ".")

# generate solvent & ions
gmx.solvate(cp="box.gro", o="solvent.gro")
gmx.grompp(f="ions.mdp", o="ions.tpr",c="solvent.gro")
gmx.genion(s="ions.tpr", o="ions.gro", neutral=True, input="SOL")

# minimize
gmx.grompp(f="min.mdp", o="min.tpr", c="ions.gro")
gmx.mdrun(deffnm="min")

# NVT equilibration
gmx.grompp(f="nvt.mdp", o="nvt.tpr", c="min.gro", r="min.gro")
gmx.mdrun(deffnm="nvt")

# NPT equilibration
gmx.grompp(f="npt.mdp", o="npt.tpr", c="nvt.gro", r="nvt.gro", t="nvt.cpt")
gmx.mdrun(deffnm="npt")

# MD run
gmx.grompp(f="md.mdp", o="traj.tpr", c="npt.gro", t="npt.cpt")
gmx.mdrun(deffn="traj")
```

7.3.2 Setup of Contact-Guided REX MD Simulation

The code example below shows a complete setup of a contact-guided REX MD simulation using different starting conformations (*decoys*) for each individual replica. It automates many system-specific and arduous tasks to eliminate possible application errors, such as mismatching system sizes across replicas, incorrect mapping of bias contacts, etc.

```
import pyrexMD.misc as misc
import pyrexMD.rex as rex
import pyrexMD.topology as top
decoy_dir = "path/to/decoy/directory"

# create rex_i directories and assign decoys
rex.assign_best_decoys(decoy_dir)
rex_dirs = rex.get_REX_DIRS()

# check for consistent topology
rex.check_REX_PDBS(decoy_dir)

# copy mdp files (ions.mdp, min.mdp, nvt.mdp, npt.mdp, rex.mdp) into working directory
misc.cp("path/to/mdp/files", ".")

# get parameters for fixed box size and solvent molecules
boxsize, maxsol = rex.WF_get_system_parameters(wdir="./rex_0_get_system_parameters/")

# create systems for each replica and minimize them
rex.WF_REX_setup(rex_dirs=rex_dirs, boxsize=boxsize, maxsol=maxsol)
rex.WF_REX_setup_energy_minimization(rex_dirs=rex_dirs, verbose=False)

# add bias contacts (RES pairs defined in DCA_fin)
top.DCA_res2atom_mapping(ref_pdb=<ref_pdb>, DCA_fin=<file_path>, n_DCA=50, usecols=(0,1))
top.DCA_modify_topology(top_fin="topol.top", DCA_used_fin=<file_path>,
                        k=10, save_as="topol_mod.top")

# prepare temperature distribution
rex.prep_REX_temps(T_0=280, n_REX=len(rex_dirs), k=0.006)

# create mdp and tpr files
rex.prep_REX_mdp(main_dir="./", n_REX=len(rex_dirs))
rex.prep_REX_tpr(main_dir="./", n_REX=len(rex_dirs))
# next: upload REX MD run files on HPC and execute production run
```

7.3.3 Interactive Plots

`pyrexMD` can generate interactive plots by linking a 2D graph to the trajectory viewer of a specific universe. It allows to quickly inspect conformations at specific values by interacting with the graph itself (e.g. via ctrl-click). In this way, additional valuable information becomes accessible through the trajectory viewer, as shown in Fig. 7.1. All typical interactions, such as rotation, translation, changing of molecule representation or inspection of atom names and distances, are possible.

```
import MDAnalysis as mda
import pyrexMD.misc as misc
import pyrexMD.core as core
import pyrexMD.topology as top
import pyrexMD.analysis.analyze as ana

# set up universe
ref = mda.Universe(<pdb_file>)
mobile = mda.Universe(<tpr_file>, <xtc_file>)

# calculate RMSD
FRAMES, TIME, RMSD = ana.get_RMSD(mobile, ref=ref, sel1="protein", sel2="protein")

# create interactive plot
IP = core.iPlot(mobile, xdata=TIME, ydata=RMSD, ylabel=r"RMSD (A)")
IP()
```

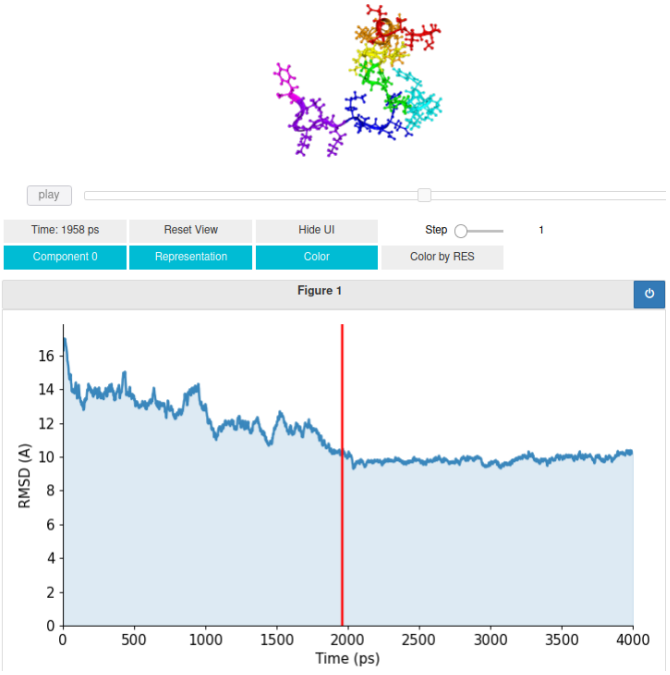

Figure 7.1. `pyrexMD`'s interactive trajectory viewer. Trajectory viewer (top) which is linked to an interactive plot (here RMSD, bottom). Conformations at specific values can be quickly inspected by interacting with the graph itself (e.g. via ctrl-click), thus making additional valuable information accessible through the trajectory viewer. Reproduced from Ref.[2] under CC BY 4.0.

7.3.4 Contact and Bias Analyses

REX is a very powerful and versatile enhanced sampling method. It improves sampling by running many replicas in parallel over a wide temperature range and allows switches of replicas between different temperatures while maintaining thermodynamic ensembles. By integrating (theoretical, experimental, or mixed) bias contacts via bias potentials, one can narrow down the search space and guide the simulations towards specific conformations. This speeds up the process and lowers the computational costs. `pyrexMD` covers many different forms of contact and bias analyses. It distinguishes mainly between two types of Q values, i.e., Q_{Native} (fraction of native contacts) and Q_{Bias} (fraction of realized bias contacts). Both types can be used for structure analyses. However, when simulating unknown target structures Q_{Native} becomes inaccessible due to the missing reference structure.

The code example below exemplarily shows the true-positive rate (TPR) analysis of considered bias contacts for a REX MD study. Predicted bias contacts are initially ranked and then compared with a reference structure. Based on the occurring distances of the residue pairs, contacts are considered true or false using a distance threshold. The code automatically suggests how many bias contacts should be picked for contact-guided REX, using guidelines from one of my studies[1]. E.g., the TPR analysis of Fig. 7.2 suggests to use 25 bias contacts with a TPR of 88% for optimal results.

```
import MDAnalysis as mda
import pyrexMD.topology as top
import pyrexMD.analysis.contacts as con

# set up universe
native = mda.Universe(<pdb_file>)
top.norm_universe(native)

# check True Positive Rate (TPR) of predicted bias contacts
con.plot_DCA_TPR(native, DCA_fin=<path_to_predicted_contacts>, n_DCA=80, d_cutoff=8.0)
```

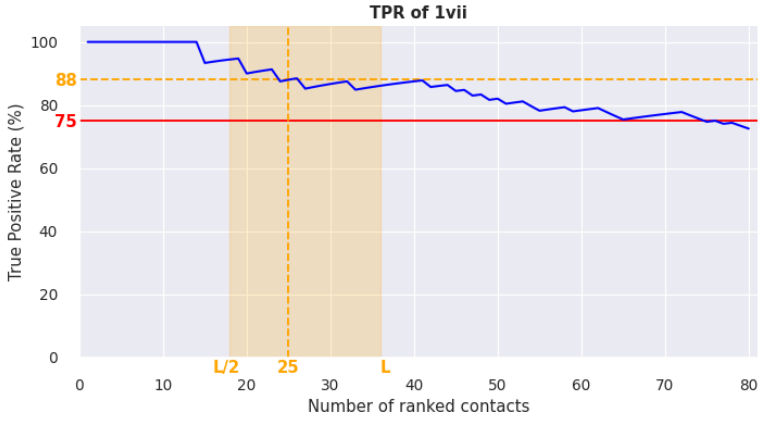

Figure 7.2. True positive rate (TPR) analysis of bias contacts with `pyrexMD`. The figure exemplarily shows the TPR (blue) of the considered bias contacts together with other relevant value guidelines for contact-guided REX MD[1], such as a minimal TPR threshold of 75% (red) and a suggested optimal number of contacts (orange) between L/2 and L, where L denotes the biomolecular sequence length. Here, the plot-function suggests to use 25 bias contacts with a TPR of 88% for optimal results. Reproduced from Ref.[2] under CC BY 4.0.

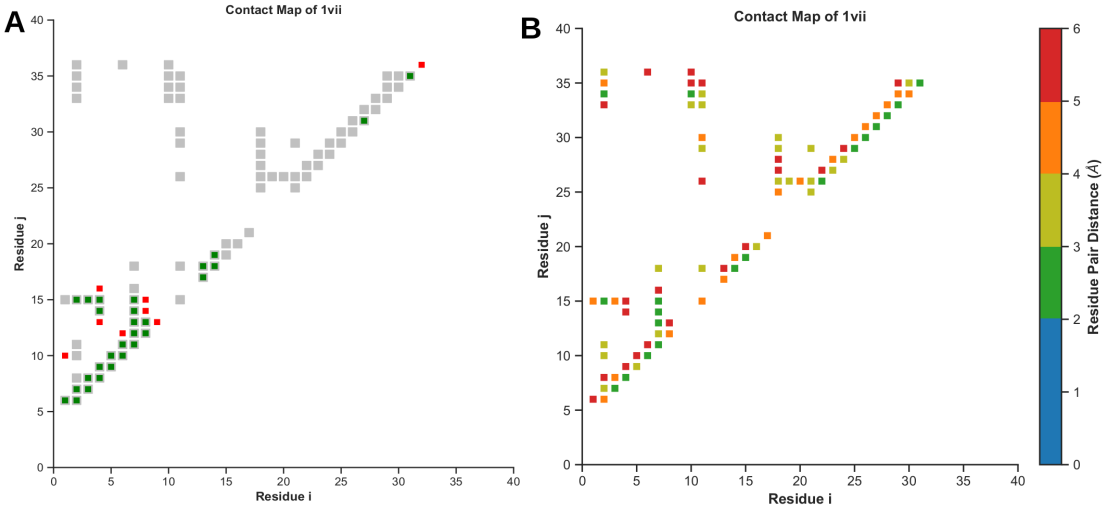

Figure 7.3. Possible contact map analysis variants of `pyrexMD`. (A) Visualization of native contacts (gray) and bias contacts (green: true-positive, red: false-positive). This variant can also be used to compare two different protein configurations and visualize if residue pairs either form new contacts or brake up. **(B)** Visualization of native contact distances to infer how strong and important each contact is with regard to structure stability.

Contact maps are also well suited for contact analyses, as they can be used to display different kinds of valuable information. `pyrexMD` offers two contact map analysis methods. Fig. 7.3A shows the first variant, where the native contacts of a reference structure are shown as gray squares. Additionally, a file containing residue pairs (bias contacts) can be passed, which displays smaller squares color in green or red, in case they are native or non-native contacts, respectively. This enables a very detailed yet compact overview of a protein structure containing all important residue contacts. Alternatively, this contact map variant can also be used to compare two different protein configurations. In this case, it can show which residues form new contacts or brake up. The second contact map variant, as seen in Fig. 7.3B, simply visualizes the native contacts of a protein structure based on their residue pair distance. This information can be used to deduce how strong and important the individual native contacts are with regard to structure stability.

7.3.5 Global Distance Test and Local Alignment Analyses

The so-called global distance test (GDT) [129,130] is a method for structure evaluation similar to the root-mean-square deviation (RMSD). However, RMSD is a sub-optimal measure of structural similarity as it strongly correlates with the largest displacement between mobile and target structure. If the mobile structure globally fits the target to a large extent and only one small segment is misaligned locally, the RMSD becomes disproportionately large. For the GDT, the mobile structure is first aligned to the target structure analogously to an RMSD analysis. To estimate how similar the two structures are, the displacement of each residual C_α atom is calculated and compared to various cutoffs. In a last step, percentages of residues with displacements below a considered threshold are used to calculate scores. The two most common scores are the total score (TS),

$$GDT_{\mathrm{TS}} = \frac{1}{4}\left(P_1 + P_2 + P_4 + P_8\right) \in [0, 100] \tag{7.1}$$

and the high-accuracy (HA) score,

$$GDT_{\mathrm{HA}} = \frac{1}{4}\left(P_{0.5} + P_1 + P_2 + P_4\right) \in [0, 100] \tag{7.2}$$

where P_x denote the percentage of residues with displacements below a distance cutoff of x Å.

While the GDT score can be used to quantify a structure, its value does not contain information about how good each region of a model fits to the reference structure. In such cases, it is better to apply local accuracy (LA) representations, as shown in Fig. 7.4. Such figures display a matrix, where each line corresponds to one structure model and each entry is color-coded based on the C_α-C_α distance between the mobile and reference structure. Therefore, it is possible to quickly check how good the local segments align compared to a reference structure. The code example below shows how generate LA plots (cf. Fig. 7.4) in `pyrexMD`.

```
import MDAnalysis as mda
import pyrexMD.misc as misc
import pyrexMD.core as core
import pyrexMD.topology as top
import pyrexMD.analysis.analyze as ana
import pyrexMD.analysis.gdt as gdt

# set up universes
ref = mda.Universe("<pdb_file>")
mobile = mda.Universe("<tpr_file>", "<xtc_file>")
top.norm_and_align_universe(mobile, ref)

# perform GDT (Global Distance Test)
GDT = gdt.GDT(mobile, ref)
GDT_percent, GDT_resids, GDT_cutoff, RMSD, FRAME = GDT

# calculate GDT scores
GDT_TS = gdt.get_GDT_TS(GDT_percent)
GDT_HA = gdt.get_GDT_HA(GDT_percent)

# rank scores
SCORES = gdt.GDT_rank_scores(GDT_percent, ranking_order="GDT_TS", verbose=False)
GDT_TS_ranked, GDT_HA_ranked, GDT_ndx_ranked = SCORES

# generate plots
ana.PLOT(xdata=frames, ydata=GDT_TS, xlabel="Frame", ylabel="GDT TS")
ana.plot_hist(GDT_TS, n_bins=20, xlabel="GDT TS", ylabel="Counts")

# Local Accuracy plot
gdt.plot_LA(mobile, ref, GDT_TS_ranked, GDT_HA_ranked, GDT_ndx_ranked)
```

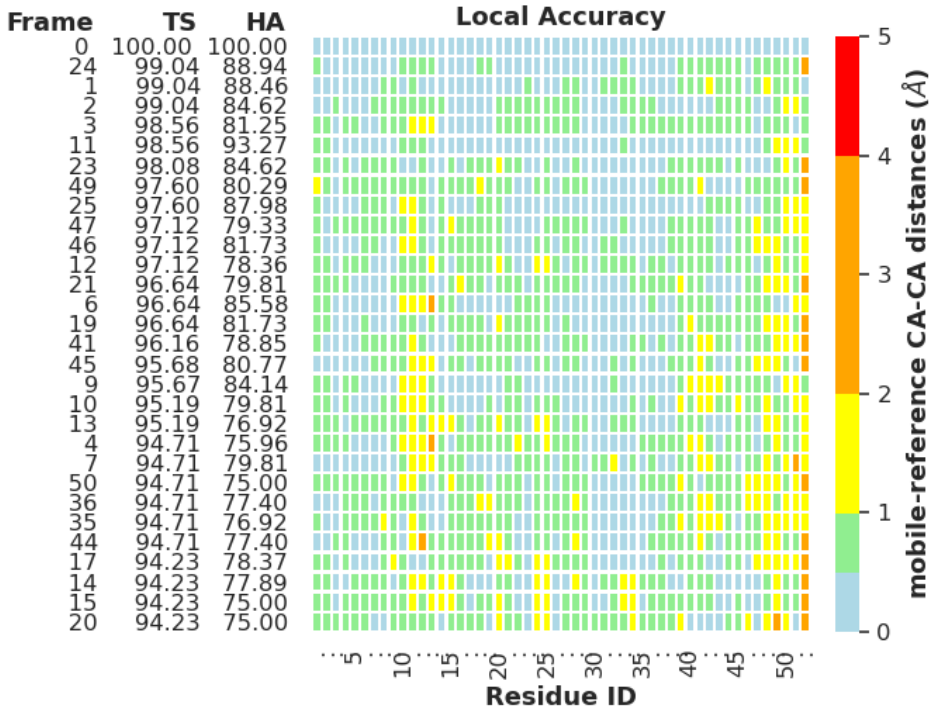

Figure 7.4. Local accuracy analysis of REX-generated protein models using `pyrexMD`. Figure indicates how good each model section is refined compared to a reference structure. Residues are color-coded to represent the CA-CA distance between the model and reference structure after fitting. *Book-keeping information* such as GDT TS, GDT HA, and frame index, are shown on the left side and can be disabled individually. Reproduced from Ref.[2] under CC BY 4.0.

7.3.6 Cluster Analyses

REX MD simulations generate large amounts of data. Depending on the project goal, filtering and clustering of structural ensembles will be necessary. `pyrexMD` integrates different methods related to cluster analyses. It offers, e.g., multidimensional scaling (MDS)[149,150] or t-distributed stochastic neighbor embedding (TNSE)[137,138] for dimension reduction. Clustering itself can be performed with, e.g., KMEANS[163,164] or DBSCAN[167,168] (density-based spatial clustering of applications with noise).

The code example below applies TSNE for dimension reduction of distance matrices (DM). Afterwards, a 'fine' and 'coarse' KMeans clustering is performed with 10 and 20 cluster centers, respectively. The results are visualized in Fig. 7.5.

```
import pyrexMD.misc as misc
import pyrexMD.analysis.cluster as clu

# load data of pre-filtered frames
QDATA = misc.pickle_load("./data/QDATA.pickle")
RMSD = misc.pickle_load("./data/RMSD.pickle")
GDT_TS = misc.pickle_load("./data/GDT_TS.pickle")
score_file = "./data/energies.log"
ENERGY = misc.read_file(score_file, usecols=1, skiprows=1)
DM = clu.read_h5("./data/DM.h5")
```

```
# apply TSNE for dimension reduction
tsne = clu.apply_TSNE(DM, n_components=2, perplexity=50)

### apply KMeans on TSNE-transformed data (two variants with low and high cluster number)
cluster10 = clu.apply_KMEANS(tsne, n_clusters=10)
cluster20 = clu.apply_KMEANS(tsne, n_clusters=20)

### plot cluster data
# here: TSNE-transformed data with n_clusters = 20
# also: plot cluster centers with different colors
clu.plot_cluster_data(cluster20, tsne)
clu.plot_cluster_center(cluster10, marker="o", color="red", ms=20)
clu.plot_cluster_center(cluster20, marker="o", color="black")
```

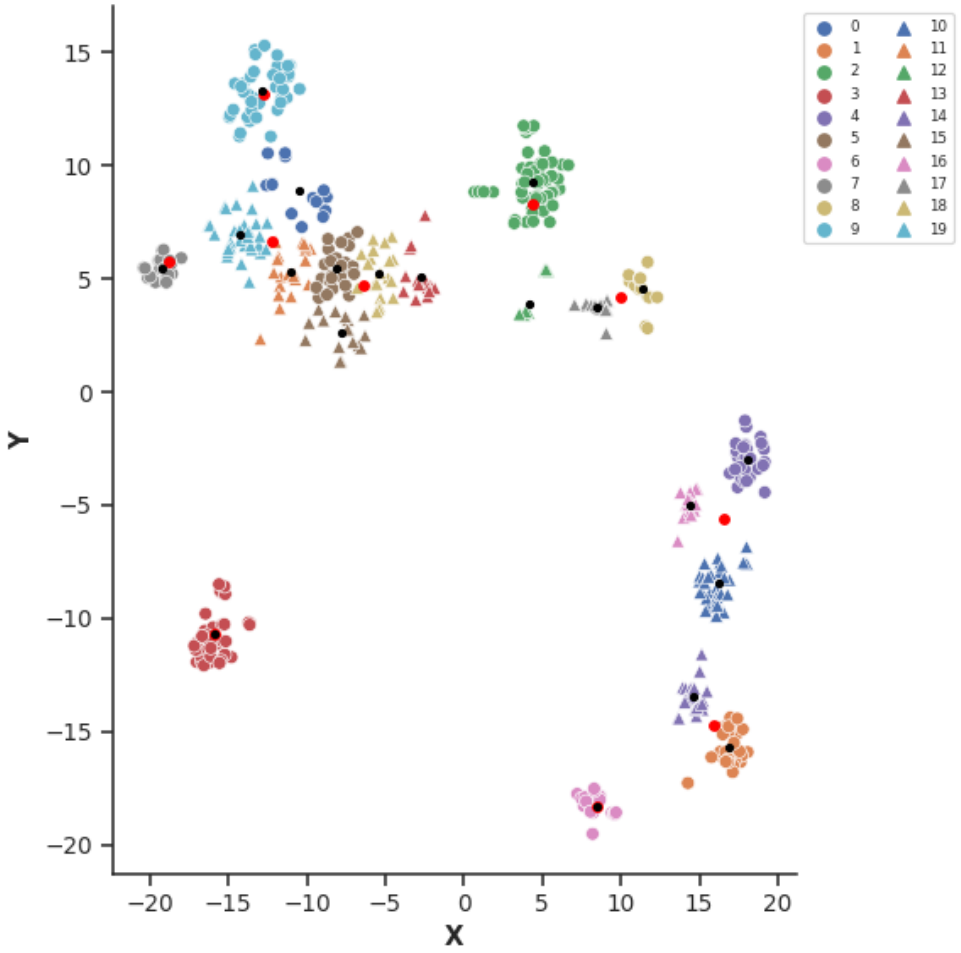

Figure 7.5. Exemplary application of dimension-reduction and clustering. TSNE representation of KMEANS-clustered ($k = 20$) protein structures according to the given code example. Additionally, cluster centers are visualized by black and red dots.

Furthermore, it is possible to link both *scores data* (energies) and *accuracy data* (GDT and RMSD) to clusters. This information can then be used to compare, sort and select the individual cluster ensembles. The code example below shows how to apply this feature.

```
### map scores (energies) and accuracy (GDT, RMSD) to clusters
cluster10_scores = clu.map_cluster_scores(cluster_data=cluster10, score_data=score_file)
cluster10_accuracy = clu.map_cluster_accuracy(cluster_data=cluster10, GDT=GDT_TS, RMSD=RMSD)
cluster20_scores = clu.map_cluster_scores(cluster_data=cluster20, score_data=score_file)
cluster20_accuracy = clu.map_cluster_accuracy(cluster_data=cluster20, GDT=GDT_TS, RMSD=RMSD)

### print table with cluster scores stats
clu.WF_print_cluster_scores(cluster_data=cluster10, cluster_scores=cluster10_scores)

### print table with cluster accuracy stats
clu.WF_print_cluster_accuracy(cluster_data=cluster10, cluster_accuracy=cluster10_accuracy)
```

The last commands are used to print a summary of the cluster scores and accuracy, which can also be saved to a log file if the *save_as* parameter is specified. The resulting output will look similar to:

```
cluster n10 scores (ranked by Emean)

ndx  size  compact | Emean    Estd    Emin      Emax      DELTA
  6   77     6.695 |-230.652  6.975  -246.249  -211.738  -7.67
  1   61      5.78 |-226.274  8.08   -241.86   -209.002  -3.292
  8   43     3.098 |-225.174  7.679  -242.951  -206.42   -2.192
  2   52     2.807 |-224.486  7.592  -240.913  -202.431  -1.504
  7   41     9.439 |-223.741  17.481 -249.136  -190.634  -0.759
  5   53     3.441 |-223.03   6.056  -237.002  -209.372  -0.048
  9   25     2.172 |-220.319  7.431  -231.002  -203.796   2.663
  0   80     9.121 |-216.962  7.09   -235.155  -200.969   6.02
  3   25     0.798 |-214.371  6.688  -228.33   -201.657   8.611
  4   43      1.91 |-194.022  2.585  -198.461  -190.412   28.96
--------------------------------------------------------------------
cluster n10 accuracy (ranked by GDT mean)

                  | GDT     GDT    GDT     GDT    | RMSD   RMSD   RMSD   RMSD
ndx  size  compact | mean    std    min     max    | mean   std    min    max
  2    52    2.807 | 77.296  1.815  73.81   80.953 | 2.555  0.154  2.182  3.076
  6    77    6.695 | 77.003  2.451  63.69   82.44  | 2.804  0.096  2.62   3.154
  1    61     5.78 | 75.943  2.325  71.728  82.142 | 2.85   0.096  2.567  3.03
  8    43    3.098 | 74.821  2.017  70.538  79.763 | 2.895  0.096  2.696  3.19
  7    41    9.439 | 73.374  14.68  41.37   94.94  | 2.873  1.192  0.996  6.501
  9    25    2.172 | 68.941  2.312  65.177  74.407 | 3.091  0.104  2.796  3.221
  0    80    9.121 | 64.695  3.943  55.057  74.703 | 3.444  0.238  2.719  3.896
  5    53    3.441 | 63.235  1.766  58.927  66.668 | 3.498  0.132  3.079  3.721
  3    25    0.798 | 60.012  2.289  56.248  63.69  | 3.804  0.087  3.684  4.043
  4    43     1.91 | 55.621  2.013  51.785  60.715 | 4.312  0.17   3.794  4.798
-------------------------------------------------------------------------------
```

PART III

CONCLUSIONS

8

Summary

In the following chapter I will summarize and highlight the results of my work regarding the application and optimization of contact-guided REX MD. Similarly to the outline of this thesis, I will shift the focus on smaller aspects and discuss their significance with respect to the general performance and utilization of this method. I will state the strengths and limitations of REX MD and highlight my contributions to elevate the expected outcome. Lastly, I will suggest further improvements and discuss possible future applications or research directions.

Proteins are complex and flexible macromolecules that act as nanosized workers in living organisms where they fulfill literally all critical tasks. Their biological function is closely related to their three dimensional structure as its shape and surface properties both define and limit environmental interactions. Many degenerative diseases, such as Alzheimer's or Parkinson's, are not caused by bacteria or viruses but instead by the misfolding and aggregation of certain proteins [232]. In general, protein misfolding can have severe consequences based on the importance of the protein's task and its relation to other biochemical interactions. Protein studies can therefore give valuable insight into associated mechanisms, which can also be used to improve drug design in order to prevent or weaken ailments.

Nowadays there exist many different methods and approaches to determine a protein's structure and function. However, to fully understand a protein and its dynamic interactions it is necessary to combine multiple sources. This also includes the mixing of both experimental and computational techniques. Each individual method is usually limited to specific aspects and cannot provide the complete picture on their own. For example, NMR spectroscopy [89] provides mainly information about the chemical environment which can be used to infer structural features. Cryo-EM [233] can only capture random planes/orientations of occurring protein conformations but cannot be used to study dynamics. On the other hand, computational methods can provide a detailed and atomic view of dynamic interactions using either MC- or MD-based approaches. These can be used to, e.g, aid the interpretation of experimental

data and bridge the gap between a static and dynamic viewpoint. However, the problem with such computational methods is that results are not measuring natural properties but can vary based on the applied model. More precisely, the outcome highly depends on the used simulation method, parameter choices, applied constraints, etc. For this reason, results should be carefully interpreted and ideally compared to experimental data.

Besides, each computational method has distinct strengths but also weaknesses. For example, normal MD simulations often struggle to reach physiologically meaningful timescales that are required for folding to the native state of a protein. Besides, simulations can get trapped in specific conformations due to potential energetic barriers that prevent the protein from further folding. Obtaining a native fold can thus become quite challenging. But is possible to overcome conformational entrapment at the cost of additional computing resources by using advanced sampling techniques such as REX. The primary goal of REX is to generate large amounts of physically meaningful structures. High temperatures provide sufficient energy to overcome local barriers while low temperatures lead to local searches of native-like conformations. Another positive feature of REX is that it can still generate thermodynamically correct ensembles despite the walk in temperature space. Its main weakness is that the computational costs are typically high since REX can be seen as running N MD simulations in parallel.

In the course of my work I extended the normal REX protocol with the goal to minimize the computational costs while increasing the chances of observing native-like conformations. The integration of a contact-based bias potential reduces the search space of REX and drives the simulation towards specific conformations. In my case, I apply an attractive potential that interacts only with presumably native contacts. A sigmoid shape was chosen as it can naturally reduce the negative influence of false-positive predictions.

In one of my studies I demonstrated the enrichment of native-like conformations by comparing contact-guided REX MD to regular and biased MD simulations[1]. Furthermore, I investigated the influence of native and non-native contacts by systematically testing many scenarios of varying bias quality and number of bias contacts. This allowed me to infer bias-quality thresholds that are necessary to obtain a significantly improved outcome as compared to regular REX. The most notable observations are 1) the true-positive rate should not fall below approximately 75% and 2) long-range contacts have a much stronger influence than short-range contacts. I concluded the findings by formulating bias guidelines and exemplarily discussed their application based on two contact-deriving methods, i.e. DCA and ResTriplet.

I also conducted a short study to optimize the sigmoid shape of my applied potential using one α-helical and one β-sheet structure. Here, I tested different values for the coupling strength λ and the equilibrium distance r_0. The direct comparison of resulting GDT distributions allowed me to identify good parameter ranges and to define an optimized potential for the general use-case.

Another important part of my work dealt with the optimization of starting structures that are used to populate each individual replica. The motivation is to provide additional pathways towards the native fold by maximizing the diversity of initial structures. At the same time, these structures should not deviate too much from the native fold, which would lead to diminishing returns in the saving of computational costs. I presented a *de novo* MC folding algorithm that is capable to quickly generate large amounts of starting structures by utilizing fragment insertion to speed up the process. Furthermore, I analyzed the generated structures of seven different proteins regarding their energy surface and structure accuracy (measured by GDT). I then presented two approaches on how to select among these structures and compared them against each other. I showed that it is generally better if structures are first clustered and then selected, instead of choosing them directly.

The potentially most critical aspect of my work was to find a solution on how to select native-like structure ensembles from REX simulations[3]. As previously mentioned, REX can generate large amounts of physically meaningful structures. But in my case I am only interested in a small subset, i.e. the highly native-like conformations. Using the entire REX trajectories of five different proteins as a basis to choose from, I presented four complex and robust algorithm chains that are capable to extract the wanted structure ensembles with high certainty. Furthermore, I introduced a numerical rating to objectively measure the selection performance of each pipeline. I investigated each variant in great detail and compared their overall performance against each other. Additionally, I discussed the pros and cons of each variant and tested if they are robust or lead to deviations in independent executions.

Lastly, I presented `pyrexMD`[2], a self-developed Python package that provides an interactive "all-purpose" environment to design, run, and analyze entire REX projects. It facilitates and automates many REX-related tasks and can greatly enhance the productivity. It also specializes in the visualization of data and offers a great variety of structure analysis and comparison functions.

In summary, the work presented in this thesis covers various critical aspects of contact-guided REX MD with the goal to optimize the end-to-end process and minimize the computational demands. This method excels in being easy-to-use as it does not require intensive parameter tuning. Besides, this method is extremely flexible and allows the integration of any kind of contact source or even a mixing of different sources. Nevertheless, REX MD still remains computationally demanding even with the presented improvements.

In 2009, it was shown that physical force fields are accurate to reversibly fold proteins on millisecond-scale MD simulations[35,36]. For example, multiple folding and unfolding events were observed for the viral protein gpW, which has a mixed structure of two α-helices and two β-sheets. In 2020, highly accurate structure prediction was achieved with a data-driven approach, i.e. with AlphaFold 2[100]. It was capable to reliably predict protein structures with an average GDT TS of almost 90, which is considered roughly equivalent to experimentally determined structures[100]. However, AlphaFold cannot provide additional insight about other meaningful events or the physical processes driving structure adoption. In the course of my work I showed that it is possible to obtain native or near-native protein structures with the physical approach of contact-guided REX MD. Physical force fields are sufficiently accurate to generate native-like conformations in biased simulations for small and mid-sized proteins. My simulations yielded structural models with GDT scores above 80 and in most cases around 90. Additionally, I demonstrated that I can reliably select native-like ensembles from 500 ns long REX simulations using my ensemble-selection algorithms.

I want to emphasize that REX simulations can be used for more than just the generation or refinement of native-like structures. My work primarily focuses on the lower-temperature data, but the remaining structural data at higher temperatures can also be used for many other analyses. For instance, it is possible to shift the perspective and instead of looking at fixed temperatures follow the individual turnaround cycles instead. This allows the exploration of many different folding paths in great detail. Additionally, since the integrated bias is known it can retrospectively be balanced out for all replicas. By doing so, one can deduce the free energy landscape of the simulated protein with statistical techniques such as the weighted histogram analysis method (WHAM)[234].

In its current state, I suggest two further improvements to contact-guided REX MD. First is to extend my study regarding the bias-potential optimization. My study provides some initial insights based on only two different structures. Additional testing is therefore required using other proteins and also with varying protein sizes. Besides, it would be beneficial to decrease the parameter step size and test more values, especially for r_0. My other suggestion is to modify the REX MD simulation in such a

way, that the simulation monitors which contacts are realized and which are not. These could then be dynamically switched on or off, with the intention to detect false-positive bias contacts and further negate their negative influence.

Lastly, I want to point out that the presented simulation method can also be successfully applied on RNA targets, which are studied more rarely than proteins. So far I only performed some initial tests and compared unbiased MD simulations with unbiased and biased REX MD at 100% TPR for three RNA targets. Initial results (cf. appendix F) indicate a high potential for this method, especially once the bias potential has been optimized for RNA targets. The achieved structure accuracy was sometimes better or on par with other methods, such as SimRNA with bias derived by mean field DCA or CoCoNet [235]. In general, it should be expected that the required simulation times are slightly longer than with protein targets. The main reasons are that 1) RNAs are typically elongated and loosely packed which makes them more flexible by nature [236] and 2) most RNA force fields are not as reliable and well-tested as protein force fields. However, recent improvements report a comparable performance to current state-of-the-art protein force fields after reparameterizating an AMBER RNA force field [237]. It may be of great interest to first optimize the parameters defining the sigmoid potential specifically for RNA targets. Afterwards, one can perform extensive testing on the general performance of contact-guided REX MD, similar to my bias-quality study [1].

PART IV

APPENDICES

A

Supplementary Information: Basics

The following content provides additional information to fully understand my work. The featured basics cover mainly technical details regarding my implementation of contact-guided REX MD. Appendix A.1 states my definition of native contacts and the mathematical conditions they have to fulfil. Appendix A.2 explains the technical integration of my sigmoid bias potential into MD simulations using `GROMACS`*. Appendix A.3 contains information regarding REX temperature distributions and their importance for an optimal simulation performance. Furthermore, I present my modified REX temperature generator and the motivation for the applied changes. Lastly, appendix A.4 exemplarily shows .mdp settings to perform REX simulations in* `GROMACS`*.*

A.1 Native Contacts

The definition of native contacts typically depends on the used representations during MD simulations. As I only perform all-atom simulations, I define native contacts via the conditions:

$$r_{ij} = ||r_i - r_j|| \leq 6\text{Å} \equiv r_{\text{nc}}, \tag{A.1}$$

$$\Delta ij = |i - j| \geq 4. \tag{A.2}$$

Eq. A.1 sets a distance threshold for C_α distances r_{ij} at 6 Å between two residues i an j. Eq. A.2 on the other hand excludes short-range pairs measured by their relative sequence distance, which correspond to the main diagonal of a contact map. Besides such contacts are irrelevant for my studies because their integration would have no significant effect due to the design of my sigmoid bias potential.

A.2 Bias Implementation

The implementation of a contact-based bias potential can be achieved via tabulated bonded interactions[109]. In GROMACS such interactions must be specified for bonds, angles and dihedrals separately, which offers maximum flexibility to the user. My intention is to apply an attractive force on selected C_α pairs that is distance dependant in order to guide protein folding while being less sensitive towards false-positive bias contacts. The potential for such bonded interactions is given by[109]

$$V_b(r_{ij}) = k f_n^b(r_{ij}), \tag{A.3}$$

with the potential V, distance r_{ij} between atoms i and j, force constant k, a lookup table f_n, and the index b representing bond interactions. Note the similarity between Eq. A.3 and Eq. 3.33, where k corresponds to the coupling strength λ and f_n is the sigmoidal potential $\sigma(r)$.

The lookup table describes the potential's shape and must be formatted with the three columns[109]

$$\text{r}, \qquad \text{f(r)}, \qquad \text{-f'(r)}$$

where the first column represents the pair distance r in nm, the second column describes the potential's shape and the last column its negative derivative. When designing the lookup table it is important to keep the spacing of Δr uniform. A single-precision build of GROMACS requires a spacing of $\Delta r = 0.002\,\text{nm}$, whereas a double-precision build requires $\Delta r = 0.0005\,\text{nm}$[109]. It is also necessary to provide a table of sufficient length. GROMACS does interpolate both potential and force based on the provided lookup table but stops the simulation if atom distances exceed the table's range. I want to emphasize that the table names are hardcoded into GROMACS and must be named according to the convention "table_bn.xvg" with $n \in \{0, 1, 2, ...\}$.

To specify which atoms are affected by the tabulated bonded interactions, the user has to modify the *bonds section* of the topology file before running GROMACS' pre-processor to create relevant MD files. A valid modification is given by the following topolgy code:

```
[ bonds ]
;  ai    aj  funct    c0    c1
  612   739      9     0    10
  596   739      9     0    10
  553   739      9     0    10
  612   704      9     0    10
  296   772      9     0    10
  315   791      9     0    10
```

Here, atom numbers are represented by ai and aj, whereas funct 9 defines the application of tabulated bonded interactions. The last two parameters, i.e. c0 and c1, stand for the hardcoded table number and the force constant k in kJ mol^{-1}, respectively. pyrexMD[2] offers build-in functions to design sigmoid-shaped lookup tables and to automatically modify topology files during the MD setup, which facilitates the integration of contact-based bias potentials. Finally, to include the lookup table into the simulation, the gmx mdrun command must be extended by

```
-tableb table_b0.xvg
```

A.3 REX Temperature Generator

In order to maximize the turnaround cycles of a REX simulation, exchange rates must be uniformly constant across the entire temperature range. The microstates of each replica typically correspond to a Boltzmann distribution, i.e $e^{-\beta E}$, with the inverse temperature $\beta = \frac{1}{k_B T}$ and the energy E. According to Eq. 3.32 exchange rates are proportional to the overlap of adjacent Boltzmann distributions. Therefore, by choosing an exponential temperature distribution for REX it is possible to maintain a nearly equivalent distribution overlap for all replicas. The simplest temperature model is given by

$$T_i = T_0 \cdot e^{ki}, \tag{A.4}$$

where T_i is the temperature of replica i and k specifies the distribution's growth speed. The problem of this simple model is that exchange rates are typically non-uniform and increase at higher temperatures. However, this can be easily fixed by modifying the temperature spacing of higher replicas. My modified REX temperature generator is given by

$$T_i = T_{i-1} + a_i \cdot \Delta, \tag{A.5}$$

$$\Delta = T_0 \cdot \left(e^{ki} - e^{k(i-1)} \right). \tag{A.6}$$

In this case, Δ denotes the temperature difference of two adjacent replicas, T_i the temperature of replica i, and a_i is a parameter to modify the temperature step size. Choosing $a_i = 1$ for all i would result in the same temperature distribution as the simple model from Eq. A.4. In contrast, by setting $a_i > 1$ the temperature spacing of adjacent replicas gets slightly increased. This in turn reduces the overlapping area of the respective Boltzmann distributions and makes the exchange rates more uniform. Based on my experience with this modified REX temperature generator, it is possible to obtain constant exchange rates across the entire temperature range when a_i is increased by approximately 2-5% every ten replicas. The drawback of this method is that each new system requires short REX runs ($\sim$ 5 runs) to screen occurring exchange rates before finding a good temperature distribution. Note that the literature offers other REX temperature generators that aim to predict ideal temperature distributions[238].

A.4 REX Settings for GROMACS

The settings below can be used in GROMACS for REX studies. Each replica requires its own .mdp file with a temperature distribution according to a REX temperature generator, as defined by Eqs. A.5 and A.6.

```
; Run parameters
integrator  = md                ; leap-frog integrator
dt          = 0.002             ; 2 fs
nsteps      = 250000000         ; nsteps * dt = 500 ns

; Output control
nstxout             = 50000     ; save coordinates every 100 ps (.trr size)
nstvout             = 50000     ; save velocities  every 100 ps (.trr size)
nstenergy           = 50000     ; save energies    every 100 ps (.edr size)
nstlog              = 5000      ; save log file    every  10 ps (.log size)
nstxout-compressed  = 5000      ; save compr.coord every  10 ps (.xtc size)
compressed-x-grps   = Protein   ; replaces xtc-grps

; Bond parameters
continuation        = no        ; first dynamics run
constraint_algorithm= lincs     ; holonomic constraints
constraints         = all-bonds ; all bonds (even heavy atom-H bonds) constrained
lincs_iter          = 1         ; accuracy of LINCS
lincs_order         = 4         ; also related to accuracy

; Neighborsearching
cutoff-scheme   = Verlet
ns_type         = grid      ; search neighboring grid cells
nstlist         = 10        ; 20 fs, largely irrelevant with Verlet
rcoulomb        = 1.0       ; short-range electrostatic cutoff (in nm)
rvdw            = 1.0       ; short-range van der Waals cutoff (in nm)

; Electrostatics
coulombtype     = PME       ; Particle Mesh Ewald for long-range electrostatics
pme_order       = 4         ; cubic interpolation
fourierspacing  = 0.16      ; grid spacing for FFT

; Temperature coupling is on
tcoupl  = V-rescale             ; modified Berendsen thermostat
tc-grps = Protein Non-Protein   ; two coupling groups - more accurate
tau_t   = 0.1       0.1         ; time constant, in ps
ref_t   = 280.00    280.00      ; reference temperature, one for each group, in K

; Pressure coupling is off
pcoupl = no ; no pressure coupling in NVT

; Periodic boundary conditions
pbc = xyz   ; 3-D PBC

; Dispersion correction
DispCorr = EnerPres ; account for cut-off vdW scheme

; Velocity generation
gen_vel  = yes   ; assign velocities from Maxwell distribution
gen_temp = 280.00  ; temperature for Maxwell distribution
gen_seed = -1   ; generate a random seed
```

B

Supplementary Information: Bias-Quality Study

The following content provides additional information that is relevant for the bias-quality study presented in section 4.1. Appendix B.1 gives a detailed summary of the applied REX temperature distributions during the Trp-Cage and Villin Headpiece simulations. It states the applied distribution function, the chosen parameters and the resulting temperatures for each replica. Appendix B.2 contains contact map figures highlighting the order of selected contact pairs and if they are native. Additionally, selected contact pairs are visualized in a 3D protein model. Lastly, in appendix B.3 I investigate two different methods of contact derivation (DCA and ResTriplet) and compare their contact predictions against each other using a contact map representation. Such information is relevant for section 4.1.4, where I formulate bias guidelines for contact-guided REX MD.

B.1 Used Temperature Distributions

Used Trp-Cage Temperatures

```
REX Temperature Distribution:
T_0 = 300 K ; DELTA = T_0 * (exp(k*i)-exp(k*(i-1)))
T_i = T_(i-1) + a_i * DELTA

Chosen Parameter:
k =0.0115
a0=1.00 for i =  0..9
a1=1.04 for i =  10..19
a2=1.08 for i =  20..29
a3=1.12 for i =  30..39
a4=1.16 for i =  40..49
a5=1.20 for i =  50..59

Temperatures:
300.00, 303.47, 306.98, 310.53, 314.12, 317.76, 321.43, 325.15, 328.91, 332.71,
336.72, 340.76, 344.86, 349.00, 353.19, 357.43, 361.72, 366.06, 370.45, 374.88,
379.55, 384.26, 389.04, 393.86, 398.74, 403.68, 408.68, 413.73, 418.84, 424.01,
429.44, 434.93, 440.48, 446.09, 451.77, 457.52, 463.33, 469.21, 475.16, 481.17,
487.48, 493.85, 500.30, 506.83, 513.43, 520.10, 526.86, 533.69, 540.60, 547.59,
554.90, 562.30, 569.79, 577.36, 585.02, 592.77, 600.61, 608.54, 616.56, 624.67
```

Used VHP Temperatures

```
REX Temperature Distribution:
T_0 = 300 K ; DELTA = T_0 * (exp(k*i)-exp(k*(i-1)))
T_i = T_(i-1) + a_i * DELTA

Chosen Parameter:
k =0.0065
a0=1.00 for i =  0..9
a1=1.04 for i =  10..19
a2=1.08 for i =  20..29
a3=1.12 for i =  30..39
a4=1.16 for i =  40..49
a5=1.20 for i =  50..59
a6=1.24 for i =  60..69
a7=1.28 for i =  70..79
a8=1.32 for i =  80..89
a9=1.36 for i =  90..99

Temperatures:
300.00, 301.96, 303.93, 305.91, 307.90, 309.91, 311.93, 313.97, 316.01, 318.07,
320.23, 322.40, 324.59, 326.79, 329.00, 331.23, 333.47, 335.73, 338.00, 340.29,
342.68, 345.09, 347.51, 349.95, 352.40, 354.87, 357.35, 359.86, 362.37, 364.91,
367.55, 370.22, 372.90, 375.60, 378.31, 381.04, 383.79, 386.56, 389.35, 392.16,
395.08, 398.02, 400.99, 403.97, 406.97, 409.99, 413.03, 416.09, 419.17, 422.27,
425.50, 428.75, 432.02, 435.31, 438.62, 441.96, 445.31, 448.69, 452.09, 455.52,
459.08, 462.66, 466.26, 469.89, 473.55, 477.23, 480.93, 484.65, 488.40, 492.18,
496.10, 500.04, 504.02, 508.02, 512.04, 516.09, 520.17, 524.27, 528.40, 532.56,
536.88, 541.22, 545.59, 549.99, 554.42, 558.88, 563.37, 567.88, 572.43, 577.00,
581.75, 586.53, 591.33, 596.17, 601.04, 605.94, 610.88, 615.84, 620.84, 625.87
```

B.2 Contact Maps

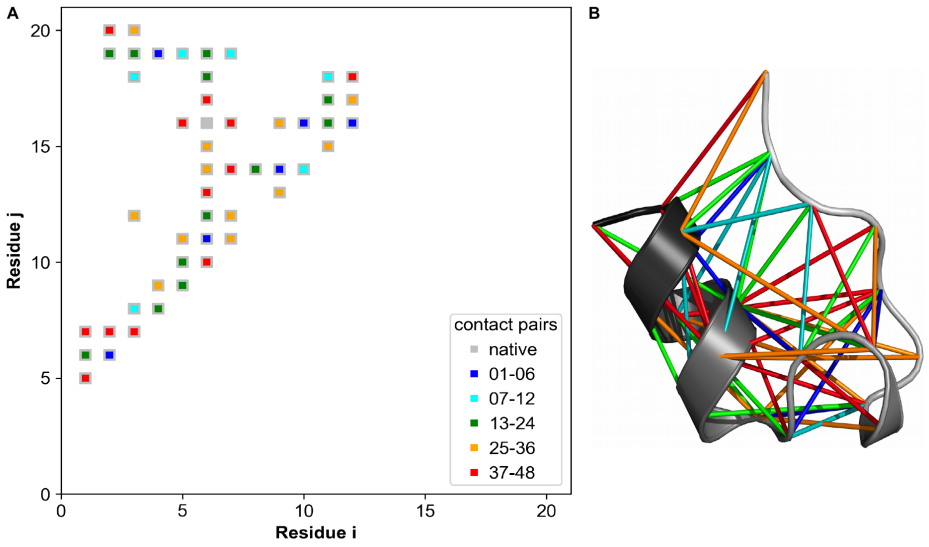

Figure B.1. Restraints used in Trp-Cage REX MD simulations at 100% TPR. (A) Contact map displaying native contacts as gray squares. Randomly selected contact pairs which were used as restraints are colored based on their batch. (B) Tertiary structure of Trp-Cage showing the contact pairs in the same color as in the contact map. Reproduced from Ref.[1] under CC BY 4.0.

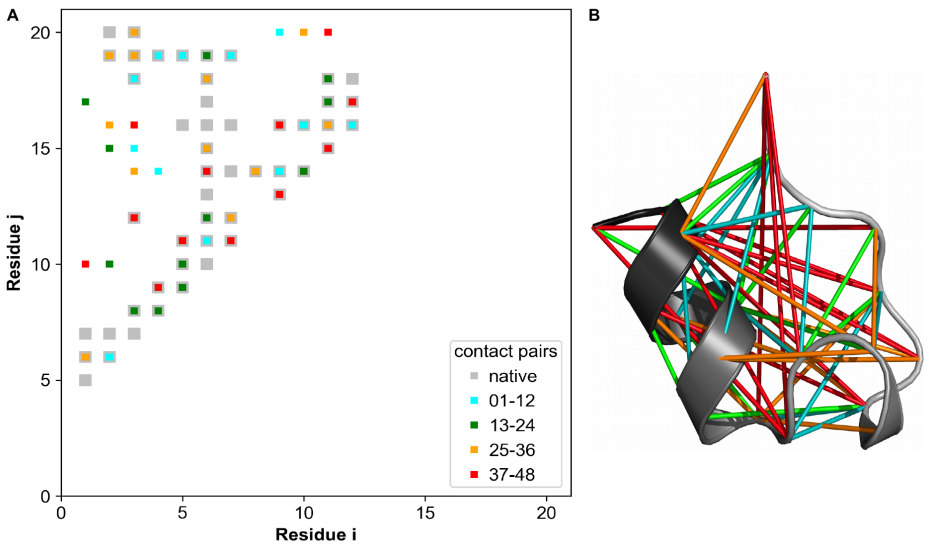

Figure B.2. Restraints used in Trp-Cage REX MD simulations at 75% TPR. (A) Contact map displaying native contacts as gray squares. Randomly selected contact pairs which were used as restraints are colored based on their batch. (B) Tertiary structure of Trp-Cage showing the contact pairs in the same color as in the contact map. Reproduced from Ref.[1] under CC BY 4.0.

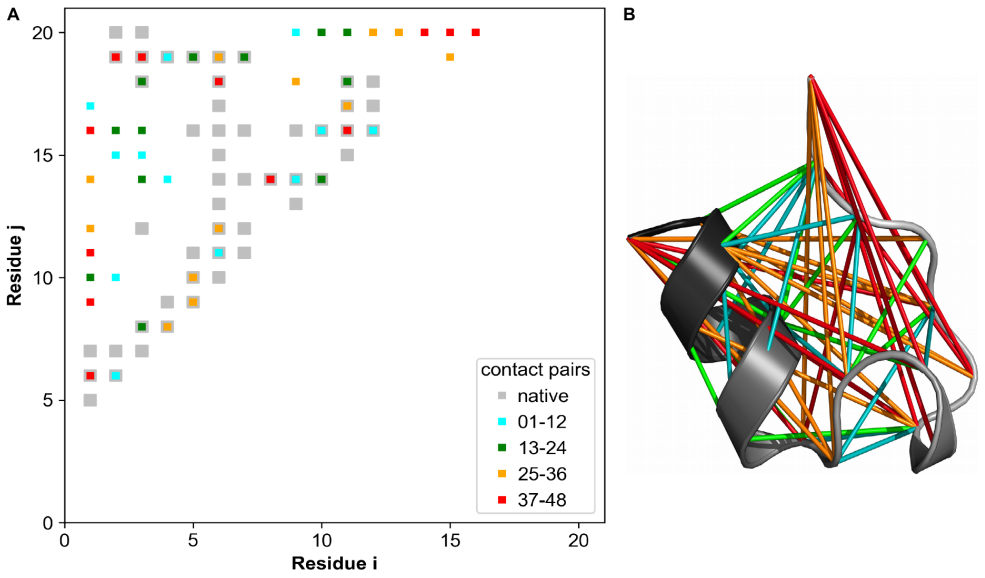

Figure B.3. Restraints used in Trp-Cage REX MD simulations at 50% TPR. (A) Contact map displaying native contacts as gray squares. Randomly selected contact pairs which were used as restraints are colored based on their batch. (B) Tertiary structure of Trp-Cage showing the contact pairs in the same color as in the contact map. Reproduced from Ref.[1] under CC BY 4.0.

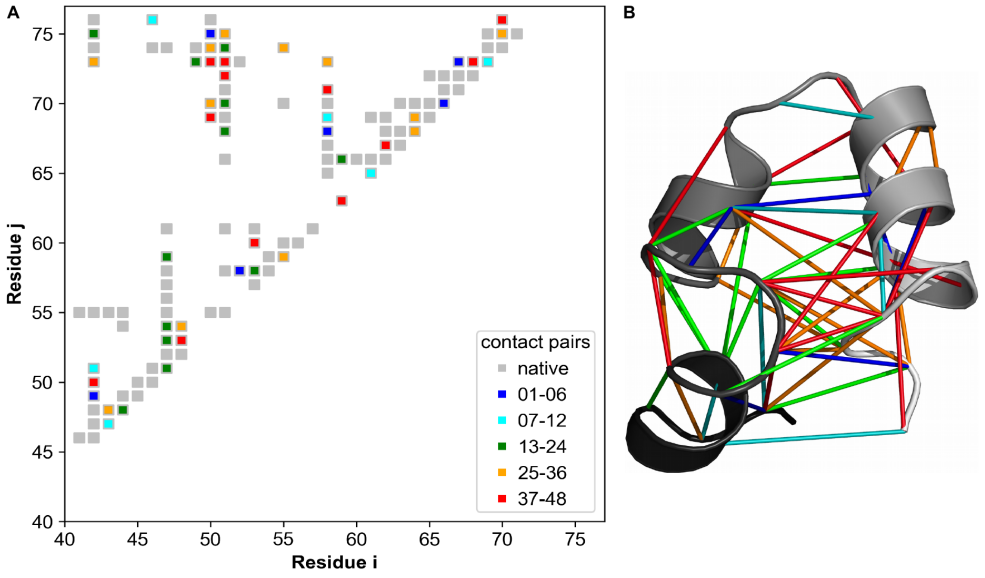

Figure B.4. Restraints used in VHP REX MD simulations at 100% TPR. (A) Contact map displaying native contacts as gray squares. Randomly selected contact pairs which were used as restraints are colored based on their batch. (B) Tertiary structure of VHP showing the contact pairs in the same color as in the contact map. Reproduced from Ref.[1] under CC BY 4.0.

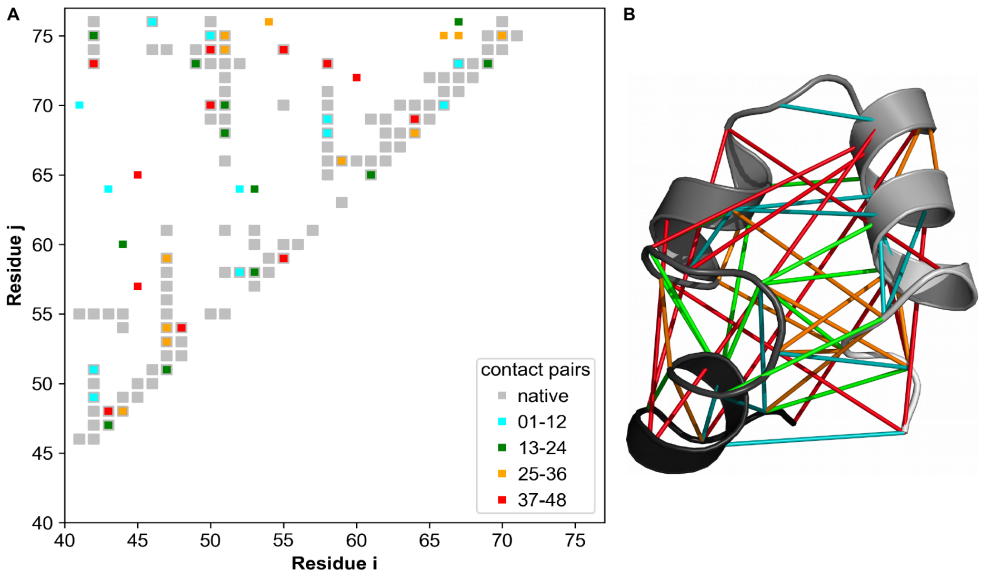

Figure B.5. Restraints used in VHP REX MD simulations at 75% TPR. (A) Contact map displaying native contacts as gray squares. Randomly selected contact pairs which were used as restraints are colored based on their batch. (B) Tertiary structure of VHP showing the contact pairs in the same color as in the contact map. Reproduced from Ref.[1] under CC BY 4.0.

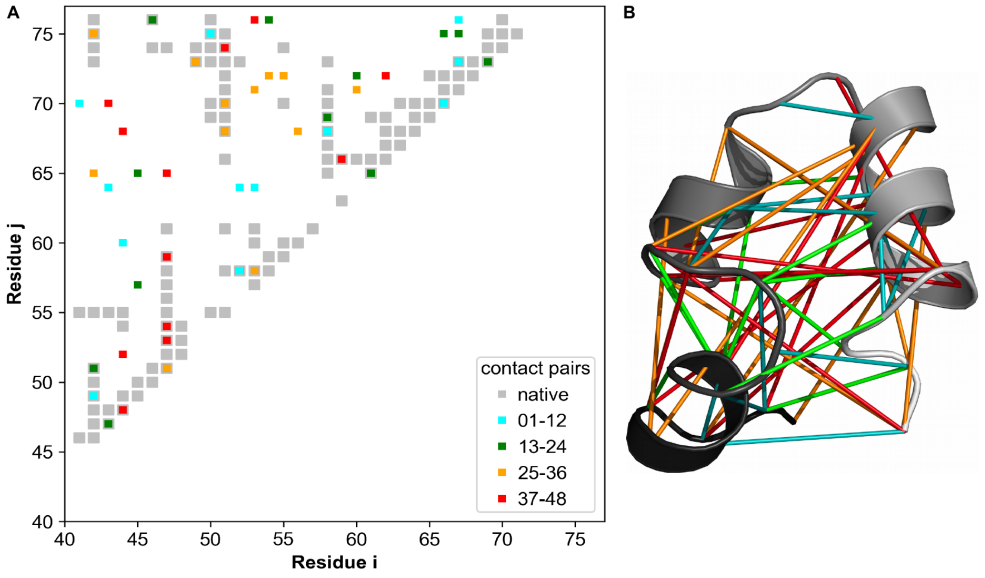

Figure B.6. Restraints used in VHP REX MD simulations at 50% TPR. (A) Contact map displaying native contacts as gray squares. Randomly selected contact pairs which were used as restraints are colored based on their batch. (B) Tertiary structure of VHP showing the contact pairs in the same color as in the contact map. Reproduced from Ref.[1] under CC BY 4.0.

B.3 Bias Guidelines: DCA vs. ResTriplet

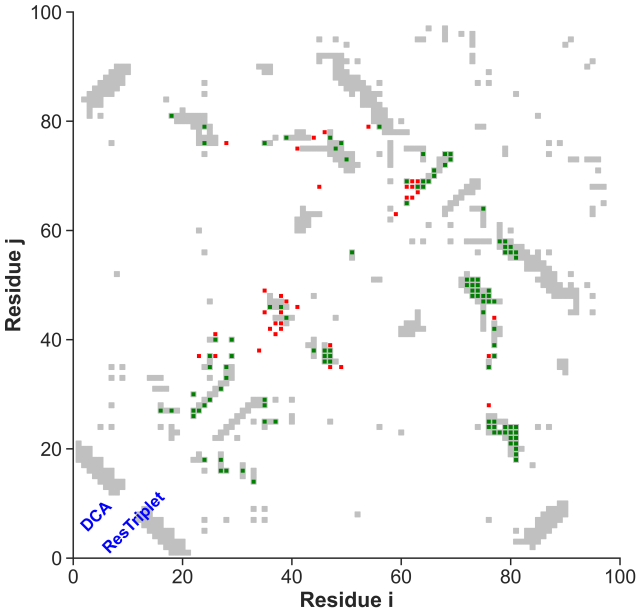

Figure B.7. Comparison of contact predictions by DCA and ResTriplet (pdbid: 1a70, N = 0.75 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 0.75 L contacts (L: sequence length).

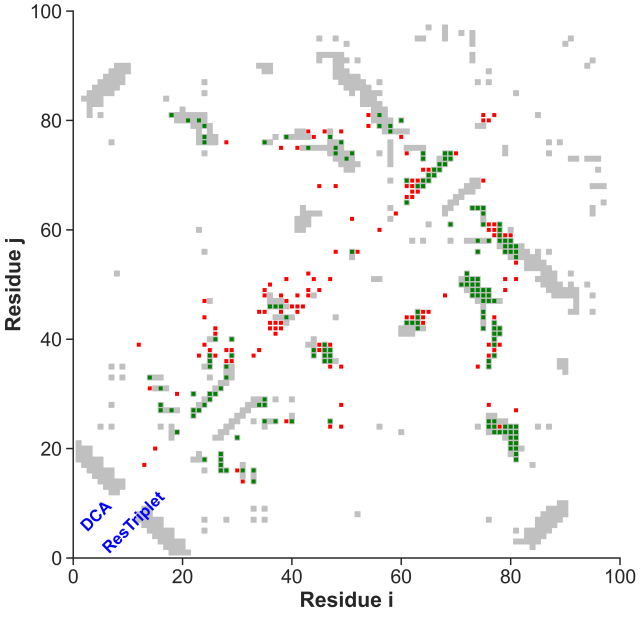

Figure B.8. Comparison of contact predictions by DCA and ResTriplet (pdbid: 1a70, N = 1.5 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

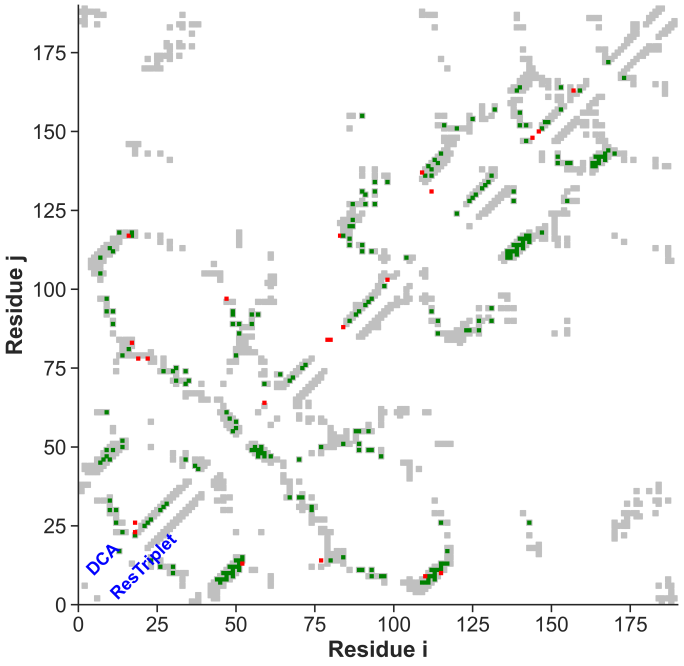

Figure B.9. Comparison of contact predictions by DCA and ResTriplet (pdbid: 1atz, N = 0.75 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 0.75 L contacts (L: sequence length).

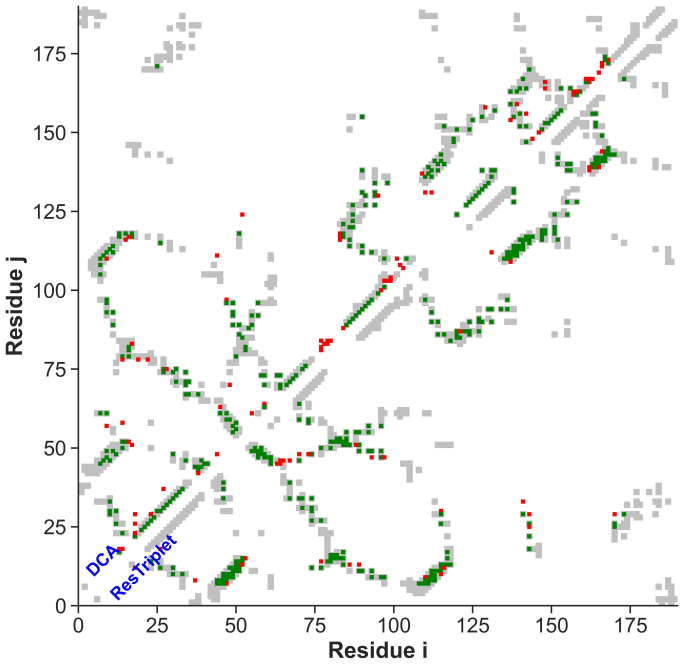

Figure B.10. Comparison of contact predictions by DCA and ResTriplet (pdbid: 1atz, N = 1.5 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

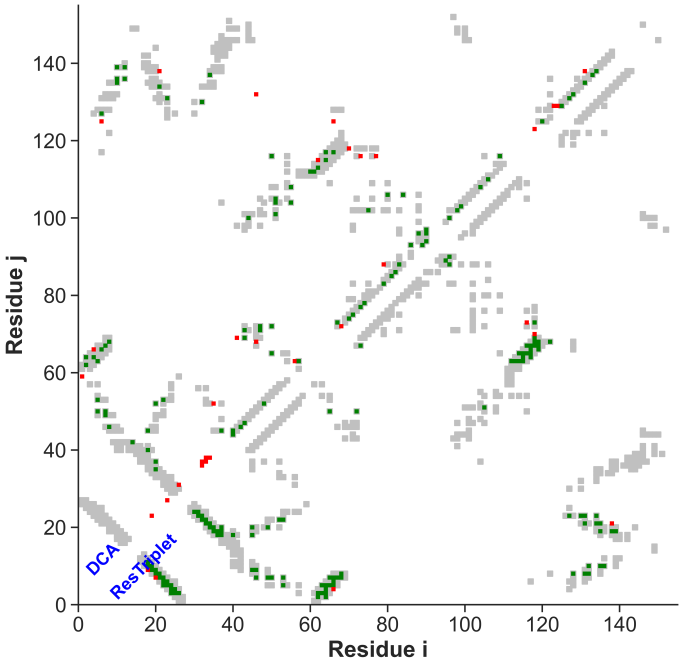

Figure B.11. Comparison of contact predictions by DCA and ResTriplet (pdbid: 1f21, N = 0.75 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 0.75 L contacts (L: sequence length).

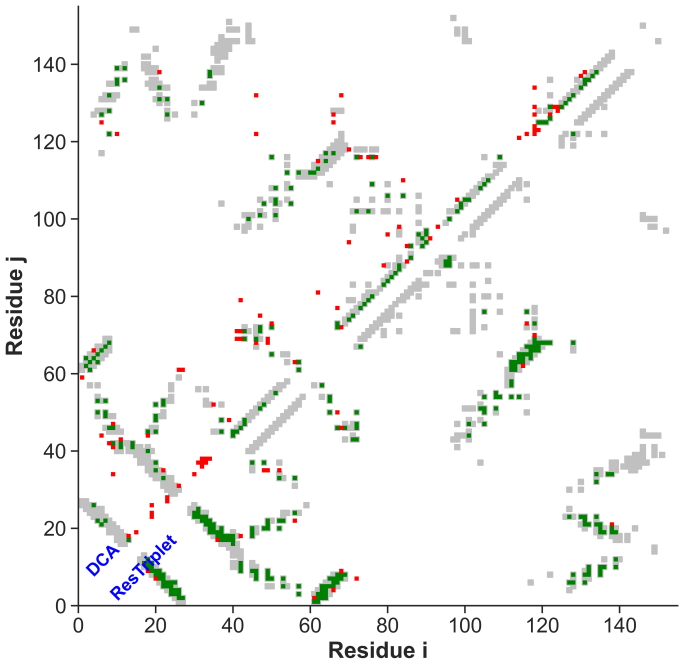

Figure B.12. Comparison of contact predictions by DCA and ResTriplet (pdbid: 1f21, N = 1.5 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

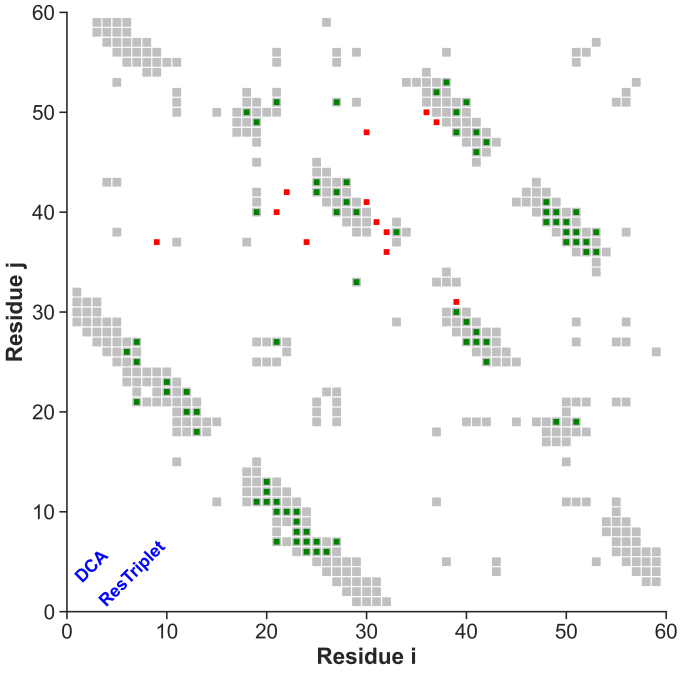

Figure B.13. Comparison of contact predictions by DCA and ResTriplet (pdbid: 2hda, N = 0.75 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 0.75 L contacts (L: sequence length).

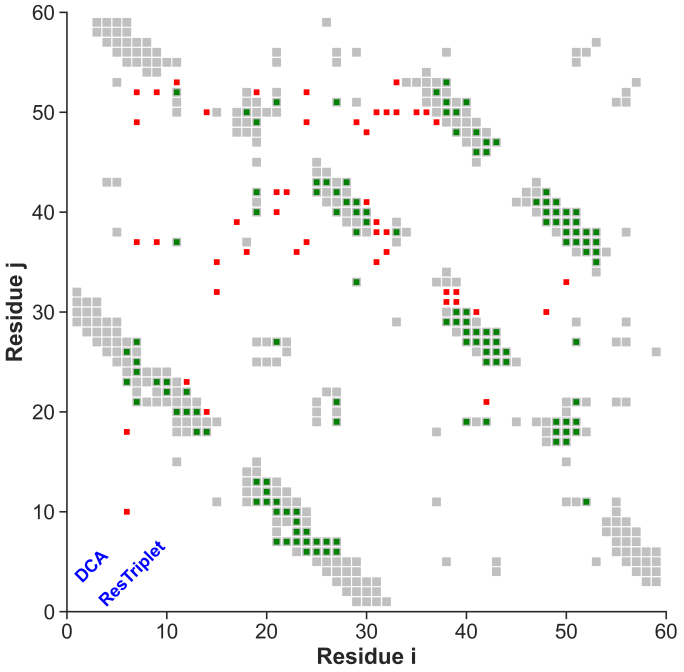

Figure B.14. Comparison of contact predictions by DCA and ResTriplet (pdbid: 2hda, N = 1.5 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

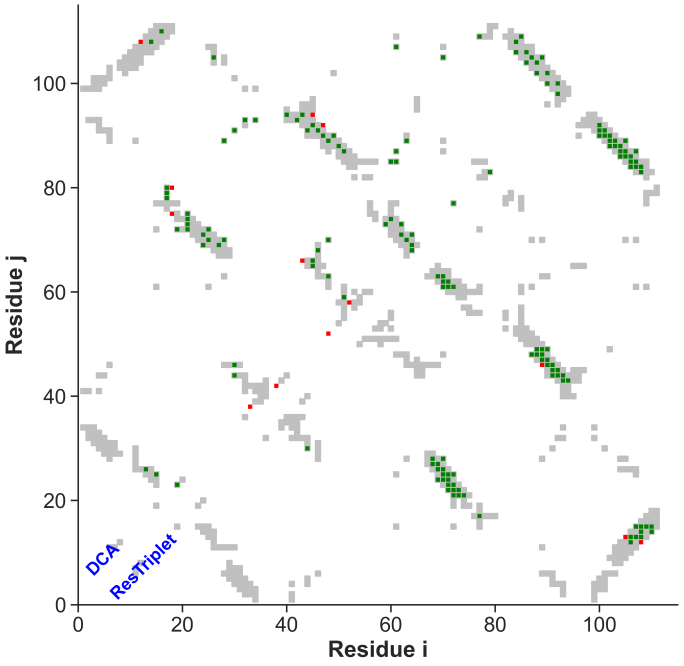

Figure B.15. Comparison of contact predictions by DCA and ResTriplet (pdbid: 2o72, N = 0.75 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 0.75 L contacts (L: sequence length).

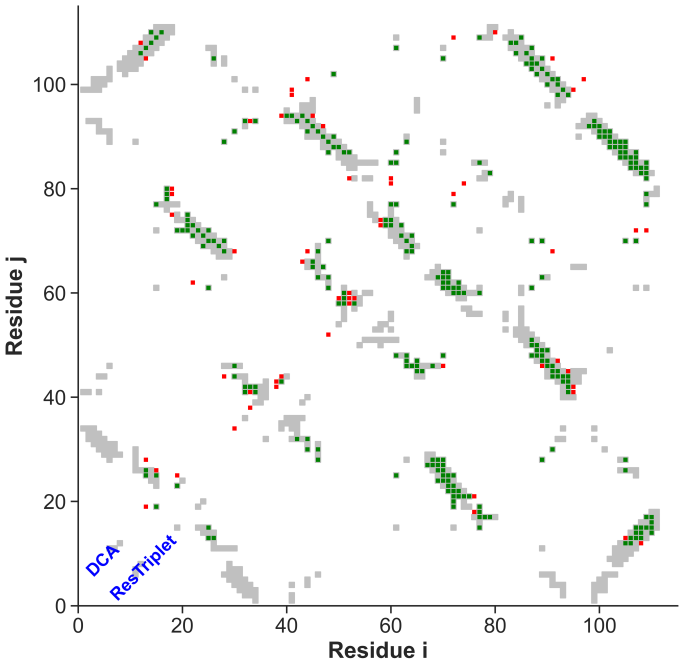

Figure B.16. Comparison of contact predictions by DCA and ResTriplet (pdbid: 2o72, N = 1.5 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

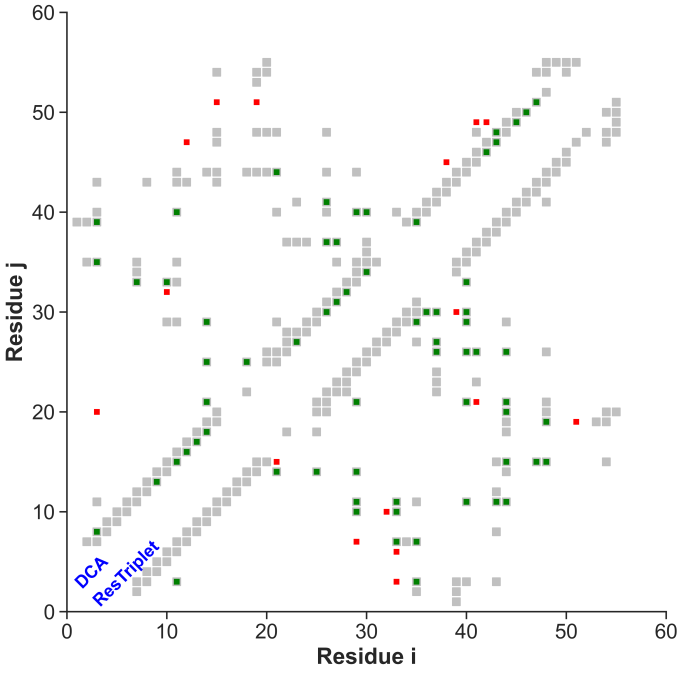

Figure B.17. Comparison of contact predictions by DCA and ResTriplet (pdbid: 2vi6, N = 0.75 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 0.75 L contacts (L: sequence length).

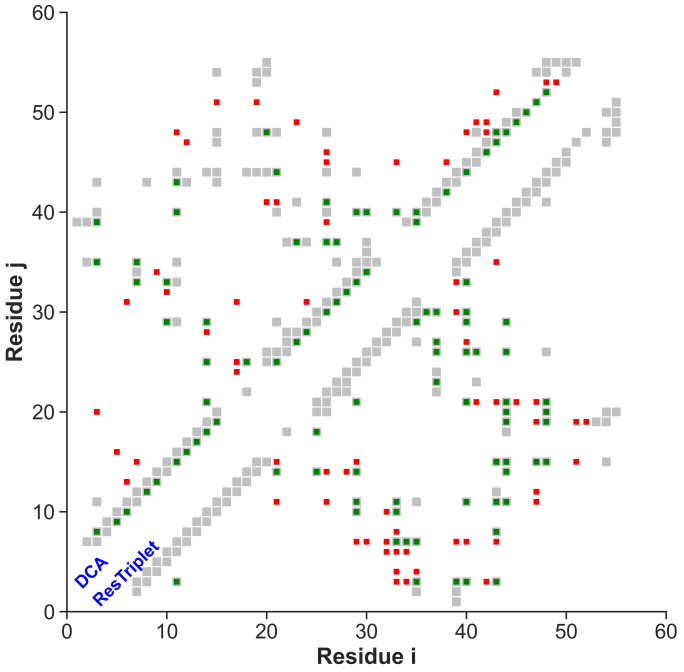

Figure B.18. Comparison of contact predictions by DCA and ResTriplet (pdbid: 2vi6, N = 1.5 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

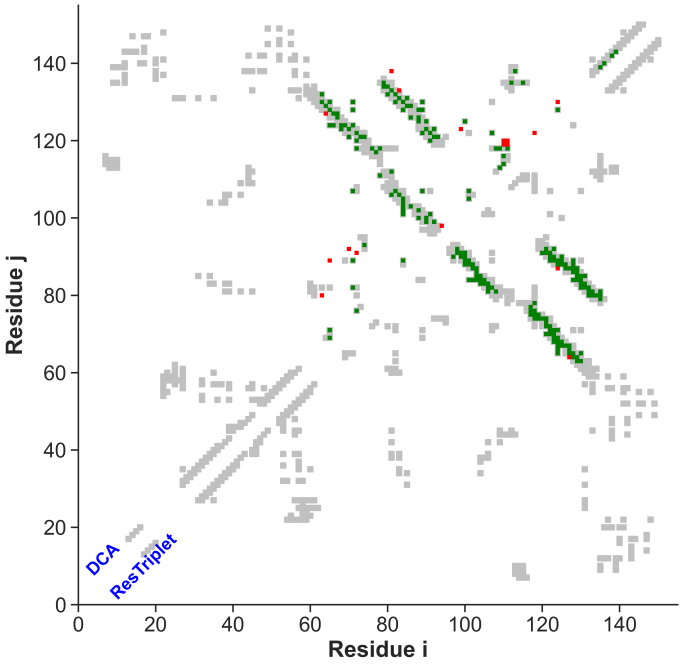

Figure B.19. Comparison of contact predictions by DCA and ResTriplet (pdbid: 3fhi, N = 0.75 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 0.75 L contacts (L: sequence length).

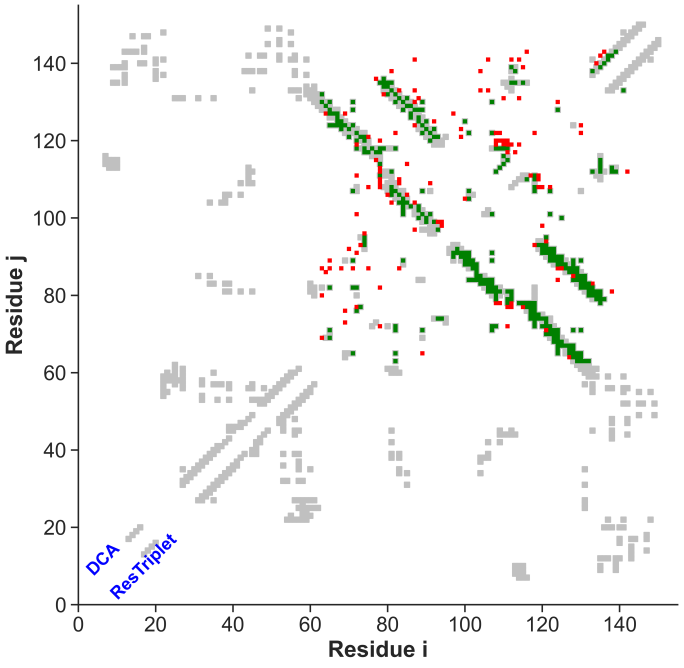

Figure B.20. Comparison of contact predictions by DCA and ResTriplet (pdbid: 3fhi, N = 1.5 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

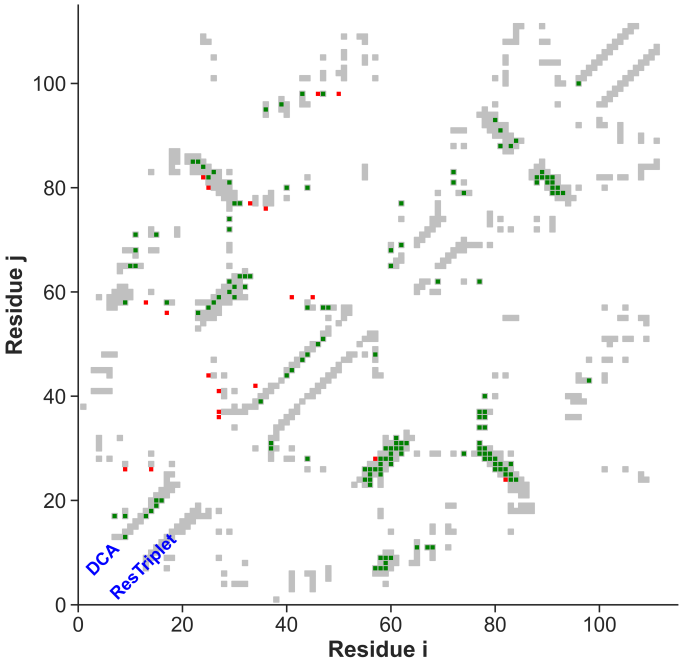

Figure B.21. Comparison of contact predictions by DCA and ResTriplet (pdbid: 3gnj, N = 0.75 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 0.75 L contacts (L: sequence length).

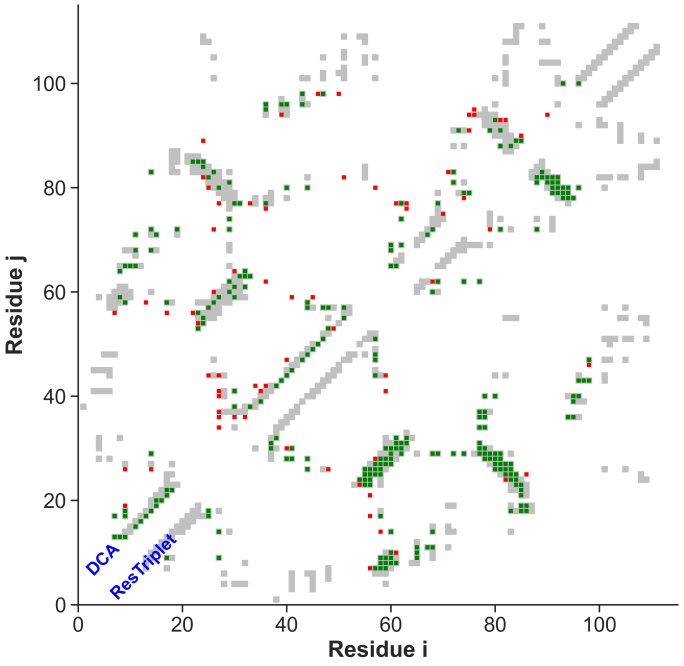

Figure B.22. Comparison of contact predictions by DCA and ResTriplet (pdbid: 3gnj, N = 1.5 L). Depicted are native contacts (gray), true-positive predictions (green) and false-positive predictions (red). DCA predictions are visualized on the upper left side and ResTriplet predictions on the lower right side. Comparison is made for 1.5 L contacts (L: sequence length).

C

Supplementary Information: Bias-Optimization Study

The following content provides additional information that is relevant for the bias-potential optimization presented in section 4.2. Appendix C.1 contains contact maps of both test proteins to visualize the applied bias contacts and to classify their type (i.e. native, true-positive, false-positive). Appendix C.2 gives a statistical overview of the structures obtained from the simulations. The featured histograms display GDT distributions for investigated cases similar to section 4.2 but focus on HA distributions instead.

C.1 Contact Maps

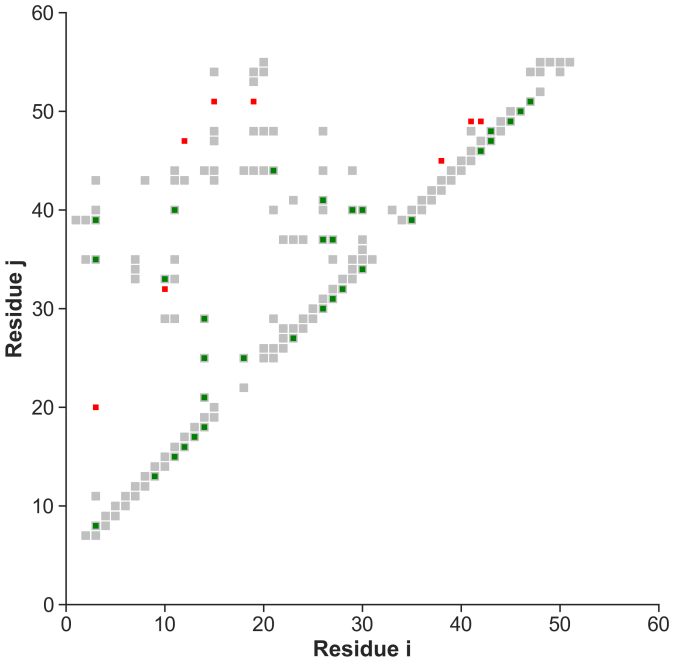

Figure C.1. Contact map of Nanog homeodomain (PDB id: 2vi6[191]).
Displayed are the native contacts (gray), true-positive bias contacts (green) and false-positive bias contacts (red). Integrated bias consisted of 40 contact pairs at 80% TPR.

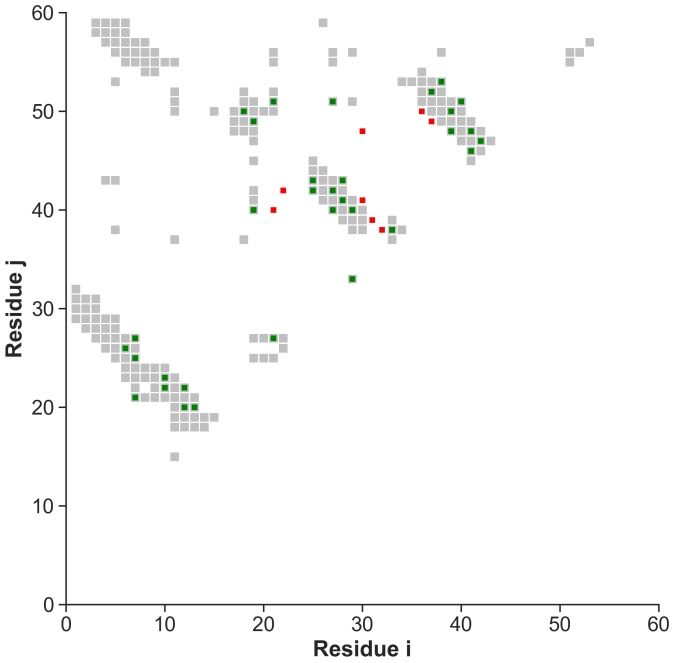

Figure C.2. Contact map of Yes SH3 domain (PDB id: 2hda[192]).
Displayed are the native contacts (gray), true-positive bias contacts (green) and false-positive bias contacts (red). Integrated bias consisted of 40 contact pairs at 80% TPR.

C.2 Histograms

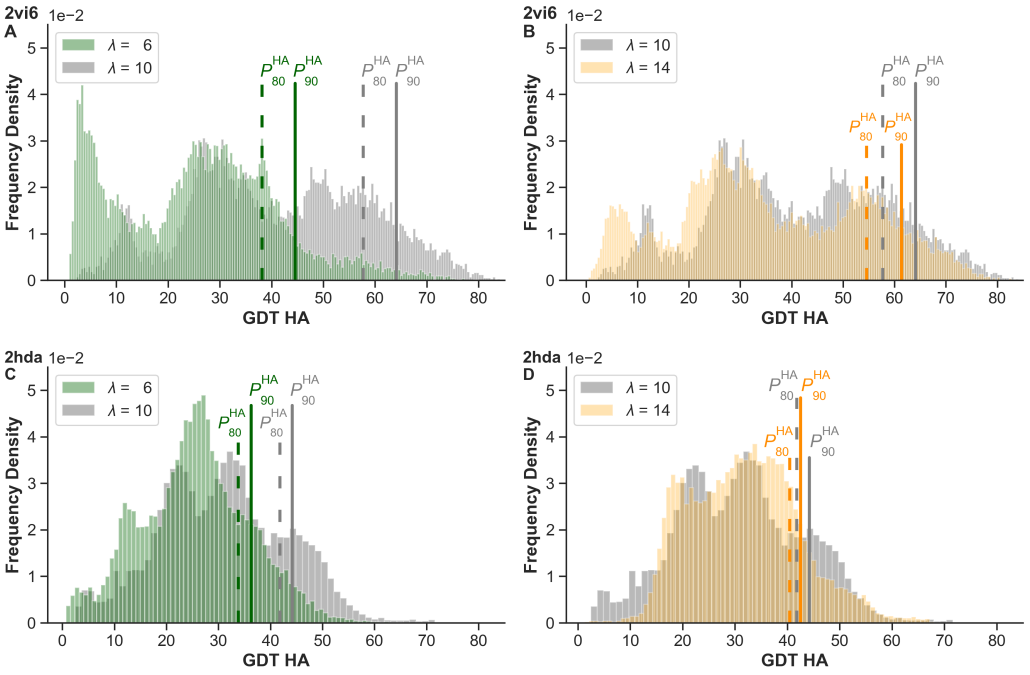

Figure C.3. GDT HA distributions based on λ parameter. Vertical lines represent the 80th and 90th percentile. **(A+B)** Nanog homeodomain (PDB id: 2vi6[191]). **(C+D)** Yes SH3 domain (PDB id: 2hda[192]).

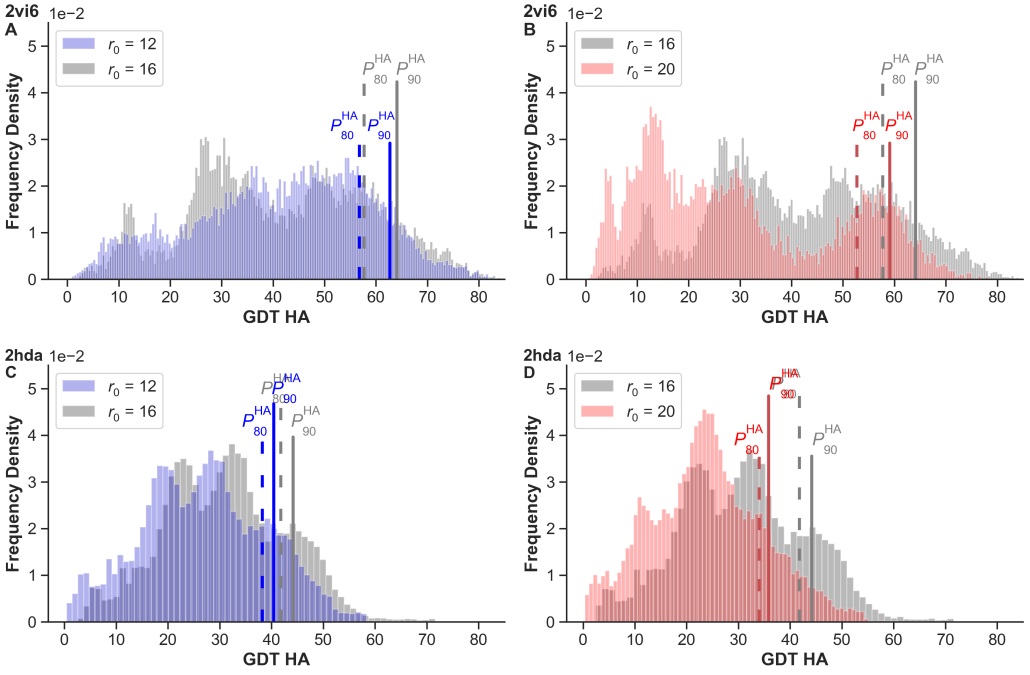

Figure C.4. GDT HA distributions based on r_0 parameter. Vertical lines represent the 80th and 90th percentile. **(A+B)** Nanog homeodomain (PDB id: 2vi6[191]). **(C+D)** Yes SH3 domain (PDB id: 2hda[192]).

D

Supplementary Information: Starting-Structure Generation

The following Python code was used to generate unique starting structures ("decoys") with `PyRosetta`[193]*. In a later step the starting structures were classified via multidimensional scaling and KMEANS clustering with* $N_{\text{clusters}} = N_{\text{replicas}}$*. During the setup of contact-guided REX MD in* `pyrexMD`[2] *it is possible to populate individual replicas with the lowest-scoring structures of each cluster. As explained in chapter 5, this diversification allows to maximize the sampling space and enables additional pathways towards the native state from the very beginning.*

D.1 De Novo Folding Algorithm (CODE)

```
# filename: denovo.py
# author: Arthur Voronin

import os
import pyrexMD.misc as _misc
import pyrosetta
from pyrosetta.rosetta import protocols
```

```
def setup_denovo_cfg(pdbid, fasta_seq, frag3mer, frag9mer, **kwargs):
    """
    Setup config file for ab initio folding algorithm.

    Args:
        pdbid (str): pdb id / pdb name
        fasta_seq (str): fasta sequence
        frag3mer (str): 3mer file path
        frag9mer (str): 9mer file path

    Keyword Args:
        fasta_seq_len (int): fasta sequence length
        frag3inserts (int): number of frag3 inserts
        frag9inserts (int): number of frag9 inserts
        folding_cycles (int): folding move cycles
        folding_repeats (int): folding move repeats
        job_name (str)
        n_decoys (int): total number of decoys
        n_cores (int): -np option for multiprocessing
        decoy_ndx_shift (int):
          | shift decoy index (output filename) by this value
          | required for multiprocessing to fix names of decoys
        kT (float): kT parameter during Monte-Carlo simulation

    Returns:
        denovo_cfg (CONFIG class)
            configs used as input for denovo.create_decoys()
    """
    default = {"pdbid": pdbid,
               "fasta_seq": fasta_seq,
               "fasta_seq_len": None,
               "frag3mer": frag3mer,
               "frag9mer": frag9mer,
               "frag3inserts": 3,
               "frag9inserts": 1,
               "folding_cycles": 1000,
               "folding_repeats": 10,
               "job_name": pdbid,
               "n_decoys": 10,
               "n_cores": 1,
               "decoy_ndx_shift": 0,
               "kT": 1.0}
    denovo_cfg = _misc.CONFIG(default)
    denovo_cfg.update_config(fasta_seq_len=len(denovo_cfg.fasta_seq))
    denovo_cfg.update_config(**kwargs)
    return denovo_cfg
```

```
def _create_decoys(denovo_cfg, output_dir="./output", fastrelax=True,
                   stream2pymol=True, save_log=True, **kwargs):
    """
    Create decoys within PyRosetta.

    Args:
        denovo_cfg (CONFIG class): output of denovo.setup_denovo_cfg()
        output_dir (str): output directory for decoys
        fastrelax (bool): apply fastrelax on decoys before dumping them as pdb
        stream2pymol (bool): stream decoys to PyMOL
        save_log (bool): save scores to logfile at <output_dir/scores.txt>

    Keyword Args:
        cprint_color (None, str): colored print color

    Returns:
        SCORES_low (list)
            centroid scores ~ score 3
        SCORES_high (list)
            fa scores ~ ref2015
    """
    default = {"cprint_color": "blue"}
    default_cfg = _misc.CONFIG(default, **kwargs)
    cfg = _misc.CONFIG(denovo_cfg, **default_cfg)

    # create output directory
    if output_dir[-1] == "/":
        output_dir = output_dir[:-1]
    if os.path.exists(output_dir) and cfg.n_cores == 1:
        msg = f"""Output directory '{output_dir}' already exists.
    Creating decoys will overwrite existing decoys in this directory.
    Proceed? [y/n]"""
        answer = input(msg).lower()
        if (answer == "y" or answer == "yes"):
            pass
        if (answer == "n" or answer == "no"):
            return
    _misc.mkdir(output_dir)

    # create decoys code
    # conversion movers
    to_centroid = pyrosetta.SwitchResidueTypeSetMover('centroid')
    to_fullatom = pyrosetta.SwitchResidueTypeSetMover('fa_standard')

    # score function and score array
    scorefxn_low = pyrosetta.create_score_function('score3')
    scorefxn_high = pyrosetta.create_score_function('ref2015')
```

```
# pose objects
# linear pose
pose_0 = pyrosetta.pose_from_sequence(cfg.fasta_seq)
pose_0.pdb_info().name(f"{cfg.pdbid} (linear)")

# test pose
pose = pyrosetta.Pose()
pose.assign(pose_0)
pose.pdb_info().name(cfg.pdbid)

# switch to centroid
to_centroid.apply(pose_0)
to_centroid.apply(pose)

# mover and fragset objects
movemap = pyrosetta.MoveMap()
movemap.set_bb(True)

fragset_3mer = pyrosetta.rosetta.core.fragment.ConstantLengthFragSet(3,
↪  cfg.frag3mer)
fragset_9mer = pyrosetta.rosetta.core.fragment.ConstantLengthFragSet(9,
↪  cfg.frag9mer)

mover_frag3 = protocols.simple_moves.ClassicFragmentMover(fragset_3mer, movemap)
mover_frag9 = protocols.simple_moves.ClassicFragmentMover(fragset_9mer, movemap)

insert_frag3 = protocols.moves.RepeatMover(mover_frag3, cfg.frag3inserts)
insert_frag9 = protocols.moves.RepeatMover(mover_frag9, cfg.frag9inserts)

folding_mover = protocols.moves.SequenceMover()
folding_mover.add_mover(insert_frag9)
folding_mover.add_mover(insert_frag3)

# MC stuff
mc = pyrosetta.MonteCarlo(pose, scorefxn_low, cfg.kT)
trial = pyrosetta.TrialMover(folding_mover, mc)
folding = protocols.moves.RepeatMover(trial, cfg.folding_cycles)
#jd = PyJobDistributor(cfg.job_name, cfg.n_decoys, scorefxn_high)

if stream2pymol:
    pmm = pyrosetta.PyMOLMover()
    pmm.keep_history(True)

# job distributor stuff
worker_jobs = int(cfg.n_decoys/cfg.n_cores)
# SCORES_low = [0]*(worker_jobs)  # scores array
# SCORES_high = [0]*(worker_jobs)  # scores array
DECOYS = []
```

```
    SCORES_low = []
    SCORES_high = []
    for i in range(1, worker_jobs+1):
        _misc.cprint(f">>> Working on decoy:
        ↪  {output_dir}/{cfg.job_name}_{cfg.decoy_ndx_shift+i}.pdb",
        ↪  cfg.cprint_color)
        DECOYS.append(f"{cfg.job_name}_{cfg.decoy_ndx_shift+i}")
        pose.assign(pose_0)
        pose.pdb_info().name(f"{cfg.job_name}_{cfg.decoy_ndx_shift+i}")
        mc.reset(pose)
        for j in range(cfg.folding_repeats):
            folding.apply(pose)
            mc.recover_low(pose)

        #SCORES_low[i] = scorefxn_low(pose)
        SCORES_low.append(scorefxn_low(pose))
        to_fullatom.apply(pose)

        if fastrelax:
            relax = protocols.relax.FastRelax()
            relax.set_scorefxn(scorefxn_high)
            relax.apply(pose)

        #SCORES_high[i] = scorefxn_high(pose)
        SCORES_high.append(scorefxn_high(pose))
        pose.dump_pdb(f"{output_dir}/{cfg.job_name}_{cfg.decoy_ndx_shift+i}.pdb")

        if stream2pymol:
            pmm.apply(pose)

    if save_log:
        logfile = f"{output_dir}/scores.txt"
        if not os.path.exists(logfile):
            with open(logfile, "w") as log:
                log.write(f"{'decoy'}\t{'score3'}\t{'ref2015'}\n")  # write header
        with open(logfile, "a") as log:
            #DECOYS = [f"{cfg.job_name}_{cfg.decoy_ndx_shift+i}" for i in
            ↪  range(worker_jobs)]
            table_str = _misc.print_table([DECOYS, SCORES_low, SCORES_high])
            log.write(table_str)

    return SCORES_low, SCORES_high
```

E

Supplementary Information: Ensemble Selection

The following content provides additional information that is relevant for the selection of representative ensemble as presented in chapter 6. Appendix E.1 gives a detailed summary of the applied REX temperature distributions during the REX simulations. It states the applied distribution function, the chosen parameters and the resulting temperatures for each replica. Appendix E.2 contains various supplementary figures, such as contact maps and correlation scatter plots. Furthermore, the MDS and TSNE representations can be used for a detailed comparison of the four presented algorithm chains from sections 6.3 and 6.4. Lastly, appendix E.3 provides additional tables, which state the starting decoy accuracy (measured by RMSD and GDT) and the used bias contacts during REX. However, most notably is the detailed summary of selected ensemble clusters and their structural accuracy relative to the protein's native state.

E.1 Used Temperature Distribution

```
REX Temperature Distribution:
T_0 = 280 K ; DELTA = T_0 * (exp(k*i)-exp(k*(i-1)))
T_i = T_(i-1) + a_j * DELTA

Chosen Parameter:
k = 0.006
a_0 = 1.00 for i = 0..9
a_1 = 1.04 for i = 10..19
a_2 = 1.08 for i = 20..29
a_3 = 1.12 for i = 30..39
a_4 = 1.16 for i = 40..49
a_5 = 1.20 for i = 50..59
a_6 = 1.24 for i = 60..69
a_7 = 1.28 for i = 70..79

Temperatures:
280.00, 281.69, 283.38, 285.09, 286.80, 288.53, 290.26, 292.01, 293.77, 295.54,
297.39, 299.25, 301.12, 303.00, 304.90, 306.80, 308.72, 310.65, 312.59, 314.54,
316.58, 318.63, 320.70, 322.77, 324.86, 326.96, 329.08, 331.21, 333.35, 335.50,
337.74, 340.00, 342.28, 344.56, 346.86, 349.18, 351.51, 353.85, 356.21, 358.58,
361.05, 363.53, 366.03, 368.55, 371.08, 373.62, 376.18, 378.76, 381.35, 383.96,
386.67, 389.40, 392.14, 394.91, 397.69, 400.48, 403.29, 406.12, 408.97, 411.83,
414.81, 417.81, 420.82, 423.85, 426.90, 429.97, 433.05, 436.16, 439.28, 442.42,
445.69, 448.97, 452.27, 455.59, 458.94, 462.30, 465.68, 469.08, 472.51, 475.95,
```

E.2 Supplementary Figures

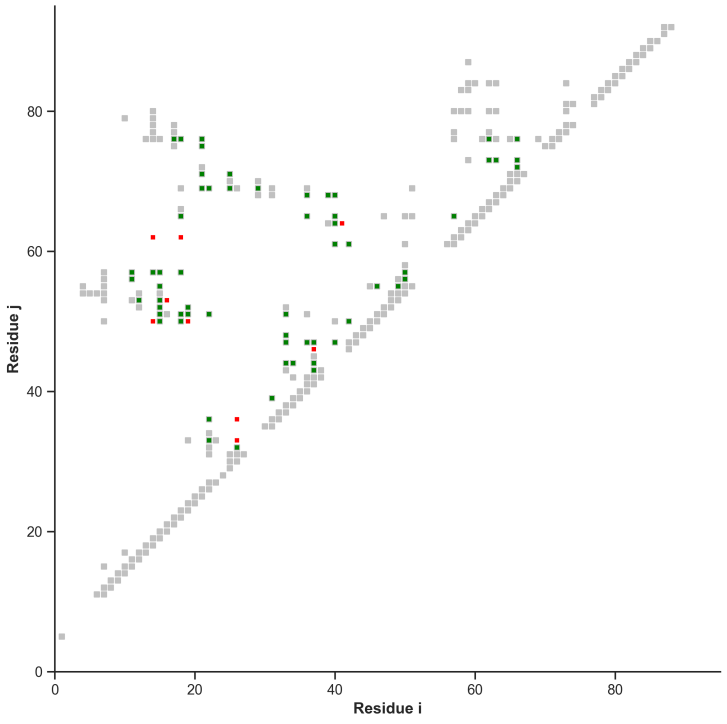

Figure E.1. Contact map of Lambda repressor (PDB id: 1lmb[210]). Displayed are the native contacts (gray), true-positive bias contacts (green) and false-positive bias contacts (red). Reproduced from Ref.[3] under CC BY 4.0.

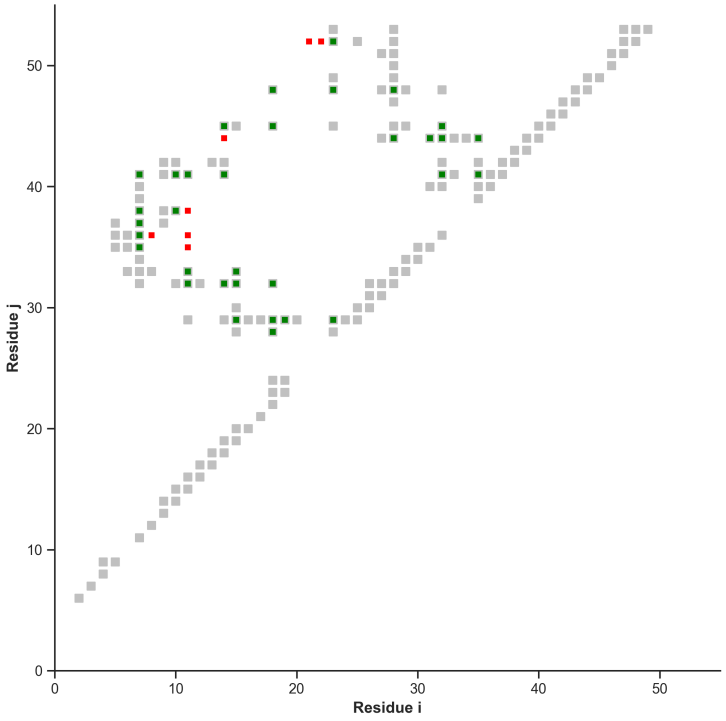

Figure E.2. Contact map of Albumin-binding domain (PDB id: 1prb[209]). Displayed are the native contacts (gray), true-positive bias contacts (green) and false-positive bias contacts (red). Reproduced from Ref.[3] under CC BY 4.0.

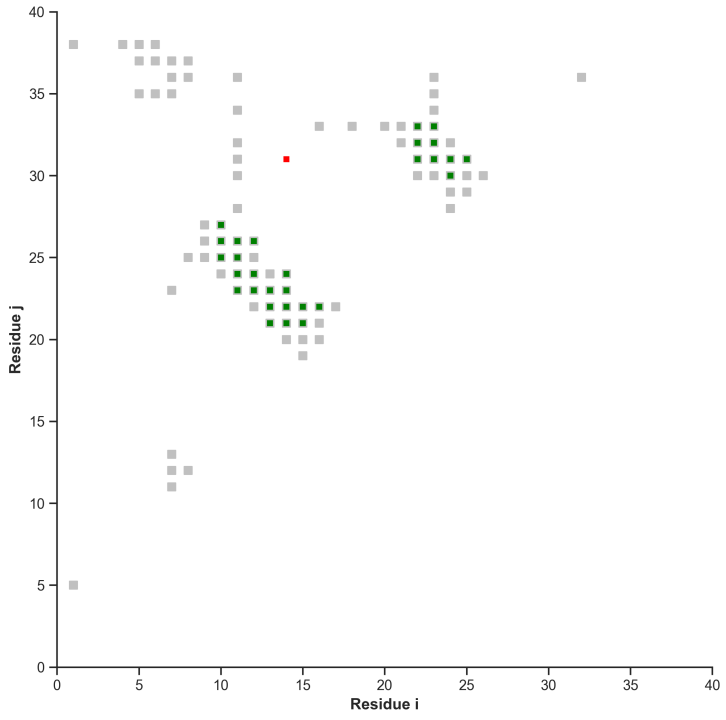

Figure E.3. Contact map of WW domain (PDB id: 2f21[212]). Displayed are the native contacts (gray), true-positive bias contacts (green) and false-positive bias contacts (red). Reproduced from Ref.[3] under CC BY 4.0.

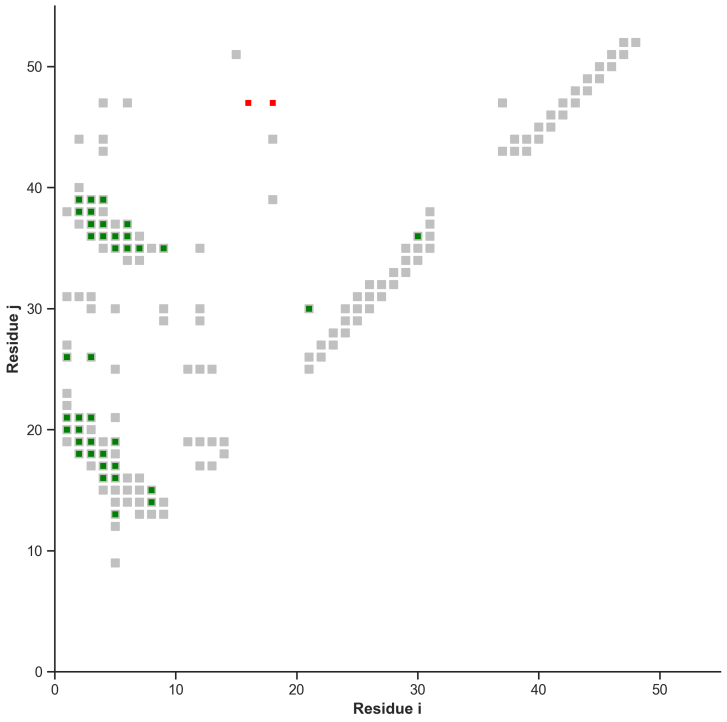

Figure E.4. Contact map of NTL9 (PDB id: 2hba[211]). Displayed are the native contacts (gray), true-positive bias contacts (green) and false-positive bias contacts (red). Reproduced from Ref.[3] under CC BY 4.0.

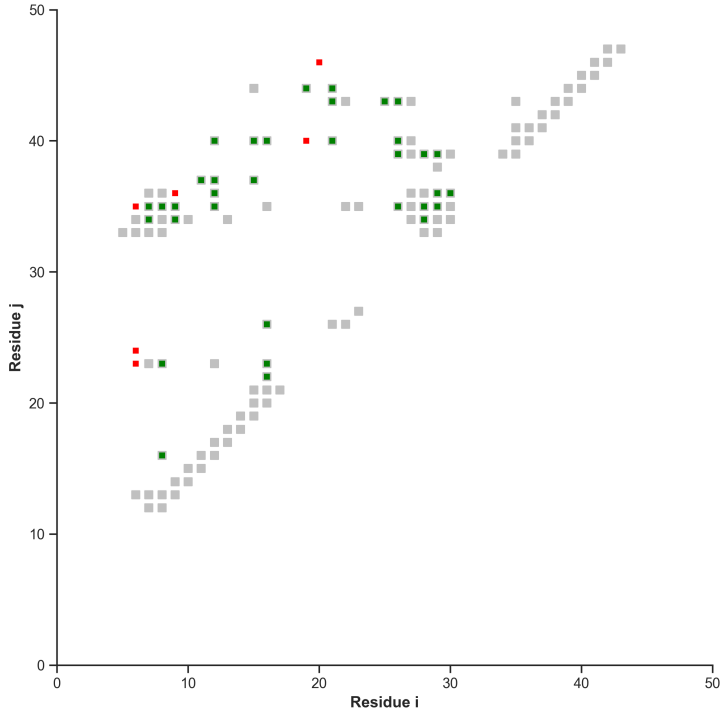

Figure E.5. Contact map of BBL (PDB id: 2wxc[208]). Displayed are the native contacts (gray), true-positive bias contacts (green) and false-positive bias contacts (red). Reproduced from Ref.[3] under CC BY 4.0.

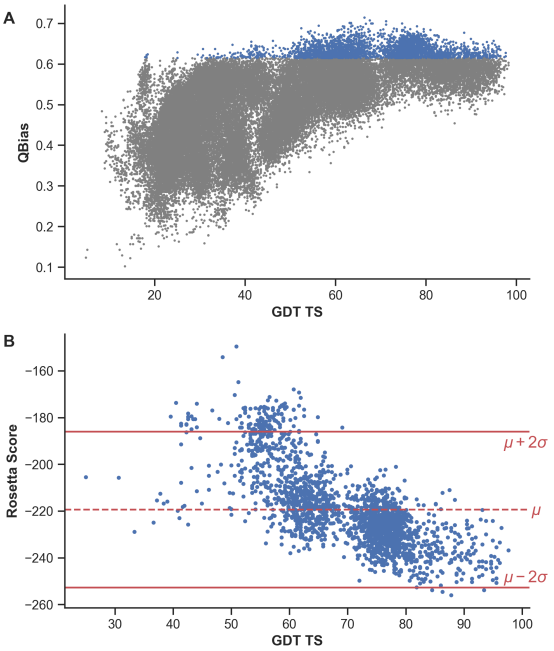

Figure E.6. Correlations of Lambda repressor (PDB id: 1lmb[210]) simulation. (A) Relation between Q_{Bias} (fraction of realized bias contacts) and GDT TS. Gray and blue colored dots represent the entire REX MD trajectory composed of 50000 structures. Blue dots highlight the 2000 structures with the highest Q_{Bias} values, which were pre-selected for the ensemble selection. **(B)** Relation between Rosetta score and GDT TS of the 2000 pre-selected structures. Figure also depicts the mean score μ (red dashed line) and $\mu \pm 2\sigma$ (red solid lines) which were used as thresholds to filter outliers during the cluster score calculations. Reproduced from Ref.[3] under CC BY 4.0.

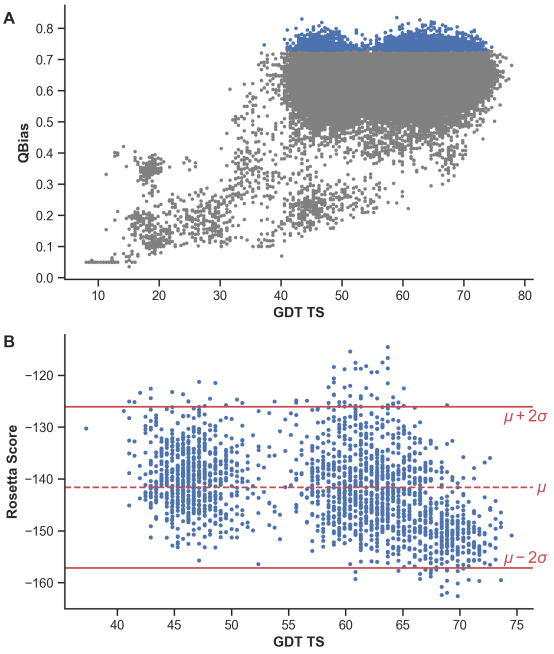

Figure E.7. Correlations of Albumin-binding domain (PDB id: 1prb[209]) simulation. (A) Relation between Q_{Bias} (fraction of realized bias contacts) and GDT TS. Gray and blue colored dots represent the entire REX MD trajectory composed of 50000 structures. Blue dots highlight the 2000 structures with the highest Q_{Bias} values, which were pre-selected for the ensemble selection. **(B)** Relation between Rosetta score and GDT TS of the 2000 pre-selected structures. Figure also depicts the mean score μ (red dashed line) and $\mu \pm 2\sigma$ (red solid lines) which were used as thresholds to filter outliers during the cluster score calculations. Reproduced from Ref.[3] under CC BY 4.0.

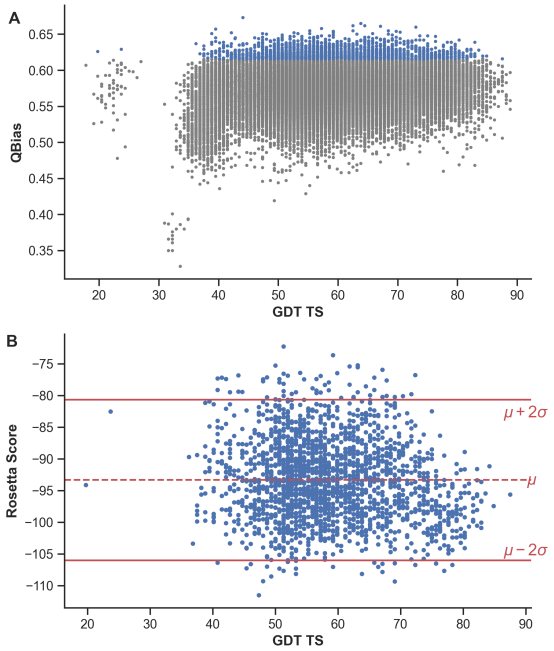

Figure E.8. Correlations of WW domain (PDB id: 2f21[212]) simulation. (A) Relation between Q_{Bias} (fraction of realized bias contacts) and GDT TS. Gray and blue colored dots represent the entire REX MD trajectory composed of 50000 structures. Blue dots highlight the 2000 structures with the highest Q_{Bias} values, which were pre-selected for the ensemble selection. **(B)** Relation between Rosetta score and GDT TS of the 2000 pre-selected structures. Figure also depicts the mean score μ (red dashed line) and $\mu \pm 2\sigma$ (red solid lines) which were used as thresholds to filter outliers during the cluster score calculations. Reproduced from Ref.[3] under CC BY 4.0.

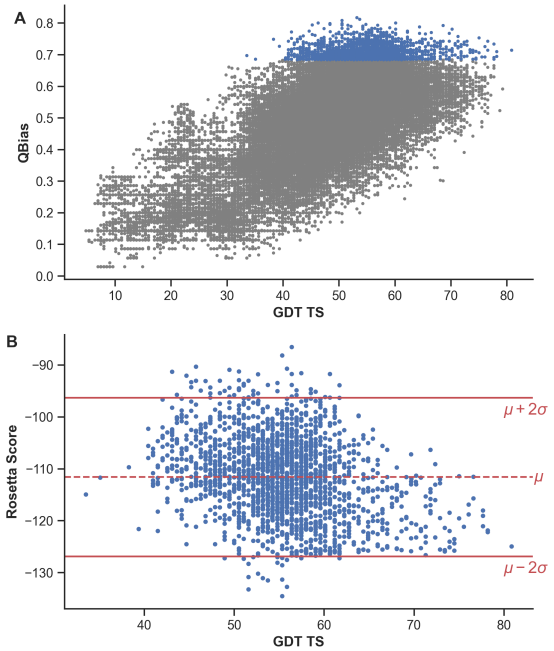

Figure E.9. Correlations of BBL (PDB id: 2wxc[208]) simulation. (A) Relation between Q_{Bias} (fraction of realized bias contacts) and GDT TS. Gray and blue colored dots represent the entire REX MD trajectory composed of 50000 structures. Blue dots highlight the 2000 structures with the highest Q_{Bias} values, which were pre-selected for the ensemble selection. **(B)** Relation between Rosetta score and GDT TS of the 2000 pre-selected structures. Figure also depicts the mean score μ (red dashed line) and $\mu \pm 2\sigma$ (red solid lines) which were used as thresholds to filter outliers during the cluster score calculations. Reproduced from Ref.[3] under CC BY 4.0.

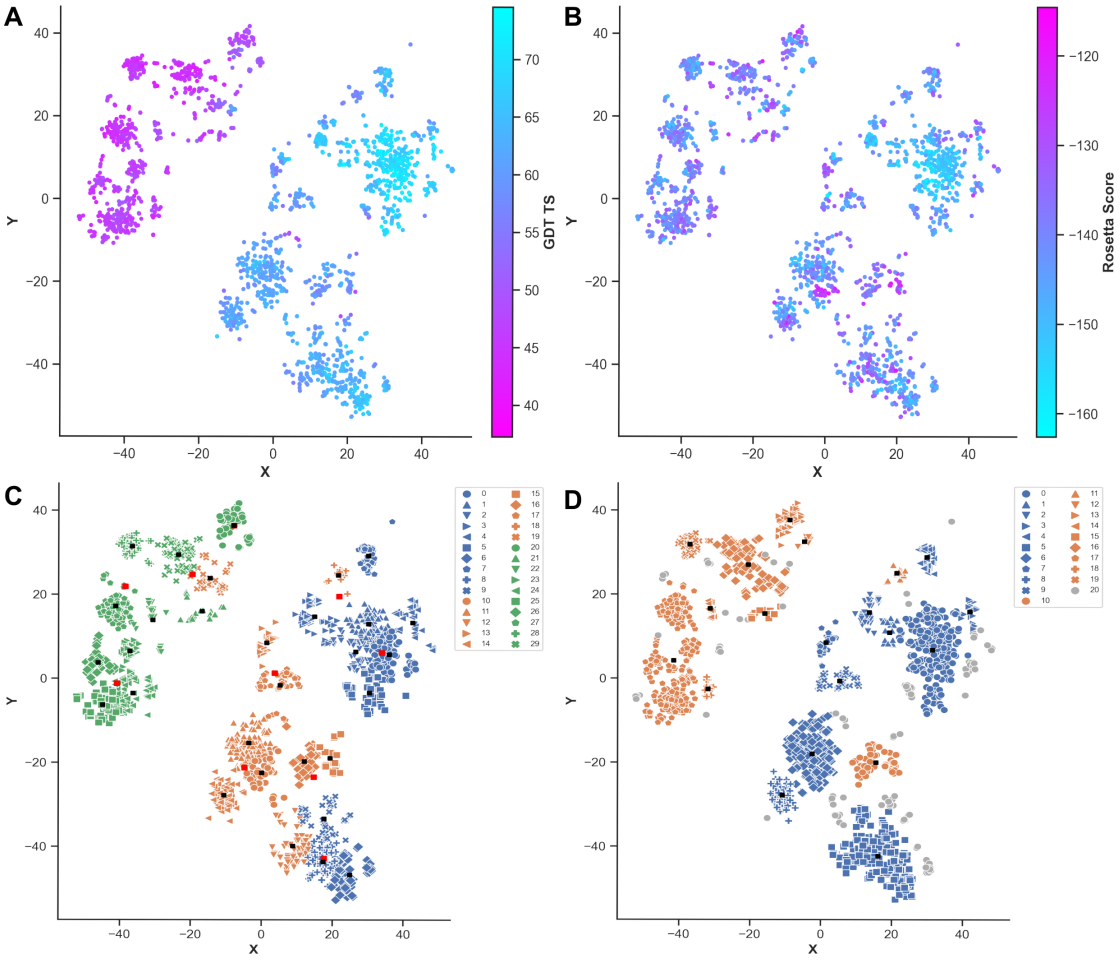

Figure E.10. TSNE representation of selected Albumin-binding domain (PDB id: 1prb[209] structures. (A) Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 3.5$, $min_{\text{pts}} = 20$). Adapted from Ref.[3] under CC BY 4.0.

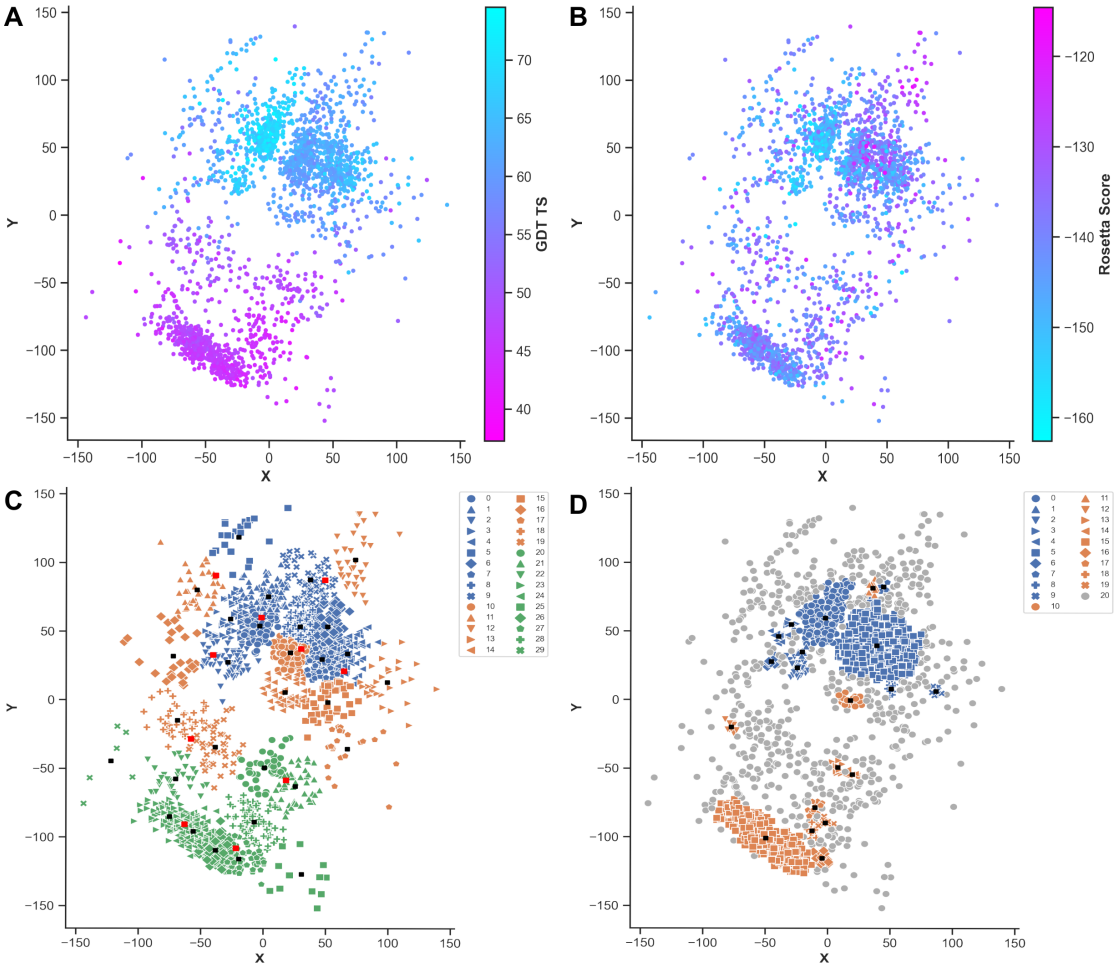

Figure E.11. MDS representation of selected Albumin-binding domain (PDB id: 1prb[209] structures. (A) Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 4.8$, $min_{\text{pts}} = 7$). Adapted from Ref.[3] under CC BY 4.0.

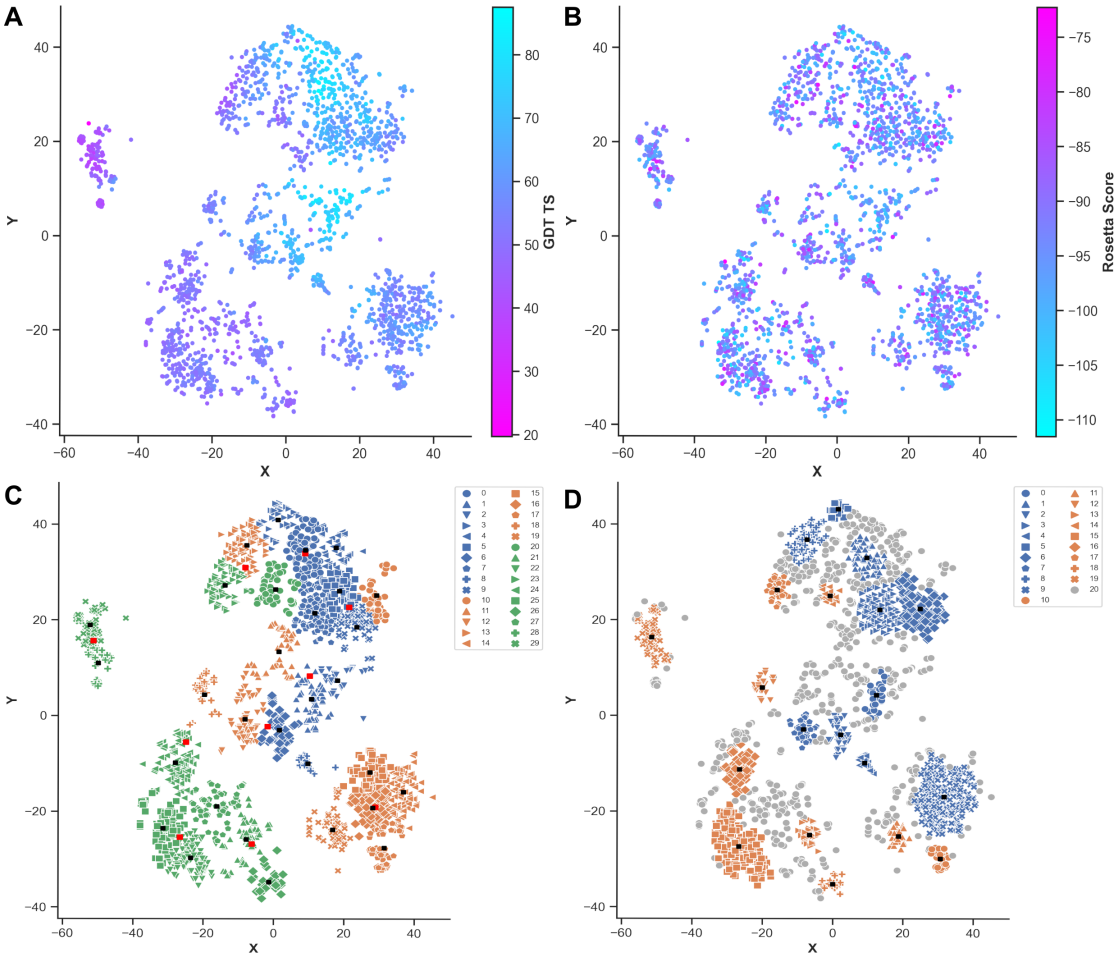

Figure E.12. TSNE representation of selected WW domain (PDB id: 2f21[212]) structures. (A) Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 2.65$, $min_{\text{pts}} = 20$). Adapted from Ref.[3] under CC BY 4.0.

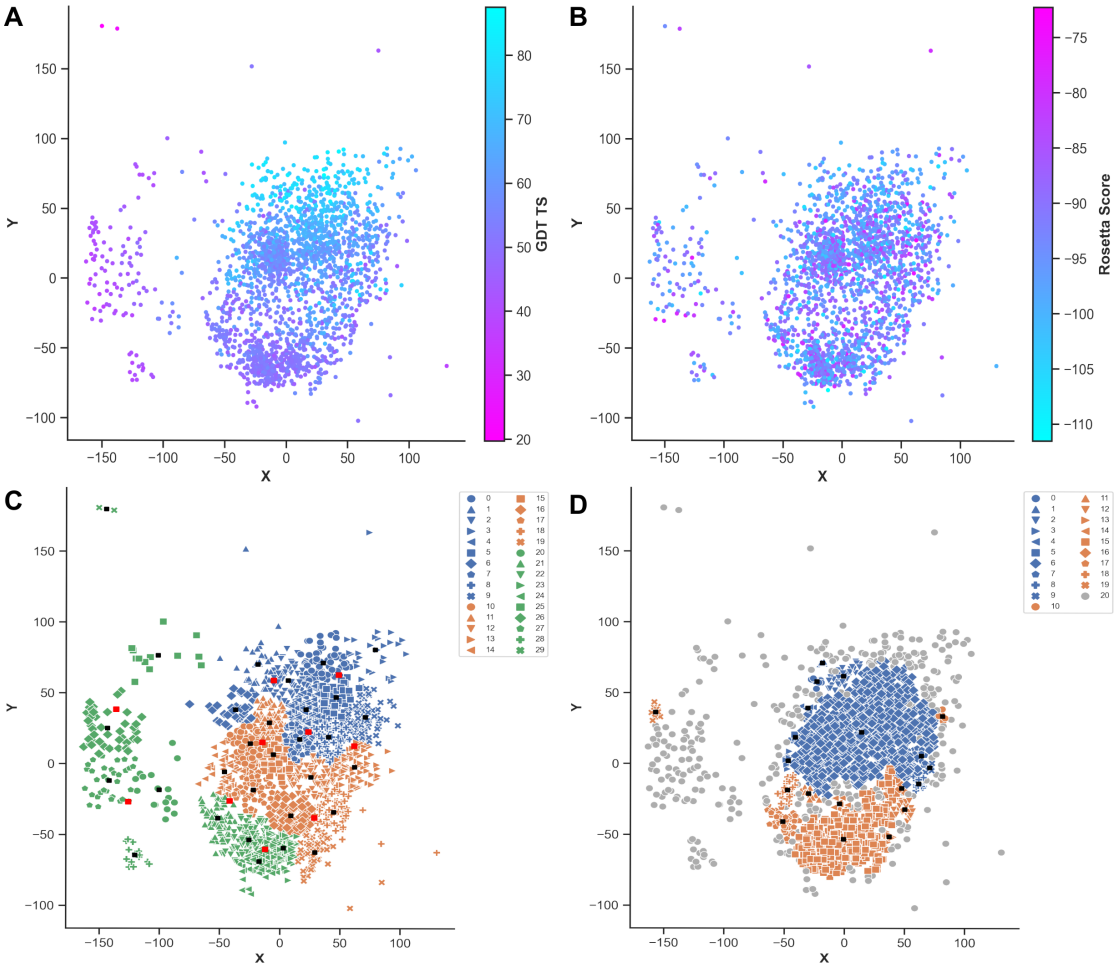

Figure E.13. MDS representation of selected WW domain (PDB id: 2f21[212]) structures. (A) Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 4.85$, $min_{\text{pts}} = 6$). Adapted from Ref.[3] under CC BY 4.0.

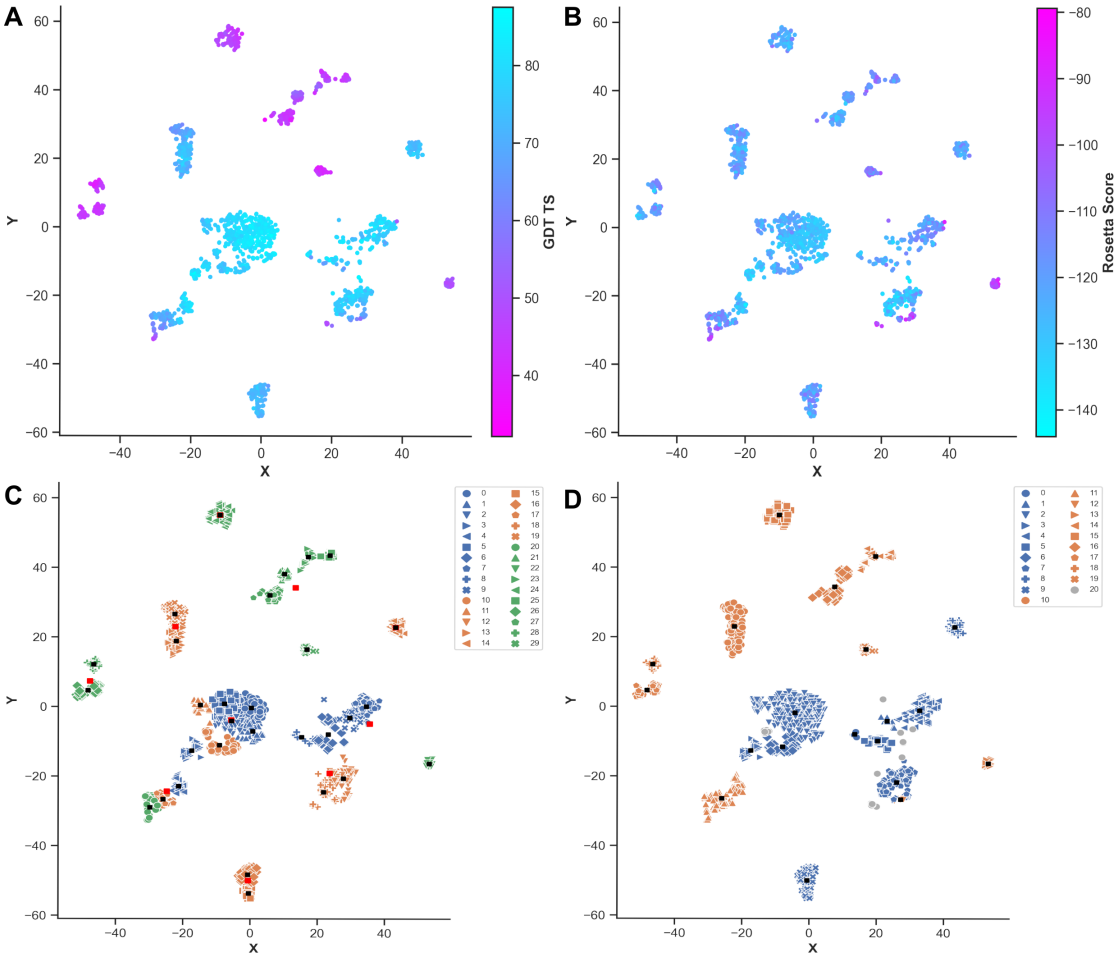

Figure E.14. TSNE representation of selected NTL9 (PDB id: 2hba[211]) structures. (A) Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 2.615$, $min_{\text{pts}} = 20$). Adapted from Ref.[3] under CC BY 4.0.

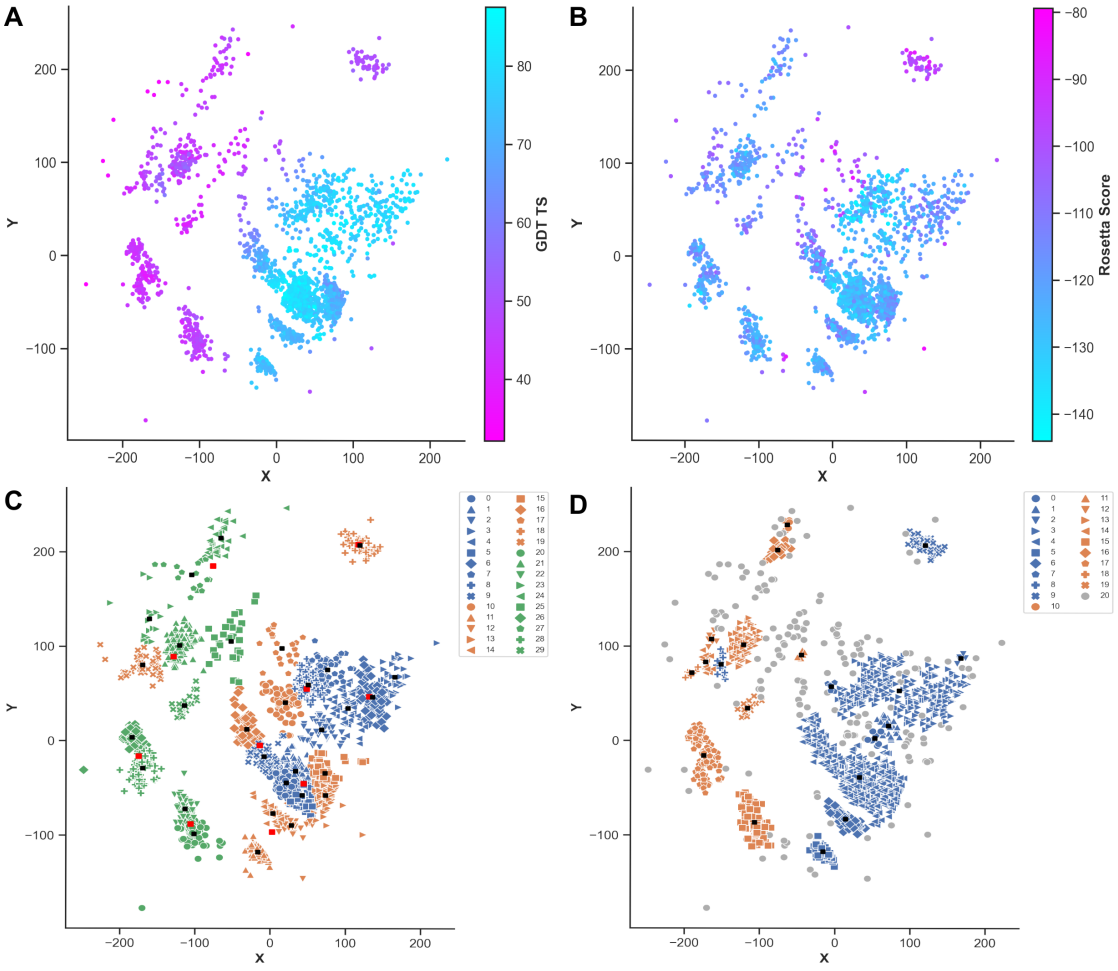

Figure E.15. MDS representation of selected NTL9 (PDB id: 2hba[211]) structures. (A) Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 8.35$, $min_{\text{pts}} = 5$). Adapted from Ref.[3] under CC BY 4.0.

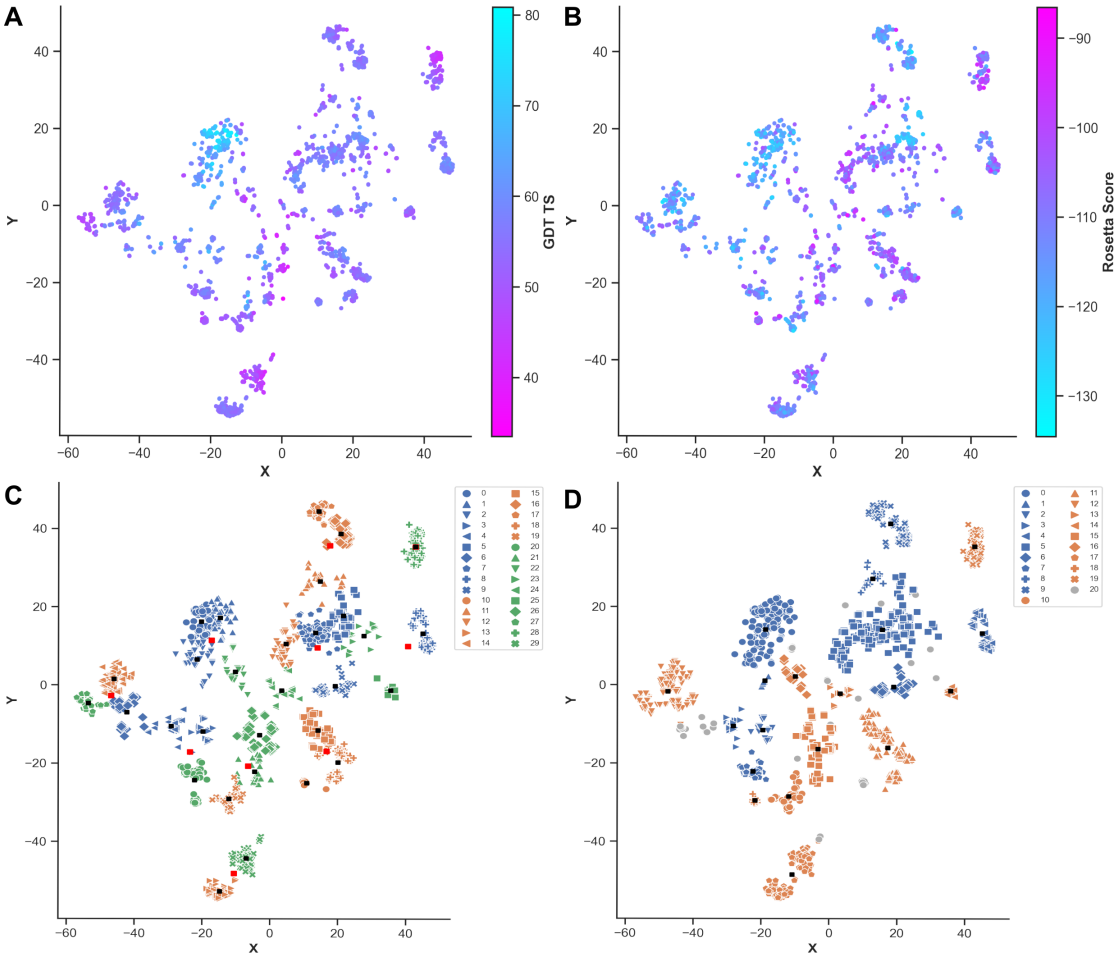

Figure E.16. TSNE representation of selected BBL (PDB id: 2wxc[208]) structures. (A) Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 3.835$, $min_{\text{pts}} = 20$). Adapted from Ref.[3] under CC BY 4.0.

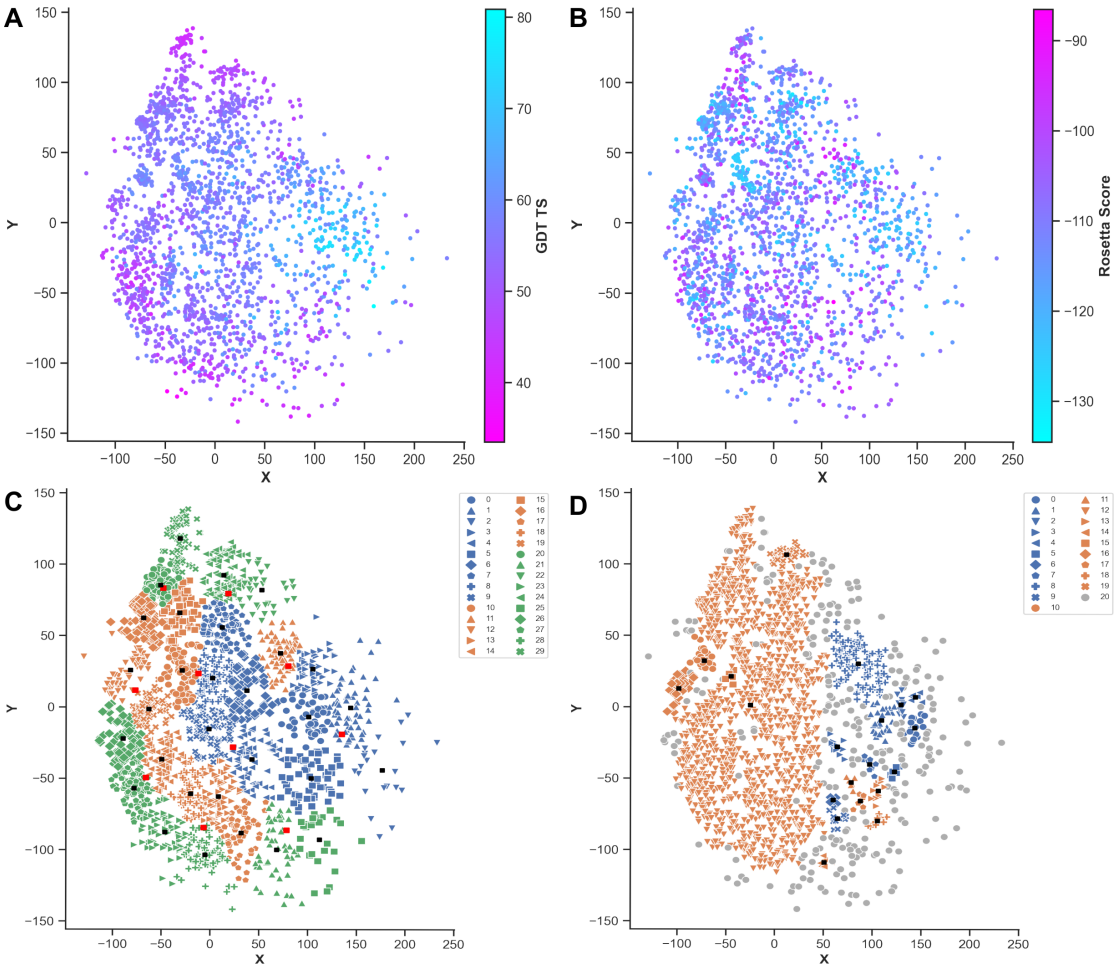

Figure E.17. MDS representation of selected BBL (PDB id: 2wxc[208]) structures. (A) Relation with refinement levels (GDT TS). **(B)** Relation with energy function (Rosetta scores). **(C)** Relation with KMEANS cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 29: worst). Cluster centers are visualized as black squares ($k = 30$) or red squares ($k = 10$). **(D)** Relation with DBSCAN cluster mapping. Cluster indices are ranked by average cluster accuracy (0: best, 19: worst, 20: noise). Cluster centers are visualized as black squares ($\varepsilon = 7.2$, $min_{\text{pts}} = 6$). Adapted from Ref.[3] under CC BY 4.0.

E.3 Supplementary Tables

Table E.1. Starting decoy accuracy of performed REX MD simulations[3]. Table shows the corresponding replica numbers, PDB ids of protein targets and global distance test total scores (GDT TS) before the simulation started.

	Decoy GDT TS of protein target						Decoy GDT TS of protein target				
replica	1lmb	1prb	2f21	2hba	2wxc	replica	1lmb	1prb	2f21	2hba	2wxc
1	32.22	63.74	39.57	30.44	52.94	37	24.12	52.72	31.29	14.98	14.97
2	31.47	51.66	38.91	17.27	35.03	38	19.00	59.84	33.94	14.61	35.29
3	34.54	65.17	39.90	25.00	25.14	39	32.36	39.22	36.75	26.69	12.30
4	19.68	67.77	38.91	19.44	35.70	40	16.56	30.92	37.92	7.85	17.24
5	14.64	67.06	47.18	17.88	41.31	41	18.05	27.02	36.76	19.57	16.44
6	9.81	55.69	39.40	17.88	50.67	42	22.82	47.28	47.68	8.94	6.15
7	10.90	61.73	50.83	21.74	17.65	43	11.78	42.54	41.56	16.06	34.22
8	22.68	59.72	33.94	26.57	52.00	44	21.87	34.83	13.74	17.87	29.01
9	53.88	63.86	39.07	17.76	48.66	45	15.33	21.09	40.40	15.70	37.03
10	5.72	56.99	51.16	13.41	22.86	46	15.87	63.39	35.43	12.32	31.15
11	15.60	52.73	35.76	14.25	11.76	47	18.80	44.20	32.12	13.16	18.05
12	15.67	26.66	33.77	17.15	43.58	48	29.63	42.65	20.70	19.20	16.04
13	25.75	39.46	38.74	13.16	20.19	49	31.81	55.33	35.93	15.34	23.26
14	38.22	45.14	30.63	21.38	40.24	50	16.83	60.9	32.78	13.28	38.24
15	10.15	46.56	50.83	17.51	35.70	51	12.94	22.63	45.03	12.08	19.25
16	15.33	67.89	77.98	14.01	39.31	52	5.38	55.21	35.76	14.01	39.17
17	11.99	51.89	39.90	18.72	53.61	53	25.68	43.01	36.59	17.88	39.97
18	14.03	65.29	37.58	13.53	42.64	54	5.04	33.65	28.64	15.34	21.39
19	11.65	36.38	35.10	29.59	29.68	55	20.50	45.50	35.92	11.96	27.27
20	6.95	52.72	32.12	14.50	33.56	56	8.93	52.02	49.83	22.70	44.12
21	43.94	10.43	34.44	22.22	25.40	57	5.66	46.33	33.61	11.60	45.19
22	21.32	60.19	40.56	14.01	43.98	58	13.76	60.78	32.45	11.47	27.14
23	30.45	63.86	41.72	13.28	39.57	59	32.15	39.10	34.94	16.55	14.04
24	10.96	44.67	60.27	19.80	13.37	60	24.86	60.90	41.06	15.22	32.22
25	19.76	44.31	36.42	16.79	25.67	61	21.05	61.85	42.05	33.58	17.11
26	26.50	62.80	32.28	20.89	25.54	62	17.85	44.19	45.53	16.79	48.80
27	12.94	56.99	35.93	14.25	47.46	63	11.31	45.50	42.38	19.20	41.98
28	26.36	58.88	43.38	12.92	39.17	64	15.32	36.49	44.70	22.34	25.80
29	0.27	27.96	35.43	39.85	30.88	65	17.64	10.31	28.64	11.36	13.37
30	14.24	52.84	45.04	17.51	34.49	66	11.31	50.36	37.25	9.06	23.53
31	11.24	38.51	37.58	24.04	23.53	67	24.38	8.65	42.38	31.88	24.20
32	24.45	53.67	12.08	23.91	47.73	68	10.63	35.78	36.92	19.32	33.83
33	11.31	16.71	48.18	19.68	39.70	69	20.78	51.54	33.11	27.66	9.49
34	26.98	68.72	41.06	13.28	39.70	70	15.12	25.36	38.08	8.21	35.16
35	16.28	49.17	38.91	27.06	33.29	71	5.31	27.84	42.05	26.69	24.33
36	11.58	37.44	36.76	13.89	35.16	72	10.42	14.93	31.79	16.30	39.04

Table E.2. Starting decoy accuracy of performed REX MD simulations[3]. Table shows the corresponding replica numbers, PDB ids of protein targets and backbone root-mean-square-deviation (RMSD) before the simulation started.

	Decoy RMSD (Å) of protein target						Decoy RMSD (Å) of protein target				
replica	1lmb	1prb	2f21	2hba	2wxc	replica	1lmb	1prb	2f21	2hba	2wxc
1	7.08	2.95	5.84	7.68	4.03	37	8.09	4.33	8.20	9.82	9.48
2	9.96	3.67	6.48	8.02	5.56	38	7.87	2.73	6.80	11.26	5.43
3	5.86	2.70	6.27	9.22	7.05	39	7.23	5.61	6.14	7.14	11.58
4	10.23	2.59	6.34	8.27	5.43	40	11.17	6.66	5.61	11.82	9.13
5	10.68	2.62	4.90	9.04	5.21	41	11.55	8.27	7.34	7.73	8.39
6	12.31	3.60	5.85	8.87	4.21	42	10.62	4.29	5.36	11.05	11.91
7	12.26	3.10	4.99	8.86	9.14	43	11.96	4.41	5.07	8.64	6.30
8	8.85	2.75	6.70	8.28	4.06	44	8.50	7.04	10.34	8.73	6.46
9	3.51	2.59	6.01	8.24	4.45	45	11.07	7.62	4.95	9.70	5.67
10	13.86	3.23	5.18	9.61	10.62	46	10.84	2.62	5.48	9.78	5.51
11	12.97	3.63	6.84	9.65	10.27	47	9.62	5.71	6.85	11.24	9.27
12	12.76	7.00	6.95	10.58	4.53	48	9.04	5.59	7.44	10.20	9.08
13	8.60	6.45	6.63	9.74	7.91	49	7.83	3.42	5.79	9.57	7.41
14	5.25	4.22	7.95	8.51	5.23	50	11.49	3.18	7.18	10.83	5.32
15	11.38	4.03	4.41	9.35	5.15	51	12.98	7.89	5.95	9.63	8.05
16	10.39	2.55	2.15	9.73	5.02	52	12.83	3.49	7.49	10.20	6.07
17	11.62	3.41	6.75	10.18	4.41	53	11.55	4.52	6.61	8.43	4.88
18	10.55	2.61	6.28	10.46	4.43	54	14.19	6.23	7.36	9.19	8.19
19	12.54	5.43	7.50	7.34	6.40	55	10.89	4.44	6.30	10.62	7.04
20	13.03	3.94	6.64	9.49	6.16	56	13.55	3.98	4.32	8.92	5.42
21	4.89	11.07	6.29	8.10	7.10	57	14.09	3.79	5.75	10.69	4.85
22	9.32	2.80	5.91	11.02	4.95	58	14.03	3.02	6.75	10.15	6.70
23	7.22	2.71	5.12	9.10	5.88	59	6.89	5.69	6.28	8.55	9.75
24	12.96	4.14	3.07	9.60	10.30	60	11.25	2.99	6.43	10.71	6.51
25	11.84	4.09	5.70	8.90	6.72	61	9.04	3.05	5.06	6.74	8.70
26	7.60	2.93	7.07	8.02	6.89	62	10.66	5.74	5.47	9.90	4.10
27	11.26	3.31	6.32	9.20	3.89	63	12.30	4.95	5.60	9.62	5.18
28	7.78	2.81	4.63	10.06	5.03	64	11.88	5.17	4.36	8.49	6.76
29	17.37	5.74	6.12	4.67	6.00	65	8.98	10.29	7.62	9.37	10.42
30	12.32	4.01	5.49	8.90	5.64	66	12.49	4.64	5.64	10.97	6.89
31	12.24	5.17	5.30	9.98	7.42	67	10.02	9.96	6.12	8.68	7.66
32	10.98	3.86	10.14	7.27	4.70	68	13.54	6.58	6.20	7.50	5.35
33	12.23	10.42	5.24	8.79	4.94	69	8.90	3.65	7.39	7.46	11.19
34	10.54	2.61	4.82	10.64	4.61	70	10.19	8.69	5.61	10.34	5.46
35	9.35	3.81	5.53	7.10	5.76	71	12.94	5.80	6.11	8.06	8.15
36	13.43	6.59	7.11	9.86	6.45	72	12.49	10.25	7.69	9.05	5.43

Table E.3. Used bias contacts during REX MD simulations[3]. Table contains the corresponding contact numbers, PDB ids of protein targets and residue pairs (resi, resj).

contact	1lmb		1prb		2f21		2hba		2wxc		contact	1lmb	
number	resi	resj	resi	resj	resi	resj	resi	resj	resi	resj	number	resi	resj
1	37	47	15	32	11	23	5	36	15	40	41	57	65
2	36	47	35	41	13	23	4	37	12	40	42	25	71
3	33	47	14	32	15	21	3	36	16	40	43	29	69
4	37	44	11	32	12	24	4	36	19	44	44	46	55
5	18	50	32	41	11	25	3	19	26	43	45	18	62
6	19	51	18	28	14	22	2	18	12	35	46	15	57
7	39	68	15	29	10	26	3	37	28	35	47	40	61
8	34	44	32	44	13	21	4	18	12	37	48	18	65
9	40	68	18	32	11	24	2	20	29	36	49	31	39
10	15	50	11	35	24	31	4	39	15	37	50	17	76
11	37	43	35	44	12	26	3	38	21	43	51	26	32
12	66	73	18	29	14	21	6	35	29	39	52	50	56
13	15	53	11	41	12	23	2	39	16	23	53	22	69
14	40	65	23	29	14	24	3	18	8	23	54	26	33
15	50	57	15	33	13	22	3	21	11	37	55	11	56
16	33	44	31	44	23	31	1	21	26	39	56	41	64
17	40	64	28	48	22	33	5	35	29	35	57	21	69
18	49	55	11	33	22	31	30	36	21	44	58	18	57
19	25	69	14	41	11	26	6	37	26	35	59	11	57
20	66	72	23	48	10	25	6	36	8	35	60	22	33
21	66	76	7	38	24	30	2	19	12	36	61	12	53
22	15	51	7	35	16	22	7	35	21	40	62	21	75
23	62	73	18	48	10	27	3	26	26	40	63	19	50
24	33	48	11	36	25	31	3	39	25	43	64	14	50
25	63	73	14	45	22	32	8	14	28	39	65	14	62
26	18	51	7	41	23	32	5	13	6	23	66	36	65
27	22	51	10	41	14	31	2	38	8	16	67	15	55
28	37	46	7	36	15	22	1	20	16	22	68	42	50
29	62	76	32	45	23	33	5	17	20	46	69	16	53
30	26	36	7	37	14	23	4	17	28	34	70	19	52
31	40	47	28	44			5	16	19	40			
32	36	68	14	44			5	19	30	36			
33	42	61	18	45			1	26	9	35			
34	14	57	19	29			16	47	9	34			
35	21	76	23	52			8	15	16	26			
36	18	76	21	52			9	35					
37	21	71	10	38			21	30					
38	15	52	11	38			2	21					
39	22	36	22	52			4	16					
40	33	51	8	36			18	47					

Table E.4. Cluster accuracy of selected clusters during different algorithm chains[3]. Table contains the PDB id, applied algorithm chain, cluster labels, cluster size, as well as GDT (global distance test) and RMSD (root-mean-square-deviation) statistics. The four selected clusters are listed in picking order based on mean Rosetta scores. Cluster labels are ranked by accuracy (0: best).

PDB id	algorithm chain	cluster	size	GDT_{mean}	GDT_{min}	GDT_{max}	$RMSD_{mean}$	$RMSD_{min}$	$RMSD_{max}$
1lmb	TSNE → KMEANS	1	95	82.52	76.49	87.80	2.76	2.35	3.06
		0	123	88.71	71.73	97.62	1.39	0.83	2.46
		6	86	76.52	71.13	80.36	2.79	2.54	3.00
		2	18	81.40	73.81	86.02	2.38	2.09	2.66
1lmb	MDS → KMEANS	2	74	82.74	76.19	87.80	2.66	2.09	3.06
		0	66	89.05	43.16	96.43	1.41	0.83	5.95
		1	61	85.43	60.42	97.62	1.60	0.93	3.69
		6	115	76.28	71.13	80.66	2.80	2.62	2.99
1lmb	TSNE → DBSCAN	1	98	82.45	73.81	87.80	2.75	2.23	3.06
		0	123	88.71	71.73	97.62	1.39	0.83	2.46
		2	230	77.15	71.13	82.44	2.78	2.54	3.04
		4	31	76.70	72.92	80.66	2.90	2.78	3.08
1lmb	MDS → DBSCAN	2	90	81.93	64.88	87.80	2.71	2.09	3.21
		0	81	89.59	71.73	97.62	1.30	0.83	2.32
		3	7	77.59	70.24	83.93	2.88	2.63	3.02
		1	5	87.80	81.85	94.64	1.44	1.21	1.85
1prb	TSNE → KMEANS	2	60	67.63	62.73	72.17	2.41	2.18	2.84
		0	98	69.56	61.79	74.53	2.29	2.02	2.94
		1	81	69.05	64.62	73.59	2.28	1.95	2.60
		3	77	65.60	57.08	73.58	2.65	2.20	3.59
1prb	MDS → KMEANS	0	183	68.89	61.79	74.53	2.32	2.02	2.62
		2	72	65.30	55.66	73.58	2.68	2.20	3.50
		1	93	68.13	57.55	73.11	2.39	1.95	3.38
		10	135	61.31	56.13	66.98	2.84	2.47	3.38
1prb	TSNE → DBSCAN	1	23	67.51	60.85	73.58	2.49	2.19	3.20
		0	271	68.76	61.32	74.53	2.35	1.95	3.03
		2	59	64.99	57.08	71.22	2.69	2.2	3.59
		14	21	48.61	45.28	50.95	4.98	4.75	5.24
1prb	MDS → DBSCAN	2	34	66.80	58.02	71.22	2.51	2.20	2.86
		0	275	68.82	59.91	74.53	2.33	1.95	3.14
		1	7	67.25	62.73	69.81	2.42	2.24	2.69
		12	10	50.47	48.58	52.36	4.25	4.09	4.38

Table E.5. Cluster accuracy of selected clusters during different algorithm chains[3]. Table contains the PDB id, applied algorithm chain, cluster labels, cluster size, as well as GDT (global distance test) and RMSD (root-mean-square-deviation) statistics. The four selected clusters are listed in picking order based on mean Rosetta scores. Cluster labels are ranked by accuracy (0: best).

PDB id	algorithm chain	cluster	size	GDT_{mean}	GDT_{min}	GDT_{max}	$RMSD_{mean}$	$RMSD_{min}$	$RMSD_{max}$
2f21	TSNE → KMEANS	1	69	73.97	42.1	82.90	2.71	1.96	4.95
		0	85	74.85	56.58	87.50	2.76	1.88	3.67
		2	34	73.10	48.68	82.24	2.85	2.07	5.11
		3	47	68.02	42.77	80.26	3.25	2.33	4.28
2f21	MDS → KMEANS	1	38	74.57	50.66	82.89	2.58	1.96	4.26
		0	62	75.49	61.84	87.50	2.67	1.88	3.46
		3	28	68.19	42.77	80.26	3.21	2.44	4.28
		2	63	71.83	47.37	82.89	2.87	2.07	4.56
2f21	TSNE → DBSCAN	0	41	76.52	66.44	82.24	2.56	2.10	3.04
		1	61	75.67	61.84	87.50	2.72	1.88	3.39
		6	108	63.29	51.97	77.63	3.55	2.69	4.35
		2	48	67.71	55.26	75.00	3.50	3.03	4.31
2f21	MDS → DBSCAN	0	9	76.53	71.71	78.95	2.51	2.07	2.89
		1	6	76.42	73.68	81.58	2.49	2.29	2.71
		8	12	58.88	43.42	82.90	4.11	2.44	5.50
		2	32	72.76	48.68	82.89	2.77	2.07	4.45
2hba	TSNE → KMEANS	3	56	80.28	73.81	87.50	1.92	1.58	2.23
		2	79	81.34	74.41	85.12	1.81	1.65	2.06
		0	107	82.40	76.78	86.91	1.76	1.55	1.95
		1	54	82.34	77.98	86.31	1.74	1.56	1.96
2hba	MDS → KMEANS	0	170	81.63	75.59	86.31	1.79	1.52	2.11
		1	175	81.01	72.62	86.91	1.83	1.57	2.49
		7	52	77.33	66.67	84.52	2.04	1.63	2.95
		8	72	76.75	67.86	84.52	2.09	1.75	2.65
2hba	TSNE → DBSCAN	3	56	80.28	73.81	87.50	1.92	1.58	2.23
		0	19	82.36	64.29	86.91	1.67	1.49	2.28
		1	21	81.26	77.38	85.12	1.79	1.55	2.03
		7	169	75.74	61.31	83.33	2.17	1.58	3.20
2hba	MDS → DBSCAN	1	31	80.57	73.22	85.71	1.77	1.61	2.03
		4	823	77.00	58.93	87.50	2.03	1.52	3.32
		0	8	82.51	78.57	85.71	1.69	1.49	1.88
		3	305	77.55	64.28	87.50	2.00	1.45	2.81
2wxc	TSNE → KMEANS	5	92	58.27	51.59	64.36	4.31	4.02	4.76
		0	67	67.86	56.91	76.59	2.55	1.74	3.34
		2	77	62.43	46.81	73.94	3.09	2.06	4.98
		1	78	64.72	44.15	80.85	2.81	1.75	4.54
2wxc	MDS → KMEANS	1	62	65.63	44.15	78.19	2.71	1.75	4.54
		10	96	57.53	47.34	64.36	4.36	4.02	4.92
		0	45	68.16	51.06	76.59	2.50	1.74	4.24
		2	22	62.65	47.87	80.85	3.10	2.00	4.20
2wxc	TSNE → DBSCAN	1	16	64.59	61.17	69.15	2.77	2.58	3.17
		0	206	64.90	44.15	80.85	2.83	1.74	4.98
		3	42	58.45	51.59	68.08	3.41	2.92	4.05
		9	126	54.51	45.22	58.51	5.08	4.53	6.02
2wxc	MDS → DBSCAN	2	10	66.54	56.91	73.40	2.57	2.03	3.38
		3	12	64.50	59.58	69.15	2.78	2.32	3.37
		0	17	70.09	53.73	78.19	2.32	1.75	4.11
		1	27	69.96	57.98	76.59	2.33	1.74	3.40

F

Supplementary Information: Outlook to RNA REX

The following content shows the results of my initial testings using contact-guided REX MD with RNA targets. It serves as an outlook to possible future applications for RNA targets. The conducted study is very similar to the comparison of MD vs. REX MD, as shown in my bias-quality study[1] in section 4.1. However, note that the applied bias potential was not optimized for RNA targets. Thus it should be expected that the general performance and results can be improved even further.

Supplementary figures are provided in appendix F.1, supplementary tables in appendix F.2, and the used temperature distribution in appendix F.3

Each REX simulation had 60 replicas and a total of 20 unique starting conformations, which were assigned via cyclic permutation. Time steps were set to 2 fs and exchange attempts each 500 MD steps. Exchange rates were in the order of $10 - 15\%$. The all-atom simulations of this study utilized the OL15 nucleic force field[239,240] and TIP3P[119,120] explicit water model.

Tested RNA targets were the TPP riboswitch (PDB id: 3d2g[241]), Adenine riboswitch (PDB id: 4tzx[242]), and the 3',3'-cGAMP riboswitch (PDB id: 4yaz[243]). Table F.1 summarizes the lowest observed RMSD values with respect to the native fold. Figs. F.1 to F.3 show a comparison of MD and REX MD simulations. Biased REX simulations applied a total of approx. L native contacts (L: sequence length), as depicted in the contact maps of Figs. F.4 to F.6.

In the case of TPP riboswitch (cf. Fig. F.1) RMSD values reached approx. 5 Å during my simulations. Contact-guided REX MD heavily outperformed all other methods, whereas normal REX reached values of 7.4 Å and biased SimRNA with mean field DCA only 8.5 Å. In the case of the second RNA target, i.e. Adenine riboswitch (cf. Fig. F.2), normal REX achieved 3.6 Å and performed slightly better than the biased variant with 4.0 Å. SimRNA + mean field DCA yielded the best results with an RMSD of 3.0 Å. I want to emphasize that the applied force field seems to be reasonably accurate and reliable, which is reflected in the good RMSD statistics of the performed MD simulations for both TPP riboswitch and Adenine riboswitch. Furthermore, the Adenine riboswitch seems to be very flexible by nature. The attractive force resulting from my applied bias potential was too strong and resulted in a slight bending of the RNA structure, which is reflected in the higher RMSD values as compared to the unbiased case. Lastly, the comparison of the 3',3'-cGAMP riboswitch simulations are shown in Fig. F.3. My performed simulations yield the best results for the biased REX case, yet again. I want to stress that the overall "poor" performance is probably related to the fact that the 3',3'-cGAMP riboswitch is a dimer but the simulations used only one molecule chain. Due to the missing counterpart and their physical interactions, the simulations are likely to adopt other structures than the native one. For this reason, all methods achieved relatively high RMSD values. While the biased REX simulations yielded values of 15.8 Å, SimRNA + CoCoNet was capable to achieve 14.0 Å.

Overall, the application of contact-guided REX MD seems to have high potential and should be studied further. Note that my simulations applied a bias potential which was optimized for proteins and not RNAs. Hence, this method should yield even better results once the bias potential has been adjusted. I strongly suggest to perform studies similar to sections 4.1 and 4.2 and fine tune the bias potential specifically for RNA targets. Nevertheless, contact-guided REX MD produced already good results which are on par with the best performances of the other mentioned methods, as shown in Table F.1.

Table F.1. Lowest observed RMSD values of performed RNA simulations. Listed are the method, PDB ids and backbone RMSDs with respect to the native fold. Values are obtained via own simulations (upper half) or other methods (lower half, obtained from SI of Ref.[235]). Best cases are highlighted for each RNA target. †: Cases with L bias contacts (L: sequence length).

method	Backbone RMSD (Å)		
	3d2g	4tzx	4yaz
MD ref	7.6	3.8	20.4
REX ref	7.4	3.6	20.3
REX biased†	5.0	4.0	15.8
SimRNA	15.7	7.5	16.1
SimRNA + mfDCA†	8.5	3.0	20.1
SimRNA + CoCoNet†	16.2	7.8	14.0

F.1 Supplementary Figures

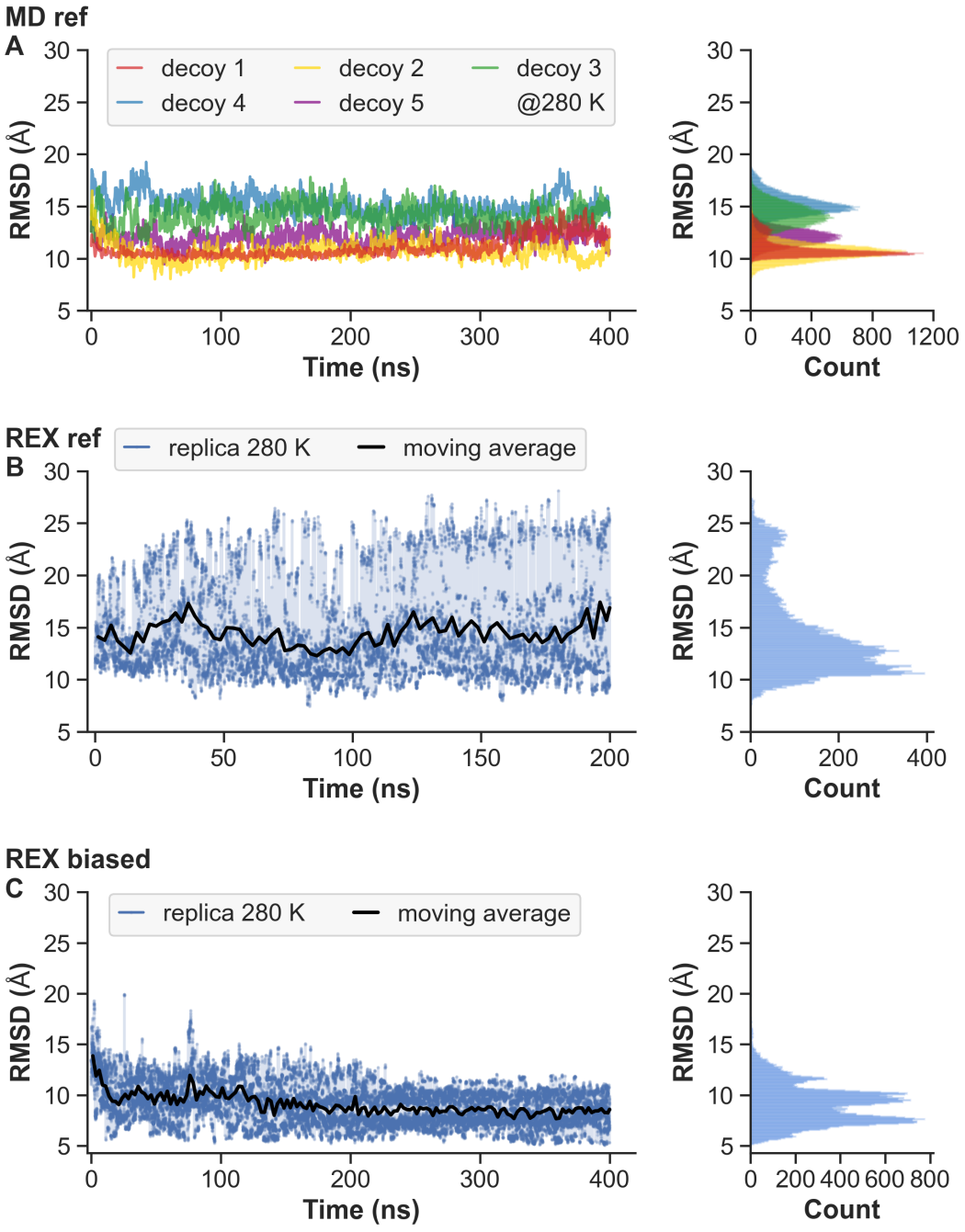

Figure F.1. RMSD comparison of simulations with TPP riboswitch (PDB id: 3d2g[241]). Backbone RMSD of MD trajectories at 280 K. MD simulations were performed with five unique starting conformations (*decoys*). **(A)** MD without bias. **(B)** REX MD without bias. **(C)** REX MD with bias.

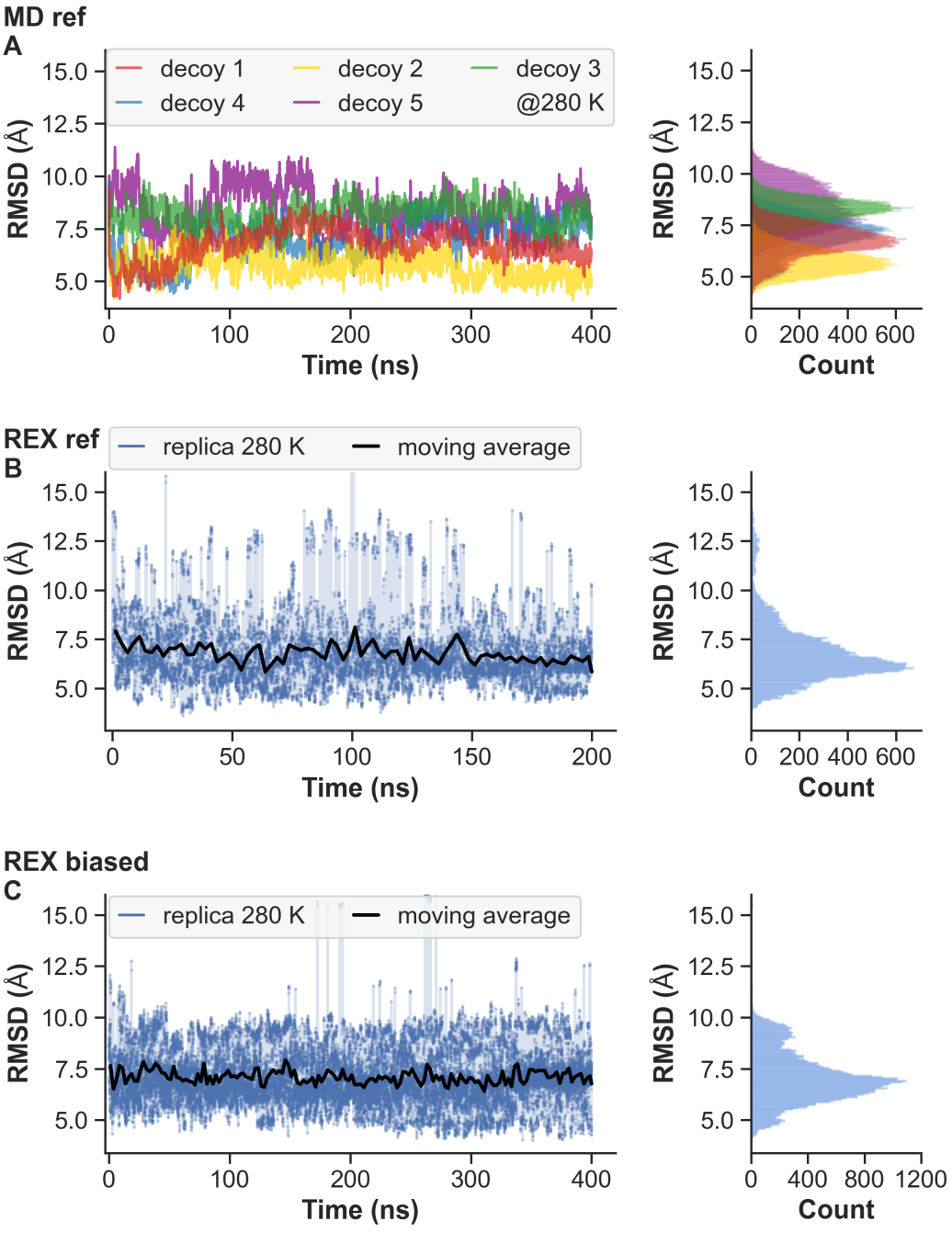

Figure F.2. RMSD comparison of simulations with Adenine riboswitch (PDB id: 4tzx[242]). Backbone RMSD of MD trajectories at 280 K. MD simulations were performed with five unique starting conformations (*decoys*). **(A)** MD without bias. **(B)** REX MD without bias. **(C)** REX MD with bias.

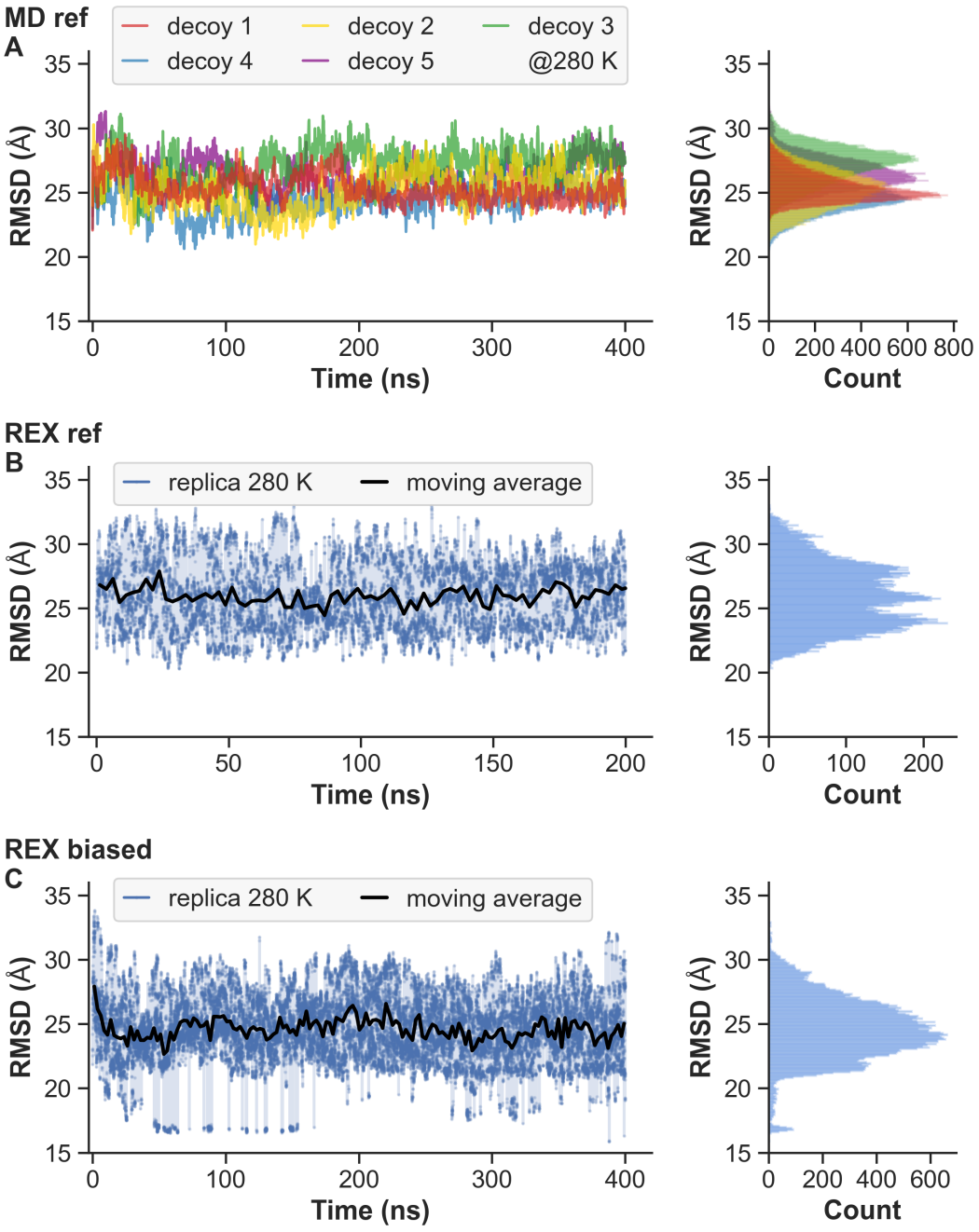

Figure F.3. RMSD comparison of simulations with 3',3'-cGAMP riboswitch (PDB id: 4yaz[243]). Backbone RMSD of MD trajectories at 280 K. MD simulations were performed with five unique starting conformations (*decoys*). **(A)** MD without bias. **(B)** REX MD without bias. **(C)** REX MD with bias.

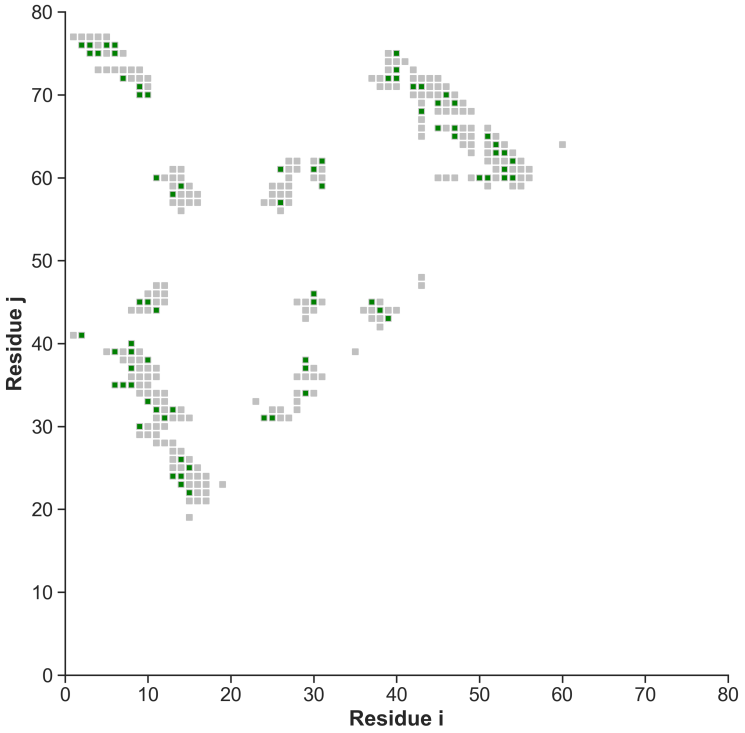

Figure F.4. Contact map of TPP riboswitch (PDB id: 3d2g[241]). Displayed are the native contacts (gray) and 75 randomly selected true-positive bias contacts (green).

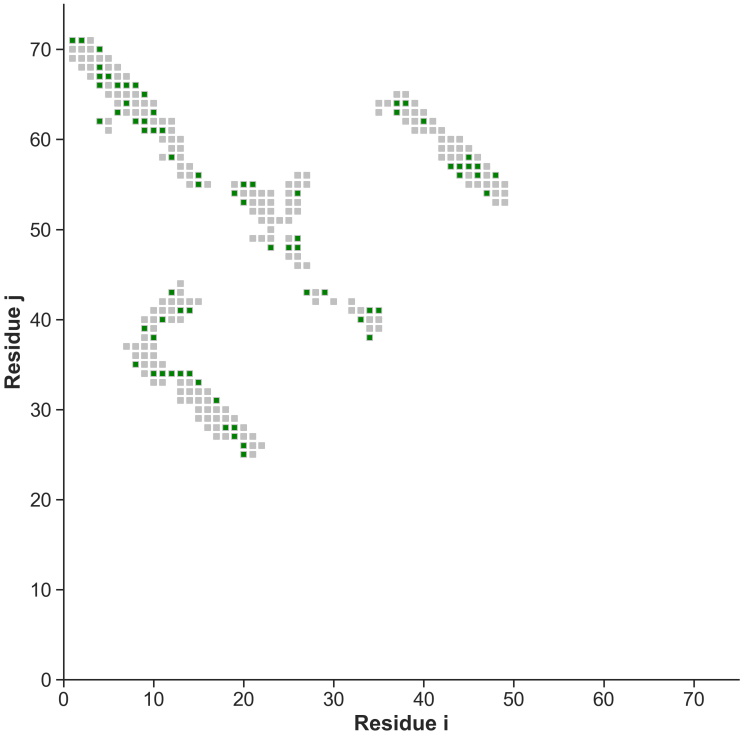

Figure F.5. Contact map of Adenine riboswitch (PDB id: 4tzx[242]). Displayed are the native contacts (gray) and 70 randomly selected true-positive bias contacts (green).

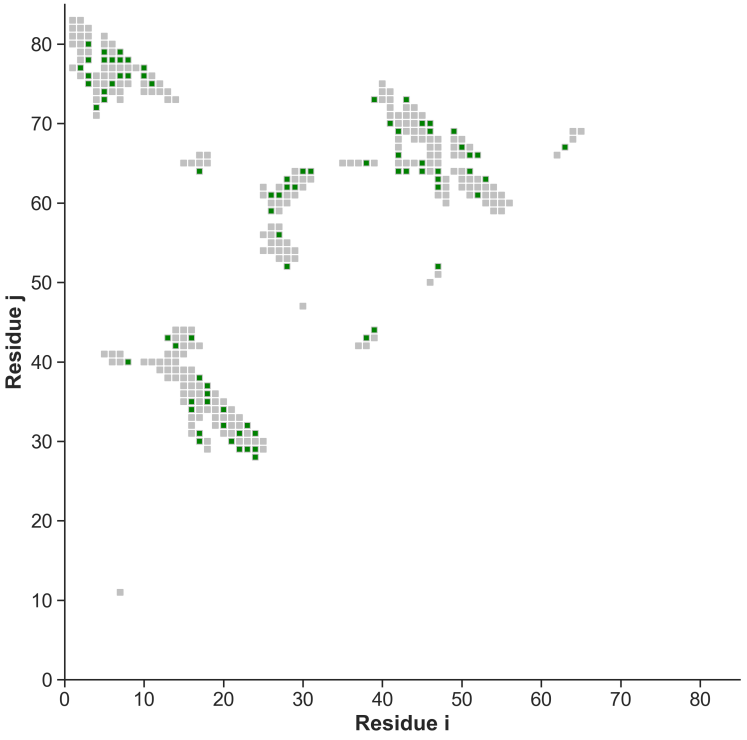

Figure F.6. Contact map of 3',3'-cGAMP riboswitch (PDB id: 4yaz[243]). Displayed are the native contacts (gray) and 80 randomly selected true-positive bias contacts (green).

F.2 Supplementary Tables

Table F.2. Starting decoy accuracy of performed REX MD simulations. Table shows the corresponding replica numbers, PDB ids of RNA targets and backbone root-mean-square-deviation (RMSD) before the simulation started.

replica	Decoy RMSD (Å)			replica	Decoy RMSD (Å)		
number	3d2g	4tzx	4yaz	number	3d2g	4tzx	4yaz
1/21/41	10.96	9.37	21.45	11/31/51	12.64	7.67	23.35
2/22/42	13.55	7.49	23.31	12/32/52	14.13	6.57	22.89
3/23/43	12.41	6.23	23.28	13/33/53	12.60	7.88	22.58
4/24/44	13.95	9.68	21.56	14/34/54	13.20	8.65	21.54
5/25/45	12.45	7.47	22.48	15/35/55	14.32	9.37	23.57
6/26/46	13.03	11.24	24.68	16/36/56	11.15	7.78	22.53
7/27/47	13.08	8.49	20.86	17/37/57	10.97	7.78	24.90
8/28/48	10.94	8.90	24.58	18/38/58	12.58	10.57	24.35
9/29/49	13.21	9.41	22.35	19/39/59	13.58	9.19	21.29
10/30/50	11.85	10.30	21.80	20/40/60	12.79	9.19	21.45

F.3 Used Temperature Distribution

```
REX Temperature Distribution:
T_0 = 280 K ; DELTA = T_0 * (exp(k*i)-exp(k*(i-1)))
T_i = T_(i-1) + a_j * DELTA

Chosen Parameter:
k = 0.0043
a_0 = 1.00 for i = 0..9
a_1 = 1.04 for i = 10..19
a_2 = 1.08 for i = 20..29
a_3 = 1.12 for i = 30..39
a_4 = 1.16 for i = 40..49
a_5 = 1.20 for i = 50..59

Temperatures:
280.00, 281.21, 282.42, 283.64, 284.86, 286.09, 287.32, 288.56, 289.80, 291.05,
292.35, 293.66, 294.98, 296.30, 297.63, 298.96, 300.30, 301.64, 302.99, 304.35,
305.76, 307.18, 308.61, 310.04, 311.48, 312.92, 314.38, 315.83, 317.30, 318.77,
320.30, 321.83, 323.38, 324.93, 326.49, 328.05, 329.62, 331.20, 332.78, 334.37,
336.03, 337.69, 339.36, 341.04, 342.72, 344.41, 346.11, 347.82, 349.53, 351.25,
353.04, 354.83, 356.64, 358.45, 360.27, 362.09, 363.93, 365.77, 367.62, 369.48,
```

Bibliography

[1] A. Voronin, M. Weiel, and A. Schug, "Including residual contact information into replica-exchange MD simulations significantly enriches native-like conformations," *PLOS ONE*, vol. 15, no. 11, pp. 1–24, 2020. doi: 10.1371/journal.pone.0242072.

[2] A. Voronin and A. Schug, "pyrexMD: Workflow-Orientated Python Package for Replica Exchange Molecular Dynamics," *Journal of Open Source Software*, vol. 6, no. 68, p. 3325, 2021. doi: 10.21105/joss.03325.

[3] A. Voronin and A. Schug, "Selection of representative structures from large biomolecular ensembles," *The Journal of Chemical Physics*, vol. 156, no. 14, p. 144102, 2022. doi: 10.1063/5.0082444.

[4] A. Hautke, A. Voronin, F. Idiris, A. Riel, F. Lindner, A. Lelièvre, B. Appel, S. Müller, A. Schug, and S. Ebbinghaus, "Conformation, condensation and mobility of CAG triplet repeat RNA in cells," *submitted*, 2022.

[5] A. V. Guzzo, "The Influence of Amino Acid Sequence on Protein Structure," *Biophysical Journal*, vol. 5, no. 6, pp. 809–822, 1965. doi: 10.1016/S0006-3495(65)86753-4.

[6] H. H. Gan, R. A. Perlow, S. Roy, J. Ko, M. Wu, J. Huang, S. Yan, A. Nicoletta, J. Vafai, D. Sun, *et al.*, "Analysis of Protein Sequence/Structure Similarity Relationships," *Biophysical Journal*, vol. 83, no. 5, pp. 2781–2791, 2002. doi: 10.1016/s0006-3495(02)75287-9.

[7] J. C. Whisstock and A. M. Lesk, "Prediction of protein function from protein sequence and structure," *Quarterly Reviews of Biophysics*, vol. 36, no. 3, pp. 307–340, 2003. doi: 10.1017/S0033583503003901.

[8] M. Varadi and P. Tompa, "The Protein Ensemble Database," *Intrinsically disordered proteins studied by NMR spectroscopy*, pp. 335–349, 2015. doi: 10.1007/978-3-319-20164-1_11.

[9] T. Mittag and J. D. Forman-Kay, "Atomic-level characterization of disordered protein ensembles," *Current Opinion in Structural Biology*, vol. 17, no. 1, pp. 3–14, 2007. doi: 10.1016/j.sbi.2007.01.009.

[10] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. Sonnhammer, *et al.*, "The Pfam protein families database," *Nucleic Acids Research*, vol. 32, no. suppl_1, pp. D138–D141, 2004. doi: 10.1093/nar/gkh121.

[11] R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, *et al.*, "The Pfam protein families database: towards a more sustainable future," *Nucleic Acids Research*, vol. 44, no. D1, pp. D279–D285, 2016. doi: 10.1093/nar/gkv1344.

[12] U. Consortium, "UniProt: a worldwide hub of protein knowledge," *Nucleic Acids Research*, vol. 47, no. D1, pp. D506–D515, 2019. doi: 10.1093/nar/gky1049.

[13] D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, "GenBank," *Nucleic Acids Research*, vol. 41, no. D1, pp. D36–D42, 2012. doi: 10.1093/nar/gks1195.

[14] url: https://www.ncbi.nlm.nih.gov/genbank/statistics/, Accessed: 2022-03-01.

[15] L. Gremer, D. Schölzel, C. Schenk, E. Reinartz, J. Labahn, R. B. Ravelli, M. Tusche, C. Lopez-Iglesias, W. Hoyer, H. Heise, *et al.*, "Fibril structure of amyloid-β (1–42) by cryo–electron microscopy," *Science*, vol. 358, no. 6359, pp. 116–119, 2017. doi: 10.1126/science.aao2825.

[16] J. A. Geraets, K. R. Pothula, and G. F. Schröder, "Integrating cryo-EM and NMR data," *Current Opinion in Structural Biology*, vol. 61, pp. 173–181, 2020. doi: 10.1016/j.sbi.2020.01.008.

[17] H. P. Erickson, "Size and Shape of Protein Molecules at the Nanometer Level Determined by Sedimentation, Gel Filtration, and Electron Microscopy," *Biological Procedures Online*, vol. 11, no. 1, pp. 32–51, 2009. doi: 10.1007/s12575-009-9008-x.

[18] url: https://www.dynamic-biosensors.com/project/list-of-protein-hydrodynamic-diameters/, Accessed: 2022-03-01.

[19] I. Reinartz, C. Sinner, D. Nettels, B. Stucki-Buchli, F. Stockmar, P. T. Panek, C. R. Jacob, G. U. Nienhaus, B. Schuler, and A. Schug, "Simulation of FRET dyes allows quantitative comparison against experimental data," *The Journal of Chemical Physics*, vol. 148, no. 12, p. 123321, 2018. doi: 10.1063/1.5010434.

[20] M. Weiel, I. Reinartz, and A. Schug, "Rapid interpretation of small-angle X-ray scattering data," *PLoS Computational Biology*, vol. 15, no. 3, pp. 1–27, 2019. doi: 10.1371/journal.pcbi.1006900.

[21] T. Nigam, K.-Y. Yiang, and A. Marathe, "Moore's Law: Technology Scaling and Reliability Challenges," *Microelectronics to Nanoelectronics: Materials, Devices & Manufacturability*, p. 1, 2012.

[22] M. M. Waldrop, "The chips are down for Moore's law," *Nature News*, vol. 530, no. 7589, p. 144, 2016. doi: 10.1038/530144a.

[23] M. A. Martí-Renom, A. C. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Šali, "Comparative Protein Structure Modeling of Genes and Genomes," *Annual Review of Biophysics and Biomolecular Structure*, vol. 29, no. 1, pp. 291–325, 2000. doi: 10.1146/annurev.biophys.29.1.291.

[24] C. N. Cavasotto and S. S. Phatak, "Homology modeling in drug discovery: current trends and applications," *Drug Discovery Today*, vol. 14, no. 13-14, pp. 676–683, 2009. doi: 10.1016/j.drudis.2009.04.006.

[25] D. E. Kim, D. Chivian, and D. Baker, "Protein structure prediction and analysis using the Robetta server," *Nucleic Acids Research*, vol. 32, no. suppl_2, pp. W526–W531, 2004. doi: 10.1093/nar/gkh468.

[26] F. Zhao, J. Peng, J. Debartolo, K. F. Freed, T. R. Sosnick, and J. Xu, "A Probabilistic and Continuous Model of Protein Conformational Space for Template-Free Modeling," *Journal of Computational Biology*, vol. 17, no. 6, pp. 783–798, 2010. doi: 10.1089/cmb.2009.0235.

[27] Z. Wang, F. Zhao, J. Peng, and J. Xu, "Protein 8-class Secondary Structure Prediction Using Conditional Neural Fields," in *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 109–114, IEEE, 2010. doi: 10.1109/BIBM.2010.5706547.

[28] J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano, "Critical assessment of methods of protein structure prediction (CASP)—Round XII," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, pp. 7–15, 2018. doi: 10.1002/prot.25415.

[29] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moult, "Critical assessment of methods of protein structure prediction (CASP)—round XIV," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 12, pp. 1607–1617, 2021. doi: 10.1002/prot.26237.

[30] M. AlQuraishi, "End-to-End Differentiable Learning of Protein Structure," *Cell Systems*, vol. 8, no. 4, pp. 292–301, 2019. doi: 10.1016/j.cels.2019.03.006.

[31] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. Nelson, A. Bridgland, *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, 2020. doi: 10.1038/s41586-019-1923-7.

[32] M. Karplus and J. Kuriyan, "Molecular dynamics and protein function," *Proceedings of the National Academy of Sciences*, vol. 102, no. 19, pp. 6679–6685, 2005. doi: 10.1073/pnas.0408930102.

[33] D. Paschek, S. Hempel, and A. E. García, "Computing the stability diagram of the Trp-cage miniprotein," *Proceedings of the National Academy of Sciences*, vol. 105, no. 46, pp. 17754–17759, 2008. doi: 10.1073/pnas.0804775105.

[34] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, "How Fast-Folding Proteins Fold," *Science*, vol. 334, no. 6055, pp. 517–520, 2011. doi: 10.1126/science.1208351.

[35] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao, *et al.*, "Anton, a Special-Purpose Machine for Molecular Dynamics Simulation," *Communications of the ACM*, vol. 51, no. 7, pp. 91–97, 2008. doi: 10.1145/1364782.1364802.

[36] D. E. Shaw, R. O. Dror, J. K. Salmon, J. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff, B. Batson, K. J. Bowers, *et al.*, "Millisecond-Scale Molecular Dynamics Simulations on Anton," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pp. 1–11, 2009. doi: 10.1145/1654059.1654126.

[37] W. Boomsma, J. Ferkinghoff-Borg, and K. Lindorff-Larsen, "Combining Experiments and Simulations Using the Maximum Entropy Principle," *PLoS Computational Biology*, vol. 10, no. 2, p. e1003406, 2014. doi: 10.1371/journal.pcbi.1003406.

[38] A. Björling, S. Niebling, M. Marcellini, D. van der Spoel, and S. Westenhoff, "Deciphering Solution Scattering Data with Experimentally Guided Molecular Dynamics Simulations," *Journal of Chemical Theory and Computation*, vol. 11, no. 2, pp. 780–787, 2015. doi: 10.1021/ct5009735.

[39] P.-c. Chen and J. S. Hub, "Interpretation of Solution X-Ray Scattering by Explicit-Solvent Molecular Dynamics," *Biophysical Journal*, vol. 108, no. 10, pp. 2573–2584, 2015. doi: 10.1016/j.bpj.2015.03.062.

[40] R. Shevchuk and J. S. Hub, "Bayesian refinement of protein structures and ensembles against SAXS data using molecular dynamics," *PLoS Computational Biology*, vol. 13, no. 10, p. e1005800, 2017. doi: 10.1371/journal.pcbi.1005800.

[41] O. F. Lange, N.-A. Lakomek, C. Farès, G. F. Schroder, K. F. Walter, S. Becker, J. Meiler, H. Grubmuller, C. Griesinger, and B. L. De Groot, "Recognition Dynamics Up to Microseconds Revealed from an RDC-Derived Ubiquitin Ensemble in Solution," *Science*, vol. 320, no. 5882, pp. 1471–1475, 2008. doi: 10.1126/science.1157092.

[42] J. D. Bryngelson, J. N. Onuchic, N. D. Socci, and P. G. Wolynes, "Funnels, Pathways, and the Energy Landscape of Protein Folding: A Synthesis," *Proteins: Structure, Function, and Bioinformatics*, vol. 21, no. 3, pp. 167–195, 1995. doi: 10.1002/prot.340210302.

[43] A. Schug and J. N. Onuchic, "From protein folding to protein function and biomolecular binding by energy landscape theory," *Current Opinion in Pharmacology*, vol. 10, no. 6, pp. 709–714, 2010. doi: 10.1016/j.coph.2010.09.012.

[44] A. Raval, S. Piana, M. P. Eastwood, and D. E. Shaw, "Assessment of the utility of contact-based restraints in accelerating the prediction of protein structure using molecular dynamics simulations," *Protein Science*, vol. 25, no. 1, pp. 19–29, 2016. doi: 10.1002/pro.2770.

[45] A. Schug, T. Herges, and W. Wenzel, "Reproducible Protein Folding with the Stochastic Tunneling Method," *Physical Review Letters*, vol. 91, no. 15, p. 158102, 2003. doi: 10.1103/PhysRevLett.91.158102.

[46] R. C. Bernardi, M. C. Melo, and K. Schulten, "Enhanced sampling techniques in molecular dynamics simulations of biological systems," *Biochimica et Biophysica Acta (BBA) - General Subjects*, vol. 1850, no. 5, pp. 872–877, 2015. doi: 10.1016/j.bbagen.2014.10.019.

[47] E. K. Peter, D. J. Manstein, J.-E. Shea, and A. Schug, "Core-md ii: A fast, adaptive, and accurate enhanced sampling method," *The Journal of Chemical Physics*, vol. 155, no. 10, p. 104114, 2021. doi: 10.1063/5.0063664.

[48] M. M. Dedmon, K. Lindorff-Larsen, J. Christodoulou, M. Vendruscolo, and C. M. Dobson, "Mapping Long-Range Interactions in α-Synuclein using Spin-Label NMR and Ensemble Molecular Dynamics Simulations," *Journal of the American Chemical Society*, vol. 127, no. 2, pp. 476–477, 2005. doi: 10.1021/ja044834j.

[49] M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, and T. Hwa, "Identification of direct residue contacts in protein–protein interaction by message passing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 1, pp. 67–72, 2009. doi: 10.1073/pnas.0805923106.

[50] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, "Direct-coupling analysis of residue coevolution captures native contacts across many protein families," *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. E1293–E1301, 2011. doi: 10.1073/pnas.1111471108.

[51] A. Schug, M. Weigt, J. N. Onuchic, T. Hwa, and H. Szurmant, "High-resolution protein complexes from integrating genomic information with molecular simulation," *Proceedings of the National Academy of Sciences*, vol. 106, no. 52, pp. 22124–22129, 2009. doi: 10.1073/pnas.0912100106.

[52] G. Uguzzoni, S. J. Lovis, F. Oteri, A. Schug, H. Szurmant, and M. Weigt, "Large-scale identification of coevolution signals across homo-oligomeric protein interfaces by direct coupling analysis," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. E2662–E2671, 2017. doi: 10.1073/pnas.1615068114.

[53] E. De Leonardis, B. Lutz, S. Ratz, S. Cocco, R. Monasson, A. Schug, and M. Weigt, "Direct-Coupling Analysis of nucleotide coevolution facilitates RNA secondary and tertiary structure prediction," *Nucleic Acids Research*, vol. 43, no. 21, pp. 10444–10455, 2015. doi: 10.1093/nar/gkv932.

[54] F. Morcos, B. Jana, T. Hwa, and J. N. Onuchic, "Coevolutionary signals across protein lineages help capture multiple protein conformations," *Proceedings of the National Academy of Sciences*, vol. 110, no. 51, pp. 20533–20538, 2013. doi: 10.1073/pnas.1315625110.

[55] R. H. Swendsen and J.-S. Wang, "Replica Monte Carlo Simulation of Spin-Glasses," *Physical Review Letters*, vol. 57, no. 21, p. 2607, 1986. doi: 10.1103/PhysRevLett.57.2607.

[56] Y. Sugita and Y. Okamoto, "Replica-exchange molecular dynamics method for protein folding," *Chemical Physics Letters*, vol. 314, no. 1-2, pp. 141–151, 1999. doi: 10.1016/S0009-2614(99)01123-9.

[57] A. Schug, T. Herges, and W. Wenzel, "All-Atom Folding of the Three-Helix HIV Accessory ProteinWith an Adaptive Parallel Tempering Method," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 4, pp. 792–798, 2004. doi: 10.1002/prot.20290.

[58] K. Y. Sanbonmatsu and A. E. García, "Structure of Met-Enkephalin in Explicit Aqueous Solution Using Replica Exchange Molecular Dynamics," *Proteins: Structure, Function, and Bioinformatics*, vol. 46, no. 2, pp. 225–234, 2002. doi: 10.1002/prot.1167.

[59] U. H. Hansmann, "Parallel tempering algorithm for conformational studies of biological molecules," *Chemical Physics Letters*, vol. 281, no. 1-3, pp. 140–150, 1997. doi: 10.1016/S0009-2614(97)01198-6.

[60] A. Ambrogelly, S. Palioura, and D. Söll, "Natural expansion of the genetic code," *Nature Chemical Biology*, vol. 3, no. 1, pp. 29–35, 2007. doi: 10.1038/nchembio847.

[61] D. S. Latchman, "Transcription Factors: An Overview," *The International Journal of Biochemistry & Cell Biology*, vol. 29, no. 12, pp. 1305–1312, 1997. doi: 10.1016/S1357-2725(97)00085-X.

[62] G. E. Mortimore, A. R. Pösö, and B. R. Lardeux, "Mechanism and Regulation of Protein Degradation in Liver," *Diabetes/metabolism Reviews*, vol. 5, no. 1, pp. 49–70, 1989. doi: 10.1002/dmr.5610050105.

[63] A. D. Theocharis, S. S. Skandalis, C. Gialeli, and N. K. Karamanos, "Extracellular matrix structure," *Advanced Drug Delivery Reviews*, vol. 97, pp. 4–27, 2016. doi: 10.1016/j.addr.2015.11.001.

[64] V. Hatzimanikatis, C. Li, J. A. Ionita, and L. J. Broadbelt, "Metabolic networks: enzyme function and metabolite structure," *Current Opinion in Structural Biology*, vol. 14, no. 3, pp. 300–306, 2004. doi: 10.1016/j.sbi.2004.04.004.

[65] G. M. Cooper, R. E. Hausman, and R. E. Hausman, *The Cell: A Molecular Approach*, vol. 4. ASM press Washington, DC, USA, 2007.

[66] J. Hankins, "The role of albumin in fluid and electrolyte balance," *Journal of Infusion Nursing*, vol. 29, no. 5, pp. 260–265, 2006. doi: 10.1097/00129804-200609000-00004.

[67] C. B. Anfinsen, "Principles that Govern the Folding of Protein Chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973. doi: 10.1126/science.181.4096.223.

[68] A. K. Dunker, J. D. Lawson, C. J. Brown, R. M. Williams, P. Romero, J. S. Oh, C. J. Oldfield, A. M. Campen, C. M. Ratliff, K. W. Hipps, *et al.*, "Intrinsically disordered protein," *Journal of Molecular Graphics and Modelling*, vol. 19, no. 1, pp. 26–59, 2001. doi: 10.1016/S1093-3263(00)00138-8.

[69] H. J. Dyson and P. E. Wright, "Intrinsically unstructured proteins and their functions," *Nature Reviews Molecular Cell Biology*, vol. 6, no. 3, pp. 197–208, 2005. doi: 10.1038/nrm1589.

[70] L. Brocchieri and S. Karlin, "Protein length in eukaryotic and prokaryotic proteomes," *Nucleic Acids Research*, vol. 33, no. 10, pp. 3390–3400, 2005. doi: 10.1093/nar/gki615.

[71] C. Levinthal, "How to fold graciously," *Mossbauer Spectroscopy in Biological Systems*, vol. 67, pp. 22–24, 1969.

[72] J. Kubelka, J. Hofrichter, and W. A. Eaton, "The protein folding 'speed limit'," *Current Opinion in Structural Biology*, vol. 14, no. 1, pp. 76–88, 2004. doi: 10.1016/j.sbi.2004.01.013.

[73] T. Veitshans, D. Klimov, and D. Thirumalai, "Protein folding kinetics: timescales, pathways and energy landscapes in terms of sequence-dependent properties," *Folding and Design*, vol. 2, no. 1, pp. 1–22, 1997. doi: 10.1016/S1359-0278(97)00002-3.

[74] B. Schuler and H. Hofmann, "Single-molecule spectroscopy of protein folding dynamics—expanding scope and timescales," *Current Opinion in Structural Biology*, vol. 23, no. 1, pp. 36–47, 2013. doi: 10.1016/j.sbi.2012.10.008.

[75] R. Zwanzig, A. Szabo, and B. Bagchi, "Levinthal's paradox," *Proceedings of the National Academy of Sciences*, vol. 89, no. 1, pp. 20–22, 1992. doi: 10.1073/pnas.89.1.20.

[76] K. A. Dill and H. S. Chan, "From Levinthal to pathways to funnels," *Nature Structural Biology*, vol. 4, no. 1, pp. 10–19, 1997. doi: 10.1038/nsb0197-10.

[77] J. N. Onuchic, Z. Luthey-Schulten, and P. G. Wolynes, "THEORY OF PROTEIN FOLDING: The Energy Landscape Perspective," *Annual Review of Physical Chemistry*, vol. 48, no. 1, pp. 545–600, 1997. doi: 10.1146/annurev.physchem.48.1.545.

[78] N. D. Socci, J. N. Onuchic, and P. G. Wolynes, "Protein Folding Mechanismsand the Multidimensional Folding Funnel," *Proteins: Structure, Function, and Bioinformatics*, vol. 32, no. 2, pp. 136–158, 1998. doi: 10.1002/(SICI)1097-0134(19980801)32:2<136::AID-PROT2>3.0.CO;2-J.

[79] R. A. Goldbeck, Y. G. Thomas, E. Chen, R. M. Esquerra, and D. S. Kliger, "Multiple pathways on a protein-folding energy landscape: Kinetic evidence," *Proceedings of the National Academy of Sciences*, vol. 96, no. 6, pp. 2782–2787, 1999. doi: 10.1073/pnas.96.6.2782.

[80] A. Barducci, M. Bonomi, and M. Parrinello, "Metadynamics," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 1, no. 5, pp. 826–843, 2011. doi: 10.1002/wcms.31.

[81] L. C. Pierce, R. Salomon-Ferrer, C. Augusto F. de Oliveira, J. A. McCammon, and R. C. Walker, "Routine Access to Millisecond Time Scale Events with Accelerated Molecular Dynamics," *Journal of Chemical Theory and Computation*, vol. 8, no. 9, pp. 2997–3002, 2012. doi: 10.1021/ct300284c.

[82] F. Pietrucci, "Strategies for the exploration of free energy landscapes: Unity in diversity and challenges ahead," *Reviews in Physics*, vol. 2, pp. 32–45, 2017. doi: 10.1016/j.revip.2017.05.001.

[83] M. A. Fares and S. A. Travers, "A Novel Method for Detecting Intramolecular Coevolution: Adding a Further Dimension to Selective Constraints Analyses," *Genetics*, vol. 173, no. 1, pp. 9–23, 2006. doi: 10.1534/genetics.105.053249.

[84] D. De Juan, F. Pazos, and A. Valencia, "Emerging methods in protein co-evolution," *Nature Reviews Genetics*, vol. 14, no. 4, pp. 249–261, 2013. doi: 10.1038/nrg3414.

[85] C.-H. Yeang and D. Haussler, "Detecting Coevolution in and among Protein Domains," *PLoS Computational Biology*, vol. 3, no. 11, p. e211, 2007. doi: 10.1371/journal.pcbi.0030211.

[86] P. A. Nuin, Z. Wang, and E. R. Tillier, "The accuracy of several multiple sequence alignment programs for proteins," *BMC Bioinformatics*, vol. 7, no. 1, pp. 1–18, 2006. doi: 10.1186/1471-2105-7-471.

[87] C. Notredame, "Recent Evolutions of Multiple Sequence Alignment Algorithms," *PLoS Computational Biology*, vol. 3, no. 8, p. e123, 2007. doi: 10.1371/journal.pcbi.0030123.

[88] J. Daugelaite, A. O'Driscoll, and R. D. Sleator, "An Overview of Multiple Sequence Alignments and Cloud Computing in Bioinformatics," *International Scholarly Research Notices*, vol. 2013, 2013. doi: 10.1155/2013/615630.

[89] F. A. Bovey, P. A. Mirau, and H. Gutowsky, *Nuclear Magnetic Resonance Spectroscopy*. Elsevier, 1988.

[90] "NMR - Interpretation," Aug 15 2020. url: https://chem.libretexts.org/@go/page/1812, Accessed: 2022-03-11.

[91] R. K. Harris, E. D. Becker, S. M. C. De Menezes, R. Goodfellow, and P. Granger, "NMR nomenclature. Nuclear spin properties and conventions for chemical shifts (IUPAC Recommendations 2001)," *Pure and Applied Chemistry*, vol. 73, no. 11, pp. 1795–1818, 2001. doi: 10.1351/pac200173111795.

[92] "Reference Material and Data." url: https://www.validnmr.com/w/index.php?title=Reference_Material_and_Data, Accessed: 2022-03-11.

[93] E. D. Becker, "Chapter 4 - Chemical Shifts," in *High Resolution NMR (Third Edition)*, pp. 83–117, San Diego: Academic Press, 2000. doi: 10.1016/B978-012084662-7/50048-X.

[94] L. Pohl and M. Eckle, "Sodium 3-Trimethylsilyltetradeuteriopropionate, a New Water-Soluble Standard for ^{1}H-NMR," *Angewandte Chemie International Edition*, vol. 8, no. 5, pp. 381–381, 1969. doi: 10.1002/anie.196903811.

[95] url: https://www.chemistrysteps.com/category/nuclear-magnetic-resonance-nmr-spectroscopy, Accessed: 2022-03-11.

[96] E. E. Abola, F. C. Bernstein, and T. F. Koetzle, "The Protein Data Bank," in *Neutrons in Biology*, pp. 441–441, Springer, 1984. doi: 10.1007/978-1-4899-0375-4_26.

[97] Y. Li, C. Zhang, E. W. Bell, D.-J. Yu, and Y. Zhang, "Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1082–1091, 2019. doi: 10.1002/prot.25798.

[98] Y. Li, J. Hu, C. Zhang, D.-J. Yu, and Y. Zhang, "ResPRE: high-accuracy protein contact prediction by coupling precision matrix with deep residual neural networks," *Bioinformatics*, vol. 35, no. 22, pp. 4647–4655, 2019. doi: 10.1093/bioinformatics/btz291.

[99] M. AlQuraishi, "AlphaFold at CASP13," *Bioinformatics*, vol. 35, no. 22, pp. 4862–4865, 2019. doi: 10.1093/bioinformatics/btz422.

[100] E. Callaway, "'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures," *Nature*, vol. 588, no. 7837, pp. 203–205, 2020. doi: 10.1038/d41586-020-03348-4.

[101] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021. doi: 10.1038/s41586-021-03819-2.

[102] M. Karplus and G. A. Petsko, "Molecular dynamics simulations in biology," *Nature*, vol. 347, no. 6294, pp. 631–639, 1990. doi: 10.1038/347631a0.

[103] G. d. M. Seabra, R. C. Walker, M. Elstner, D. A. Case, and A. E. Roitberg, "Implementation of the SCC-DFTB Method for Hybrid QM/MM Simulations within the Amber Molecular Dynamics Package," *The Journal of Physical Chemistry A*, vol. 111, no. 26, pp. 5655–5664, 2007. doi: 0.1021/jp070071l.

[104] G. Groenhof, "Introduction to QM/MM Simulations," *Biomolecular Simulations*, pp. 43–66, 2013. doi: 10.1007/978-1-62703-017-5_3.

[105] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case, "Development and Testing of a General Amber Force Field," *Journal of Computational Chemistry*, vol. 25, no. 9, pp. 1157–1174, 2004. doi: 10.1002/jcc.20035.

[106] P. Bjelkmar, P. Larsson, M. A. Cuendet, B. Hess, and E. Lindahl, "Implementation of the CHARMM Force Field in GROMACS: Analysis of Protein Stability Effects from Correction Maps, Virtual Interaction Sites, and Water Models," *Journal of Chemical Theory and Computation*, vol. 6, no. 2, pp. 459–466, 2010. doi: 10.1021/ct900549r.

[107] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. Berendsen, "GROMACS: Fast, Flexible, and Free," *Journal of Computational Chemistry*, vol. 26, no. 16, pp. 1701–1718, 2005. doi: 10.1002/jcc.20291.

[108] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, "GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers," *SoftwareX*, vol. 1, pp. 19–25, 2015. doi: 10.1016/j.softx.2015.06.001.

[109] Lindahl, Abraham, Hess, and van der Spoel, "GROMACS 2020 Manual," Jan. 2020. doi: 10.5281/zenodo.3562512.

[110] K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw, "Improved side-chain torsion potentials for the Amber ff99SB protein force field," *Proteins: Structure, Function, and Bioinformatics*, vol. 78, no. 8, pp. 1950–1958, 2010. doi: 10.1002/prot.22711.

[111] J. E. Lennard-Jones, "Cohesion," *Proceedings of the Physical Society (1926-1948)*, vol. 43, no. 5, p. 461, 1931. doi: 10.1088/0959-5309/43/5/301.

[112] G. Bussi, D. Donadio, and M. Parrinello, "Canonical sampling through velocity rescaling," *The Journal of Chemical Physics*, vol. 126, no. 1, p. 014101, 2007. doi: 10.1063/1.2408420.

[113] P. H. Hünenberger, "Thermostat Algorithms for Molecular Dynamics Simulations," in *Advanced Computer Simulation*, pp. 105–149, Springer, 2005. doi: 10.1007/b99427.

[114] H. J. Berendsen, J. v. Postma, W. F. Van Gunsteren, A. DiNola, and J. R. Haak, "Molecular dynamics with coupling to an external bath," *The Journal of Chemical Physics*, vol. 81, no. 8, pp. 3684–3690, 1984. doi: 10.1063/1.448118.

[115] M. Parrinello and A. Rahman, "Polymorphic transitions in single crystals: A new molecular dynamics method," *Journal of Applied Physics*, vol. 52, no. 12, pp. 7182–7190, 1981. doi: 10.1063/1.328693.

[116] S. Nosé and M. Klein, "Constant pressure molecular dynamics for molecular systems," *Molecular Physics*, vol. 50, no. 5, pp. 1055–1076, 1983. doi: 10.1080/00268978300102851.

[117] K. Luby-Phelps, "Cytoarchitecture and Physical Properties of Cytoplasm: Volume, Viscosity, Diffusion, Intracellular Surface Area," *International Review of Cytology*, vol. 192, pp. 189–221, 1999. doi: 10.1016/S0074-7696(08)60527-6.

[118] J. S. D'Arrigo, "Screening of membrane surface charges by divalent cations: an atomic representation," *American Journal of Physiology-Cell Physiology*, vol. 235, no. 3, pp. C109–C117, 1978. doi: 10.1152/ajpcell.1978.235.3.C109.

[119] W. L. Jorgensen, "Transferable Intermolecular Potential Functions for Water, Alcohols, and Ethers. Application to Liquid Water," *Journal of the American Chemical Society*, vol. 103, no. 2, pp. 335–340, 1981. doi: 10.1021/ja00392a016.

[120] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, "Comparison of simple potential functions for simulating liquid water," *The Journal of Chemical Physics*, vol. 79, no. 2, pp. 926–935, 1983. doi: 10.1063/1.445869.

[121] J. L. Abascal, R. G. Fernández, C. Vega, and M. A. Carignano, "The melting temperature of the six site potential model of water," *The Journal of Chemical Physics*, vol. 125, no. 16, p. 166101, 2006. doi: 10.1063/1.2360276.

[122] P. Florová, P. Sklenovsky, P. Banáš, and M. Otyepka, "Explicit Water Models Affect the Specific Solvation and Dynamics of Unfolded Peptides While the Conformational Behavior and Flexibility of Folded Peptides Remain Intact," *Journal of Chemical Theory and Computation*, vol. 6, no. 11, pp. 3569–3579, 2010. doi: 10.1021/ct1003687.

[123] H. J. Berendsen, J. P. Postma, W. F. van Gunsteren, and J. Hermans, "Interaction Models for Water in Relation to Protein Hydration," in *Intermolecular Forces*, pp. 331–342, Springer, 1981. doi: 10.1007/978-94-015-7658-1_21.

[124] H. Berendsen, J. Grigera, and T. Straatsma, "The Missing Term in Effective Pair Potentials," *Journal of Physical Chemistry*, vol. 91, no. 24, pp. 6269–6271, 1987. doi: 10.1021/j100308a038.

[125] D. Sindhikara, Y. Meng, and A. E. Roitberg, "Exchange frequency in replica exchange molecular dynamics," *The Journal of Chemical Physics*, vol. 128, no. 2, p. 01B609, 2008. doi: 10.1063/1.2816560.

[126] D. J. Sindhikara, D. J. Emerson, and A. E. Roitberg, "Exchange Often and Properly in Replica Exchange Molecular Dynamics," *Journal of Chemical Theory and Computation*, vol. 6, no. 9, pp. 2804–2808, 2010. doi: 10.1021/ct100281c.

[127] J. R. Perilla, G. Zhao, M. Lu, J. Ning, G. Hou, I.-J. L. Byeon, A. M. Gronenborn, T. Polenova, and P. Zhang, "CryoEM Structure Refinement by Integrating NMR Chemical Shifts with Molecular Dynamics Simulations," *The Journal of Physical Chemistry B*, vol. 121, no. 15, pp. 3853–3863, 2017. doi: 10.1021/acs.jpcb.6b13105.

[128] I. Kufareva and R. Abagyan, "Methods of Protein Structure Comparison," in *Homology Modeling*, pp. 231–257, Springer, 2011. doi: 10.1007/978-1-61779-588-6_10.

[129] A. Zemla, "LGA: a method for finding 3D similarities in protein structures," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3370–3374, 2003.

[130] V. Modi and R. L. Dunbrack Jr, "Assessment of refinement of template-based models in CASP11," *Proteins: Structure, Function, and Bioinformatics*, vol. 84, pp. 260–281, 2016.

[131] "CASP 14 results (refinement targets)," 2020. url: https://predictioncenter.org/casp14/results.cgi?view=tables&target=R1029, Accessed: 2021-12-14.

[132] R. Wüst, *Mathematik für Physiker und Mathematiker: Band 1: Reelle Analysis und Lineare Algebra*. John Wiley & Sons, 2008.

[133] L. Van Der Maaten, E. Postma, J. Van den Herik, *et al.*, "Dimensionality Reduction: A Comparative Review," *Journal of Machine Learning Research*, vol. 10, no. 66-71, p. 13, 2009.

[134] M. Song, H. Yang, S. H. Siadat, and M. Pechenizkiy, "A comparative study of dimensionality reduction techniques to enhance trace clustering performances," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3722–3737, 2013. doi: 10.3923/jai.2010.119.134.

[135] P. Pudil and J. Novovičová, "Novel Methods for Feature Subset Selection with Respect to Problem Knowledge," in *Feature Extraction, Construction and Selection*, pp. 101–116, Springer, 1998. doi: 10.1007/978-1-4615-5725-8_7.

[136] M. L. Braun, J. Buhmann, and K.-R. Müller, "Denoising and Dimension Reduction in Feature Space," *Advances in Neural Information Processing Systems*, vol. 19, p. 185–192, 2006.

[137] L. Van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[138] L. Van Der Maaten, "Accelerating t-SNE using Tree-Based Algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[139] G. E. Hinton and S. Roweis, "Stochastic Neighbor Embedding," *Advances in Neural Information Processing Systems*, vol. 15, 2002. url: https://www.cs.toronto.edu/~hinton/absps/sne.pdf.

[140] L. Bottou, "Stochastic Gradient Descent Tricks," in *Neural Networks: Tricks of the Trade*, pp. 421–436, Springer, 2012. doi: 10.1007/978-3-642-35289-8_25.

[141] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. doi: 10.1214/aoms/1177729694.

[142] J. Shlens, "Notes on Kullback-Leibler Divergence and Likelihood Theory," *arXiv preprint*, 2014. arXiv: 1404.2000.

[143] I. Wallach and R. Lilien, "The protein–small-molecule database, a non-redundant structural resource for the analysis of protein-ligand binding," *Bioinformatics*, vol. 25, no. 5, pp. 615–620, 2009. doi: 10.1093/bioinformatics/btp035.

[144] A. R. Jamieson, M. L. Giger, K. Drukker, H. Li, Y. Yuan, and N. Bhooshan, "Exploring nonlinear feature space dimension reduction and data representation in breast CADx with Laplacian eigenmaps and t-SNE," *Medical Physics*, vol. 37, no. 1, pp. 339–351, 2010. doi: 10.1118/1.3267037.

[145] A. Miao, J. Zhuang, Y. Tang, Y. He, X. Chu, and S. Luo, "Hyperspectral Image-Based Variety Classification of Waxy Maize Seeds by the t-SNE Model and Procrustes Analysis," *Sensors*, vol. 18, no. 12, p. 4391, 2018. doi: 10.3390/s18124391.

[146] V. Pratap, A. Sriram, P. Tomasello, A. Hannun, V. Liptchinsky, G. Synnaeve, and R. Collobert, "Massively Multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters," *arXiv preprint*, 2020. doi: 10.48550/arXiv.2007.03001.

[147] J. Birjandtalab, M. B. Pouyan, and M. Nourani, "Nonlinear Dimension Reduction for EEG-Based Epileptic Seizure Detection," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 595–598, IEEE, 2016. doi: 10.1109/BHI.2016.7455968.

[148] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks.," in *ISMIR*, vol. 10, pp. 339–344, Citeseer, 2010.

[149] I. Borg and P. J. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media, 2005. doi: 10.1007/0-387-28981-X.

[150] C. S. Ding, *Fundamentals of Applied Multidimensional Scaling for Educational and Psychological Research*. Springer, 2018. doi: 10.1007/978-3-319-78172-3.

[151] J. B. Kruskal, "Multidimensional scaling in archaeology: Time is not the only dimension," *Mathematics in the archaeological and historical sciences*, vol. 119, p. 32, 1971.

[152] M. S. Venkatarajan and W. Braun, "New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical–chemical properties," *Molecular Modeling Annual*, vol. 7, no. 12, pp. 445–453, 2001. doi: 10.1007/s00894-001-0058-5.

[153] J. Cameron, "Nuclear Physics: Pattern Recognition in Nuclear Gamma—ray Spectra," *Multidimensional Scaling: History, Theory, and Applications*, p. 163, 1987.

[154] L. G. Cooper, "A Review of Multidimensional Scaling in Marketing Research ," *Applied Psychological Measurement*, vol. 7, no. 4, pp. 427–450, 1983. doi: 10.1177/014662168300700404.

[155] B. E. Lauderdale and T. S. Clark, "Scaling Politically Meaningful Dimensions Using Texts and Votes," *American Journal of Political Science*, vol. 58, no. 3, pp. 754–771, 2014. doi: 10.1111/ajps.12085.

[156] N. Jaworska and A. Chupetlovska-Anastasova, "A Review of Multidimensional Scaling (MDS) and its Utility in Various Psychological Domains," *Tutorials in Quantitative Methods for Psychology*, vol. 5, no. 1, pp. 1–10, 2009. doi: 10.20982/tqmp.05.1.p001.

[157] C. Williams, "On a Connection between Kernel PCA and Metric Multidimensional Scaling," *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[158] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964. doi: 10.1007/BF02289565.

[159] J. W. Sammon, "A Nonlinear Mapping for Data Structure Analysis," *IEEE Transactions on Computers*, vol. 100, no. 5, pp. 401–409, 1969. doi: 10.1109/T-C.1969.222678.

[160] L. Duan, L. Xu, Y. Liu, and J. Lee, "Cluster-based outlier detection," *Annals of Operations Research*, vol. 168, no. 1, pp. 151–168, 2009. doi: 10.1007/s10479-008-0371-9.

[161] T. S. Madhulatha, "An Overview on Clustering Methods," *arXiv preprint*, 2012. arXiv: 1205.1117.

[162] J. Swarndeep Saket and S. Pandya, "An Overview of Partitioning Algorithms in Clustering Techniques," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 5, no. 6, pp. 1943–1946, 2016.

[163] D. Steinley, "K-means clustering: A half-century synthesis," *British Journal of Mathematical and Statistical Psychology*, vol. 59, no. 1, pp. 1–34, 2006. doi: 10.1348/000711005X48266.

[164] A. Vouros, S. Langdell, M. Croucher, and E. Vasilaki, "An empirical comparison between stochastic and deterministic centroid initialisation for K-means variations," *Machine Learning*, vol. 110, no. 8, pp. 1975–2003, 2021. doi: 10.1007/s10994-021-06021-7.

[165] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009. doi: 10.1016/j.eswa.2008.01.039.

[166] X. Jin and J. Han, "K-Medoids Clustering," in *Encyclopedia of Machine Learning*, pp. 564–565, Springer, Boston, MA, 2010. doi: 10.1007/978-0-387-30164-8_426.

[167] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, pp. 226–231, 1996.

[168] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017. doi: 10.1145/3068335.

[169] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," *ACM SIGMOD Record*, vol. 28, no. 2, pp. 49–60, 1999. doi: 10.1145/304181.304187.

[170] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases," in *VLDB*, vol. 98, pp. 428–439, 1998.

[171] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin, "A robust EM clustering algorithm for Gaussian mixture models," *Pattern Recognition*, vol. 45, no. 11, pp. 3950–3961, 2012. doi: 10.1016/j.patcog.2012.04.031.

[172] F. Aurenhammer and R. Klein, "Voronoi Diagrams," *Handbook of Computational Geometry*, vol. 5, no. 10, pp. 201–290, 2000.

[173] M. Erwig, "The Graph Voronoi Diagram with Applications," *Networks: An International Journal*, vol. 36, no. 3, pp. 156–163, 2000. doi: 10.1002/1097-0037(200010)36:3<156::AID-NET2>3.0.CO;2-L.

[174] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020. doi: 10.3390/electronics9081295.

[175] J. M. Pena, J. A. Lozano, and P. Larranaga, "An empirical comparison of four initialization methods for the K-Means algorithm," *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027–1040, 1999. doi: 10.1016/S0167-8655(99)00069-0.

[176] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Systems with Applications*, vol. 40, no. 1, pp. 200–210, 2013. doi: 10.1016/j.eswa.2012.07.021.

[177] T. M. Kodinariya and P. R. Makwana, "Review on Determining Number of Cluster in K-Means Clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[178] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding," Technical Report 2006-13, Stanford, 2006.

[179] O. Bachem, M. Lucic, S. H. Hassani, and A. Krause, "Approximate K-Means++ in Sublinear Time," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[180] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010. doi: 10.1016/j.patrec.2009.09.011.

[181] A. Starczewski, P. Goetzen, and M. J. Er, "A New Method for Automatic Determining of the DBSCAN Parameters," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, no. 3, pp. 209–221, 2020. doi: 10.2478/jaiscr-2020-0014.

[182] J. W. Neidigh, R. M. Fesinmeyer, and N. H. Andersen, "Designing a 20-residue protein," *Nature Structural Biology*, vol. 9, no. 6, pp. 425–430, 2002. doi: 10.1038/nsb798.

[183] C. J. McKnight, P. T. Matsudaira, and P. S. Kim, "NMR structure of the 35-residue villin headpiece subdomain," *Nature Structural Biology*, vol. 4, no. 3, pp. 180–184, 1997. doi: 10.1038/nsb0397-180.

[184] L. Qiu, S. A. Pabit, A. E. Roitberg, and S. J. Hagen, "Smaller and Faster: The 20-Residue Trp-Cage Protein Folds in 4 μs," *Journal of the American Chemical Society*, vol. 124, no. 44, pp. 12952–12953, 2002. doi: 10.1021/ja0279141.

[185] A. Hałabis, W. Żmudzińska, A. Liwo, and S. Ołdziej, "Conformational Dynamics of the Trp-Cage Miniprotein at Its Folding Temperature," *The Journal of Physical Chemistry B*, vol. 116, no. 23, pp. 6898–6907, 2012. doi: 10.1021/jp212630y.

[186] P. L. Freddolino and K. Schulten, "Common Structural Transitions in Explicit-Solvent Simulations of Villin Headpiece Folding," *Biophysical Journal*, vol. 97, no. 8, pp. 2338–2347, 2009. doi: 10.1016/j.bpj.2009.08.012.

[187] H. Lee, M. Turilli, S. Jha, D. Bhowmik, H. Ma, and A. Ramanathan, "DeepDriveMD: Deep-Learning Driven Adaptive Molecular Simulations for Protein Folding," in *2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*, pp. 12–19, IEEE, 2019. doi: 10.1109/DLS49591.2019.00007.

[188] H. Lei, C. Wu, H. Liu, and Y. Duan, "Folding free-energy landscape of villin headpiece subdomain from molecular dynamics simulations," *Proceedings of the National Academy of Sciences*, vol. 104, no. 12, pp. 4925–4930, 2007. doi: 10.1073/pnas.0608432104.

[189] W. L. DeLano *et al.*, "PyMOL: An Open-Source Molecular Graphics Tool," *CCP4 Newsletter on Protein Crystallography*, vol. 40, no. 1, pp. 82–92, 2002.

[190] Schrödinger, LLC, "The PyMOL Molecular Graphics System, Version 1.8," November 2015.

[191] R. Jauch, C. K. L. Ng, K. S. Saikatendu, R. C. Stevens, and P. R. Kolatkar, "Crystal Structure and DNA Binding of the Homeodomain of the Stem Cell Transcription Factor Nanog," *Journal of Molecular Biology*, vol. 376, no. 3, pp. 758–770, 2008. doi: 10.1016/j.jmb.2007.11.091.

[192] J. M. Martín-García, I. Luque, P. L. Mateo, J. Ruiz-Sanz, and A. Cámara-Artigas, "Crystallographic structure of the SH3 domain of the human c-Yes tyrosine kinase: Loop flexibility and amyloid aggregation," *FEBS Letters*, vol. 581, no. 9, pp. 1701–1706, 2007. doi: 10.1016/j.febslet.2007.03.059.

[193] S. Chaudhury, S. Lyskov, and J. J. Gray, "PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta," *Bioinformatics*, vol. 26, no. 5, pp. 689–691, 2010. doi: 10.1093/bioinformatics/btq007.

[194] J. J. Gray, S. Chaudhury, and S. Lyskov, "PyRosetta Interactive Molecular Modeling for Proteins: User's Manual." url: https://graylab.jhu.edu/pyrosetta/downloads/documentation/PyRosetta_Manual.pdf, Accessed: 2022-03-14.

[195] J. K. Leman, B. D. Weitzner, S. M. Lewis, J. Adolf-Bryfogle, N. Alam, R. F. Alford, M. Aprahamian, D. Baker, K. A. Barlow, P. Barth, *et al.*, "Macromolecular modeling and design in Rosetta: recent methods and frameworks," *Nature Methods*, vol. 17, no. 7, pp. 665–680, 2020. doi: 10.1038/s41592-020-0848-2.

[196] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker, "Protein Structure Prediction Using Rosetta," *Methods in Enzymology*, vol. 383, pp. 66–93, 2004. doi: 10.1016/S0076-6879(04)83004-0.

[197] K. W. Kaufmann, G. H. Lemmon, S. L. DeLuca, J. H. Sheehan, and J. Meiler, "Practically Useful: What the Rosetta Protein Modeling Suite Can Do for You," *Biochemistry*, vol. 49, no. 14, pp. 2987–2998, 2010. doi: 10.1021/bi902153g.

[198] C. A. Rohl, C. E. Strauss, D. Chivian, and D. Baker, "Modeling Structurally Variable Regions in HomologousProteins With Rosetta," *Proteins: Structure, Function, and Bioinformatics*, vol. 55, no. 3, pp. 656–677, 2004. doi: 10.1002/prot.10629.

[199] R. Trevizani, F. L. Custódio, K. B. Dos Santos, and L. E. Dardenne, "Critical Features of Fragment Libraries for Protein Structure Prediction," *PLOS ONE*, vol. 12, no. 1, p. e0170131, 2017. doi: 10.1371/journal.pone.0170131.

[200] D. J. Lipman and W. R. Pearson, "Rapid and Sensitive Protein Similarity Searches," *Science*, vol. 227, no. 4693, pp. 1435–1441, 1985. doi: 10.1126/science.2983426.

[201] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proceedings of the National Academy of Sciences*, vol. 85, no. 8, pp. 2444–2448, 1988. doi: 10.1073/pnas.85.8.2444.

[202] F. Khatib, S. Cooper, M. D. Tyka, K. Xu, I. Makedon, Z. Popović, and D. Baker, "Algorithm discovery by protein folding game players," *Proceedings of the National Academy of Sciences*, vol. 108, no. 47, pp. 18949–18953, 2011. doi: 10.1073/pnas.1115898108.

[203] R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. Kappel, *et al.*, "The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design," *Journal of Chemical Theory and Computation*, vol. 13, no. 6, pp. 3031–3048, 2017. doi: 10.1021/acs.jctc.7b00125.

[204] A. B. Rubenstein, K. Blacklock, H. Nguyen, D. A. Case, and S. D. Khare, "Systematic Comparison of Amber and Rosetta Energy Functions for Protein Structure Evaluation," *Journal of Chemical Theory and Computation*, vol. 14, no. 11, pp. 6015–6025, 2018. doi: 10.1021/acs.jctc.8b00303.

[205] E. H. Baugh, S. Lyskov, B. D. Weitzner, and J. J. Gray, "Real-Time PyMOL Visualization for Rosetta and PyRosetta," *PLOS ONE*, vol. 6, no. 8, p. e21931, 2011. doi: 10.1371/journal.pone.0021931.

[206] D. Freedman, R. Pisani, and R. Purves, "Statistics: International Student Edition," *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[207] Y. Zhang, H. Wu, and L. Cheng, "Some New Deformation Formulas about Variance and Covariance," in *2012 Proceedings of International Conference on Modelling, Identification and Control*, pp. 987–992, IEEE, 2012.

[208] H. Neuweiler, T. D. Sharpe, T. J. Rutherford, C. M. Johnson, M. D. Allen, N. Ferguson, and A. R. Fersht, "The Folding Mechanism of BBL: Plasticity of Transition-State Structure Observed within an Ultrafast Folding Protein Family," *Journal of Molecular Biology*, vol. 390, no. 5, pp. 1060–1073, 2009.

[209] M. U. Johansson, M. de Château, M. Wikström, S. Forsén, T. Drakenberg, and L. Björck, "Solution Structure of the Albumin-binding GA Module: A Versatile Bacterial Protein Domain," 1997.

[210] L. J. Beamer and C. O. Pabo, "Refined 1.8 Å Crystal Structure of the λ Repressor-Operator Complex," *Journal of Molecular Biology*, vol. 227, no. 1, pp. 177–196, 1992.

[211] J.-H. Cho, W. Meng, S. Sato, E. Y. Kim, H. Schindelin, and D. P. Raleigh, "Energetically significant networks of coupled interactions within an unfolded protein," *Proceedings of the National Academy of Sciences*, vol. 111, no. 33, pp. 12079–12084, 2014.

[212] M. Jäger, Y. Zhang, J. Bieschke, H. Nguyen, M. Dendle, M. E. Bowman, J. P. Noel, M. Gruebele, and J. W. Kelly, "Structure–function–folding relationship in a WW domain," *Proceedings of the National Academy of Sciences*, vol. 103, no. 28, pp. 10648–10653, 2006.

[213] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized Dimensionality Reduction for k-Means Clustering," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 1045–1062, 2014. doi: 10.1109/TIT.2014.2375327.

[214] M. T. Nelson, W. Humphrey, A. Gursoy, A. Dalke, L. V. Kalé, R. D. Skeel, and K. Schulten, "NAMD: a Parallel, Object-Oriented Molecular Dynamics Program," *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 10, no. 4, pp. 251–268, 1996. doi: 10.1177/109434209601000401.

[215] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, "Scalable Molecular Dynamics with NAMD," *Journal of Computational Chemistry*, vol. 26, no. 16, pp. 1781–1802, 2005. doi: 10.1002/jcc.20289.

[216] 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, *et al.*, "Lammps - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Computer Physics Communications*, vol. 271, p. 108171, 2022. doi: 10.1016/j.cpc.2021.108171.

[217] W. Humphrey, A. Dalke, and K. Schulten, "VMD: Visual Molecular Dynamics," *Journal of Molecular Graphics*, vol. 14, no. 1, pp. 33–38, 1996. doi: 10.1016/0263-7855(96)00018-5.

[218] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, "UCSF Chimera — A Visualization System for Exploratory Research and Analysis," *Journal of Computational Chemistry*, vol. 25, no. 13, pp. 1605–1612, 2004. doi: 10.1002/jcc.20084.

[219] E. C. Meng, E. F. Pettersen, G. S. Couch, C. C. Huang, and T. E. Ferrin, "Tools for integrated sequence-structure analysis with UCSF Chimera," *BMC Bioinformatics*, vol. 7, no. 1, pp. 1–10, 2006. doi: 10.1186/1471-2105-7-339.

[220] Richard J. Gowers, Max Linke, Jonathan Barnoud, Tyler J. E. Reddy, Manuel N. Melo, Sean L. Seyler, Jan Domański, David L. Dotson, Sébastien Buchoux, Ian M. Kenney, and Oliver Beckstein, "MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations," in *Proceedings of the 15th Python in Science Conference* (Sebastian Benthall and Scott Rostrup, eds.), pp. 98 – 105, 2016. doi: 10.25080/Majora-629e541a-00e.

[221] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, "MDAnalysis: A Toolkit for the Analysis of Molecular Dynamics Simulations," *Journal of Computational Chemistry*, vol. 32, no. 10, pp. 2319–2327, 2011. doi: 10.1002/jcc.21787.

[222] 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, and V. S. Pande, "MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories," *Biophysical Journal*, vol. 109, no. 8, pp. 1528–1532, 2015. doi: 10.1016/j.bpj.2015.08.015.

[223] O. Beckstein, A. Somogyi, E. Denning, J. Domanski, R. Gowers, A. White, D. Dotson, P. Lacerda, I. Kenney, M. Mohebifar, P. Loche, M. Linke, T. Heavey, A. Berg, and Y. Polyachenko, "Becksteinlab/GromacsWrapper: Release 0.8.2," sep 2021. doi: 10.5281/zenodo.5498364.

[224] H. Nguyen, D. A. Case, and A. S. Rose, "NGLview–interactive molecular graphics for Jupyter notebooks," *Bioinformatics*, vol. 34, no. 7, pp. 1241–1242, 2018. doi: 10.1093/bioinformatics/btx789.

[225] N. Schmid, A. P. Eichenberger, A. Choutko, S. Riniker, M. Winger, A. E. Mark, and W. F. van Gunsteren, "Definition and testing of the GROMOS force-field versions 54A7 and 54B7," *European Biophysics Journal*, vol. 40, no. 7, pp. 843–856, 2011. doi: 10.1007/s00249-011-0700-9.

[226] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, "Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids," *Journal of the American Chemical Society*, vol. 118, no. 45, pp. 11225–11236, 1996. doi: 10.1021/ja9621760.

[227] M. Bonomi, D. Branduardi, G. Bussi, C. Camilloni, D. Provasi, P. Raiteri, D. Donadio, F. Marinelli, F. Pietrucci, R. A. Broglia, *et al.*, "PLUMED: A portable plugin for free-energy calculations with molecular dynamics," *Computer Physics Communications*, vol. 180, no. 10, pp. 1961–1972, 2009. doi: 10.1016/j.cpc.2009.05.011.

[228] G. A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni, and G. Bussi, "PLUMED 2: New feathers for an old bird," *Computer Physics Communications*, vol. 185, no. 2, pp. 604–613, 2014. doi: 10.1016/j.cpc.2013.09.018.

[229] H. Sidky, Y. J. Colón, J. Helfferich, B. J. Sikora, C. Bezik, W. Chu, F. Giberti, A. Z. Guo, X. Jiang, J. Lequieu, *et al.*, "SSAGES: Software Suite for Advanced General Ensemble Simulations," *The Journal of Chemical Physics*, vol. 148, no. 4, p. 044104, 2018. doi: 10.1063/1.5008853.

[230] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, *et al.*, "Jupyter Notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, vol. 2016, 2016. doi: 10.3233/978-1-61499-649-1-87.

[231] J. M. Perkel, "Why Jupyter is data scientists' computational notebook of choice," *Nature*, vol. 563, no. 7732, pp. 145–147, 2018. doi: 10.1038/d41586-018-07196-1.

[232] E. Reynaud *et al.*, "Protein Misfolding and Degenerative Diseases," *Nature Education*, vol. 3, no. 9, p. 28, 2010.

[233] Y. Cheng, N. Grigorieff, P. A. Penczek, and T. Walz, "A Primer to Single-Particle Cryo-Electron Microscopy," *Cell*, vol. 161, no. 3, pp. 438–449, 2015. doi: 10.1016/j.cell.2015.03.050.

[234] S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman, "The Weighted Histogram Analysis Method for Free-Energy Calculations on Biomolecules. I. The Method," *Journal of Computational Chemistry*, vol. 13, no. 8, pp. 1011–1021, 1992. doi: 10.1002/jcc.540130812.

[235] M. B. Zerihun, F. Pucci, and A. Schug, "CoCoNet—boosting RNA contact prediction by convolutional neural networks," *Nucleic Acids Research*, vol. 49, no. 22, pp. 12661–12672, 2021. doi: 10.1093/nar/gkab1144.

[236] S. Fulle and H. Gohlke, "Analyzing the Flexibility of RNA Structures by Constraint Counting," *Biophysical Journal*, vol. 94, no. 11, pp. 4202–4219, 2008. doi: 10.1529/biophysj.107.113415.

[237] D. Tan, S. Piana, R. M. Dirks, and D. E. Shaw, "RNA force field with accuracy comparable to state-of-the-art protein force fields," *Proceedings of the National Academy of Sciences*, vol. 115, no. 7, pp. E1346–E1355, 2018. doi: 10.1073/pnas.1713027115.

[238] A. Patriksson and D. van der Spoel, "A temperature predictor for parallel tempering simulations," *Physical Chemistry Chemical Physics*, vol. 10, no. 15, pp. 2073–2077, 2008. doi: 10.1039/B716554D.

[239] M. Zgarbová, M. Otyepka, J. Šponer, A. Mládek, P. Banáš, T. E. Cheatham, and P. Jurecka, "Refinement of the Cornell et al. Nucleic Acids Force Field Based on Reference Quantum Chemical Calculations of Glycosidic Torsion Profiles," *Journal of Chemical Theory and Computation*, vol. 7, no. 9, pp. 2886–2902, 2011. doi: 10.1021/ct200162x.

[240] M. Zgarbová, J. Šponer, M. Otyepka, T. E. Cheatham, R. Galindo-Murillo, and P. Jurecka, "Refinement of the Sugar–Phosphate Backbone Torsion Beta for AMBER Force Fields Improves the Description of Z- and B-DNA," *Journal of Chemical Theory and Computation*, vol. 11, no. 12, pp. 5723–5736, 2015. doi: 10.1021/acs.jctc.5b00716.

[241] S. Thore, C. Frick, and N. Ban, "Structural Basis of Thiamine Pyrophosphate Analogues Binding to the Eukaryotic Riboswitch," *Journal of the American Chemical Society*, vol. 130, no. 26, pp. 8116–8117, 2008. doi: 10.1021/ja801708e.

[242] J. Zhang and A. R. Ferré-D'Amaré, "Dramatic Improvement of Crystals of Large RNAs by Cation Replacement and Dehydration," *Structure*, vol. 22, no. 9, pp. 1363–1371, 2014. doi: 10.1016/j.str.2014.07.011.

[243] A. Ren, X. C. Wang, C. A. Kellenberger, K. R. Rajashankar, R. A. Jones, M. C. Hammond, and D. J. Patel, "Structural Basis for Molecular Discrimination by a 3',3'-cGAMP Sensing Riboswitch," *Cell Reports*, vol. 11, no. 1, pp. 1–12, 2015. doi: 10.1016/j.celrep.2015.03.004.