UNSUPERVISED LEARNING FOR MOLECULAR STRUCTURE DISCOVERIES

by

Kazi Lutful Kabir
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computer Science

Committee:

_____ Dr. Amarda Shehu, Dissertation Director

_____ Dr. Zoran Duric, Committee Member

_____ Dr. Shuochao Yao, Committee Member

_____ Dr. Wanli Qiao, Committee Member

_____ Dr. David S. Rosenblum, Department Chair

Date: _____ Summer 2022
George Mason University
Fairfax, VA

Unsupervised Learning for Molecular Structure Discoveries

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Kazi Lutful Kabir
Master of Science
George Mason University, 2019
Bachelor of Science
Military Institute of Science and Technology, 2014

Director: Dr. Amarda Shehu, Professor
Department of Computer Science

Summer 2022
George Mason University
Fairfax, VA

Acknowledgments

Firstly, I would like to thank my advisor, Prof. Amarda Shehu, for her continuous guidance, untiring support, prolific mentoring, and cordial assistance throughout my graduate studies and research. She has made a significant contribution towards improving my efficacy in addressing research problems and my technical expressions of the strategies to approach them. Her constructive and detailed feedback on my work has always been immensely productive. I would also like to thank one of our recent lab alumni, Dr. Nasrin Akhter, for her insightful discussions on my work from time to time. I would also like to thank my internship mentors at Los Alamos National Laboratory (LANL): Dr. Gopinath Chennupati, Dr. Boian Alexandrov, Dr. Raviteja Vangara, Dr. Hristo Djidjev, and Dr. Manish Bhattarai for providing me wonderful opportunities to expand the scope of research. I would also like to thank Dr. Ruth Nussinov from National Cancer Institute (NCI) for collaborating in one of the projects and providing us with the molecular dynamics simulation data of the complex antibody systems. I would also like to thank all the researchers within this community whose articles, tools, and software have facilitated my research and advance my work in computational biology and unsupervised learning. I owe special thanks to my mother, my wife, and rest of my family members for their sacrifices and unconditional support. Finally, I would like to thank Prof. Zoran Duric, Prof. Shuochao Yao, and Prof. Wanli Qiao for their constructive feedback and willingness to serve on my Ph.D. dissertation committee.

Table of Contents

Page

List of Tables . . . viii
List of Figures . . . ix
Abstract . . . xi
1 Introduction . . . 1
1.1 Problem Statement . . . 4
1.2 Contribution . . . 4
2 Driving Problems . . . 7
2.1 From Structures to Markov State Models of (Structural) Dynamics . . . 7
2.1.1 Molecular Dynamics Simulation . . . 7
2.2 Estimation of Model Accuracy (EMA) . . . 9
3 Preliminaries . . . 14
3.1 Protein Architecture . . . 14
3.2 Representation . . . 15
3.2.1 Cartesian Coordinates . . . 15
3.2.2 Dihedral Angles . . . 15
3.2.3 Coarse-grained Representation . . . 16
3.3 Domain-Specific Proximity/Distance Measure . . . 18
3.4 Internal Structure Energy: Potential Energy Functions . . . 20
3.5 Interaction Energy . . . 20
3.6 Energy Landscape . . . 22
4 Graph-embedding Methods to Organize Protein Structure Space . . . 24
4.1 A Graph-based Embedding of Structures . . . 24
4.2 Organize Structures into Communities . . . 25
4.2.1 Community Detection Methods . . . 26
4.2.2 Metrics for Evaluating Community Detection Methods . . . 28
4.3 Selection Strategies to Rank the Groups/Communities . . . 31
4.4 Evaluation Dataset and Metric . . . 32
4.4.1 Evaluation Metric . . . 33

4.5 Evaluation Results . . . 34
4.6 Summary . . . 40
5 Organization of Structure Space to Structural Dynamics . . . 42
5.1 Application Setting . . . 42
5.2 Organizing Structures in MD Trajectories . . . 44
5.2.1 From MD Trajectories to a Markov State Model (MSM) . . . 44
5.2.2 State Identification via Clustering over a Graph Embedding of Structures . . . 46
5.2.3 State Identification via Detection of Energy Landscape Basins . . . 47
5.2.4 MSM Construction . . . 48
5.3 Evaluation Dataset . . . 49
5.4 Evaluation of MSM . . . 49
5.4.1 Convergence Analysis . . . 49
5.4.2 MSM Model Visualization . . . 52
5.4.3 Relating Long-lived States to Experimentally-known Structures . . . 53
5.5 Summary . . . 54
6 Feature-oriented Factorization Methods to Organize Protein Structure Space . . . 55
6.1 Non-negative Matrix Factorization (NMF)-based Framework to Organize Protein Structure Space . . . 55
6.1.1 Non-negative Matrix Factorization (NMF) . . . 57
6.1.2 Feature Extraction . . . 58
6.1.3 NMF-backed Structure Groups . . . 60
6.1.4 Structure-Group Selection . . . 61
6.1.5 Functionally-relevant Structure Selection . . . 62
6.2 Evaluation Dataset . . . 63
6.3 Evaluation Results . . . 65
6.3.1 Group Selection Results . . . 65
6.3.2 Single Structure Selection Results . . . 68
6.4 Summary . . . 70
7 Non-parametric, Feature-free Factorization-based Method to Organize Protein Structure Space . . . 71
7.1 Symmetric Non-negative Matrix Factorization (SNMF)-based Framework to Organize Protein Structure Space . . . 71
7.1.1 From Structure Ensemble to Structure Similarity Matrix . . . 72
7.1.2 From Structure Similarity Matrix to number of Structure Groups . . . 73

7.1.3 Organizing Structures into Groups 75
7.1.4 Determining the Best Structure-Group 76
7.1.5 Determining the Best Structure in a Group 76
7.2 Evaluation Dataset 77
7.3 Evaluation Results 79
7.3.1 Running Time Comparison 79
7.3.2 Group Purity Comparison 80
7.3.3 Loss Comparison 82
7.3.4 Statistical Significance Analysis 83
7.4 Summary 85
8 Tensor Decomposition-based Method 86
8.1 Non-negative Tensor Factorization (NTF)-based Framework to Organize Protein Structure Space 86
8.1.1 Stage I: From Structures to Groups 87
8.1.2 Stage II: Ranking Groups 90
8.1.3 Stage III: Partitioning Groups into Subgroups 90
8.1.4 Stage IV: Scoring each Structure 90
8.2 Experimental Setup and Evaluation Results 91
8.2.1 Evaluation Metrics 92
8.2.2 Comparative Evaluation on Correlation with TM-Score 92
8.2.3 Loss-based Comparison 93
8.2.4 Statistical Significance Analysis 94
8.3 Summary 95
9 Summarization of Structural Dynamics of Antibody System 96
9.1 Application Setting and Objective 96
9.2 Organizing Structures of Antibody-antigen bound Molecular System 97
9.2.1 From MD Trajectories to an MSM of Dynamics 97
9.2.2 Conformation Generation and Preparation 101
9.2.3 Preparation of Conformations for Analysis 101
9.2.4 Evaluation Setup 104
9.3 Evaluation Results 104
9.3.1 Visual Comparison of Embedded Landscapes 104
9.3.2 Basin Analysis 106
9.3.3 Summarization and Comparison of Dynamics 107
9.3.4 Visualization of the Largest Basins. 110

9.3.5 Discussion 110
9.4 Summary 110
10 Conclusion and Future Work 112
Bibliography 115

List of Tables

Table Page

3.1 Energy Terms in REF15 Energy Function 21
4.1 Rosetta Generated Structure Ensemble Dataset of Proteins 33
4.2 Rank by different Selection Strategies of the Community with the highest purity 38
4.3 Comparison of **Sel-S+E** to other selection strategies on best rank via **1-sided** Fisher's and Barnard's tests 39
4.4 Comparison of **Sel-S+E** to other selection strategies on best rank via **2-sided** Fisher's and Barnard's tests. 40
5.1 Comparison of Structures in the states with the experimentally-identified NMR ensembles 53
6.1 Targets in the Benchmark Dataset 63
6.2 Targets in the CASP Dataset 64
6.3 Quantitative Comparison on CASP targets 68
7.1 Targets in the Benchmark Dataset 78
7.2 Targets in the CASP Dataset 79
7.3 Running Time Comparison 80
7.4 Loss Comparison: Benchmark Targets 82
7.5 Loss Comparison: CASP Targets 83
7.6 Statistical significance of different methods via Friedman's tests with Hommel's post-hoc analysis 84
8.1 Target-wise Pearson correlation with respect to true TM-Score 92
8.2 Comparison of NTF-REL, SNMF-DS, ProQ3D, ProQ4, and NMF-MAD on RMSD, TM-Score, and GDT-TS loss 93
8.3 Statistical significance of various methods determined via Friedman's tests with Hommel's post-hoc analysis 94

List of Figures

Figure Page

1.1 Primary sequence (of amino acids) and experimentally-resolved tertiary structure of the hemoglobin protein 2
3.1 Backbone Dihedral Angles 16
3.2 Reference Points for USR Metrics 17
4.1 Comparison of Community Detection Algorithms 35
4.2 Comparison of the selection strategies on the purity of the top community . 36
4.3 Comparison of the selection strategies on the purity of the top three communities 37
5.1 Schematic of the computational pipeline for converting MD trajectories into MSM of System's dynamics 44
5.2 Implied timescale plots for state space discretization via different methods . 51
5.3 Visualization of the best model of dynamics 52
6.1 Schematic of the NMF-based framework 56
6.2 Illustration of NMF decomposition of the feature matrix 57
6.3 Comparison of four unsupervised basin-based and two NMF-based structure selection methods 65
6.4 Comparison of two NMF-based structure selection methods and MUFOLD-CL on CASP targets 67
6.5 Superimposition of the Structures under each difficulty category (easy, medium, hard) selected by NMF-MAD over known wet-laboratory structures under PDB ID 1tig, 1hz6(A), and 1cc5 69
7.1 The schematics of the framework operationalized by SNMF-DS 71
7.2 The EigenGap curve computed over structures of CASP Target `T1008-D1` . 74
7.3 An illustration of the factorization of a symmetric non-negative similarity matrix 75
7.4 The purity of the group/cluster selected by SNMF-DS, NMF-MAD, and MUFOLD-CL over the benchmark targets 80

7.5 The purity of the group/cluster selected by SNMF-DS, NMF-MAD, and MUFOLD-CL over the CASP targets 81
8.1 (a) Schematic of NTF-REL, (b) Finding the number of latent features with non-negative RESCAL factorization. 87
8.2 The stability analysis for one of the CASP protein targets, `T1008-D1` 89
9.1 Embedding of $5,000$ sampled conformations (selected every 32nd frame in MD trajectories) for both antibody and antigen-antibody complex 105
9.2 The minimum RMSD (Å) (over backbone atoms) between the focal minimum conformation representing a basin and the initial conformations of the MD trajectories 106
9.3 Pie chart of adjusted state populations of the stationary distribution for the 6 top-populated macrostates/basins & MSM dynamics with transitions between basins 108
9.4 Focal minima conformations corresponding to basins B1 to B6 109

Abstract

UNSUPERVISED LEARNING FOR MOLECULAR STRUCTURE DISCOVERIES

Kazi Lutful Kabir, PhD

George Mason University, 2022

Dissertation Director: Dr. Amarda Shehu

We have long known that form determines function. This is particularly true of biological molecules, which utilize their three-dimensional structures to interface with one another and propagate chemical reactions in the living cell. We also now better understand how vast and rich the structure space available to a molecule is and how little we know about what information to extract from this space to better characterize the structure(s)-function(s) relationship in biological molecules. This dissertation puts forth computational concepts and techniques to support this goal. Particularly, we develop algorithms to organize the structure space of a molecule and reveal one or more important structural states of small molecules, macromolecules, and complexated molecules. The algorithms proposed here fall under the umbrella of unsupervised learning but leverage explicit or implicit embeddings of molecular structures in discrete data-structures, such as graphs, to better utilize proximity in structure space for capturing structural states. The proposed algorithms employ diverse formalizations and show the power of those formalizations in addressing increasingly complex problems and application settings. Rigorous evaluation on hallmark problems in computational structural biology suggests that the leveraged formalizations and proposed algorithms advance research on unsupervised learning of the organization of molecular structure spaces.

While the focus of the algorithms presented here is on molecular structure data, the techniques we describe are of general utility to any domain where the ultimate objective is to obtain informative organizations of high-dimensional spatial data.

Chapter 1: Introduction

Molecules are three-dimensional objects. Their building units, atoms, occupy positions in three-dimensional space. The different spatial arrangements of the connected set of atoms in a molecule give rise to what we refer to as tertiary structures. Having this three-dimensional understanding of a molecule, in addition to its chemical formula, is central to understanding molecular function, as molecules use their structures to complement, stick to, and so interact with one another [1]. Summarily said, structure *carries* function [2].

We have known the central role that molecular structure plays for decades, particularly for biological macromolecules (to which refer as biomolecules), such as proteins, RNA, and DNA. Fig. 1.1 relates the tertiary structure of the hemoglobin protein, a molecule that carries oxygen in our blood and whose mutations are implicated in a large number of blood disorders. The left panel of Fig. 1.1 shows the string of the building units, the amino acids, that comprise a protein molecule. They are abbreviated into one-letter codes. Each letter hides anywhere from a few to a dozen atoms. This string of amino acids "hold" hands via peptide bonds, giving rise to a highly flexible chain. The right panel of Fig. 1.1 shows one spatial arrangement, one tertiary structure, through which hemoglobin is able to bind oxygen.

What we know of proteins is that, indeed, they are inherently dynamic/plastic. While experiments are able to capture them in one or at most a few tertiary structures, we now know that protein molecules have great structural plasticity [3]. They leverage their ability to assume different tertiary structures in the cell to regulate interactions with other molecular partners and control many cellular processes [4]. In the words of Richard Feynman [5], "Everything that living things do can be understood in terms of the jiggling and wiggling of atoms."

Figure 1.1: Primary sequence (of amino acids) and experimentally-resolved tertiary structure of the hemoglobin protein [6]

Simulation (via physics-based models that track movements of atoms in response to physical forces) is often used to capture these changes. Other methods not based on physics-based models but operating under the umbrella of stochastic optimization [7, 8] have also been shown powerful in complementing experiments and providing a broader view of the large, high-dimensional structure space of a dynamic protein molecule.

Unraveling the organization of the protein structure space provides valuable insights into connecting structure(s) with function. In particular, the dynamic nature of a protein molecule is captured in the concept of a structural state. A state is a homogeneous set/group of structures. In the context of machine learning, we can think of a state as a cluster. Indeed, clustering algorithms have been popular in organizing structures of a molecule into states. However, the actual concept of a state utilizes not just proximity in the structure space but additionally energy. The inherent dynamics of a molecule is more evident when one considers the energy landscape. In its simplest exposition, the energy landscape is a lifted space; that is, the structure space is associated with an additional dimension of energy by evaluating/associating an energetic value/score with each structure. When considering an uncomplexed molecule (in isolation, not bound to others), we evaluate a structure by its internal energy, also known as potential energy. We describe this in greater detail in

Chapter 3, where we relate some preliminaries, but the main idea is that physics-based interactions among atoms in a particular structure can be summed up to associate with a structure an energy value. A state is then a group of structurally- and energetically-similar structures. The energy values distinguish a thermodynamically-stable state from a meta-stable state.

What we often observe in wet laboratories is a stable state. A molecule dwells longer in a stable state. In the energy landscape, a state corresponds to a broad and deep basin or valley [9]. An extended amount of time needs to pass to collect sufficient kinetic energy via small thermal vibrations for the molecule to navigate outside of a basin; that is, escape a stable state. What we do not observe in wet laboratories is the diversity of stable and semi-stable states. The main reason is that a molecule does not stay in a semi-stable state for too long. However, such states are particularly rich in information, as they provide us with a mechanistic understanding of how exactly a molecule switches between two different stable states [9]. Such states may be further stabilized by the presence of a binding molecule; therefore, obtaining a broad view of the diversity of stable and semi-stable structural states is essential to better connecting structure to function [4].

As we have laid out earlier, the energy landscape provides an inherent organization of the structure space by taking into account both the proximity of structures and their energetics [10]. Routinely in this dissertation, we will use the terms energy landscape and structure-energy space interchangeably so as to remind readers of the pairing of structures with energies. The energy landscape can be leveraged to expose the relationship between protein structure, dynamics, and function [10, 11]. However, no computational method reveals the energy landscape explicitly; instead, what one draws in *silico* is an ensemble of energy-evaluated structures that constitute points sampled from the landscape. These points may not accurately represent certain regions of the landscape due to sample bias of the methods that have to allocate limited computational resources to explore large, high-dimensional search spaces related to the corresponding structure spaces [12].

1.1 Problem Statement

A fundamental question presents itself: given a set of computed tertiary structures of a molecule, which provide us with a limited and possibly biased view of the structure space, and with possibly associated energies, which provide us with a limited and biased view of the energy landscape, how can we dissect the information available and expose the structural states present in the absence of any *a priori* knowledge? This question underlies two hallmark problems that drive the conceptualization and design of the algorithms proposed here. These problems are summarized in Chapter 2, and the preliminaries essential for readers to understand the molecular structure, architecture, structure representation, and benchmark proximity metrics for comparing tertiary structures are related in Chapter 3.

1.2 Contribution

This dissertation approaches answers to the above fundamental question under the umbrella of machine learning. In particular, the dissertation advances computational work in unsupervised learning for organizing the tertiary structure space (and, where available, the associated energy landscape) of a molecule to identify structural states relevant to biological function, so that we can learn directly from structure data. Specifically, contributions are made along the following dimensions:

- In Chapter 4, we address the setting when energies are not available, and we only have access to structure data. Effectively, we harness tertiary structures only as a first-order approximation of our answer to the question formulated above. Making connections with graph mining, we investigate graph embeddings of structures sampled in *silico* over the structure space of a protein molecule. In particular, we leverage graph-based clustering algorithms and evaluate the ability of community detection algorithms to organize tertiary structures of a protein molecule into communities/groups. We assess the quality and relevance of these groups with known functionally-relevant structures

and elucidate top-performing community detection algorithms in a comparative setting.

- Chapter 5 focuses on the tertiary structures of a small molecule obtained via physics-based simulation. In this setting, we leverage both structures and internal structure energies provided by the simulation platform to organize structures into groups corresponding to basins. Building over Markov State Models, where we draw analogies between basins and Markov states, we build a discrete model of dynamics that quantifies the state-to-state transitions at equilibrium, revealing precious mechanistic information about a molecule. We compare and evaluate the impact of various algorithms for identifying states into the quality of the resulting state model and show that considering energy and so utilizing basins as states provide us with better models of dynamics.

- In Chapter 6, we start a new thread of research that improves upon graph embeddings on both time and memory demands while still leveraging adjacency matrix information. We leverage a matrix factorization formulation and show its power in addressing a challenging problem setting. In addition to evaluating the quality of obtained structure groups, we also perform a rank-based selection and show that the resulting method outperforms basin-based and other state-of-the-art methods.

- Buoyed by these results, in Chapters 7 and 8, we capitalize further on this thread of research via novel matrix factorization-based methods. In particular, we propose novel tensor-factorization methods. We outline several ideas in detail, as well as specific ways forward on enhancing these methods for application settings where, in addition to evaluating structure groups, we evaluate (and score) single structures for their relevance. This is an important contribution that allows us to place our work in a broader context and additionally compare with existing state-of-the-art methods that do not leverage unsupervised learning but devise novel scoring functions and utilize them to score (and select) individual structures in a structure set.

- In Chapter 9, we expand on our evaluation and application settings. In particular, we organize complexated structures of antibody-antigen bound molecular systems and show the power of the methods we have developed in exposing valuable information on the role of the structure and dynamics in unbound and and bound antibodies. Finally, Chapter 10 concludes this dissertation by highlighting the major contributions and providing future research directions.

Chapter 2: Computational Structural Biology Problems Driving this Dissertation

The concepts and algorithms we present in this dissertation are motivated by two hallmark problems in computational structural biology. They both necessitate the organization of the structures available to a molecule as a primary step, but for different downstream tasks. Since these problems not only motivate our methods but also determine the evaluation setting for what we propose in various chapters of this dissertation, we describe them in detail here for the interested reader.

2.1 From Structures to Markov State Models of (Structural) Dynamics

Wet laboratory techniques have been able to elucidate the exquisitely complex equilibrium dynamics of many biomolecules. Despite significant progress in single-molecule techniques, they can only strike a few snapshots of a biomolecule transitioning between structures while navigating its structure space [2]. In this regard, the results obtained from simulation studies have been very informative. The structure space of a protein molecule accessed via MD simulation can be leveraged to build discrete models of protein structure dynamics.

2.1.1 Molecular Dynamics Simulation

Molecular dynamics (MD) simulations speculate how every atom in a protein or a molecular system of interest will move over time according to a general model of the physics regulating inter-atomic interactions [13]. The basic idea behind an MD simulation is the following: given the positions of all atoms in a biomolecular system, it is possible to calculate the force

exerted on each atom by all of the other atoms. Hence, Newton's laws of motion can be used to speculate the spatial position of each atom as a function of time [14]. Specifically, step by step through time, repeated calculation of the forces on each atom and then leveraging those forces to update the position and velocity of each atom. The resulting trajectory is, fundamentally, a series of three-dimensional snapshots that describes the atomic-level configuration of the system at each point during the simulated time interval [14]. In particular, each trajectory is a list of structures accessed consecutively during the simulation, in time steps of a specific magnitude. The forces in an MD simulation are calculated using a molecular mechanics force field model. The commonly used molecular dynamics simulation software include AMBER [15], GROMACS [16], NAMD [17], CHARMM [18].

An MD simulation provides a local view of the structure space of a molecule. To capture the structural dynamics of the molecule of interest, we need to find the answer to the question that is how to integrate the trajectories obtained from MD simulations. Moreover, extracting and quantifying equilibrium dynamics from MD simulations remains a challenge [19]. An MD simulation provides a limited sampling of the structure space accessed at equilibrium by a molecule of interest. As a result, many MD simulations are required to obtain a broader view of the structure space. Hence, summarizing and quantifying the equilibrium dynamics of a molecule require deriving information from various MD trajectories. In the past decade, Markov State Models (MSM) have emerged as a tool to do so [19, 20].

Building an MSM involves organizing the structures (microstates) accessed in simulation into structural states (macrostates). Once the macrostates are identified, transitions observed between structures in simulation can be mapped to transitions between structural states, and counts of state-to-state transitions can be mathematically converted into transition probabilities [20]. In this multi-stage process, a crucial step that regulates the quality of the derived MSM is the determination of states.

2.2 Estimation of Model Accuracy (EMA)

As presented in Chapter 1, computational methods are now considered to be precious tools in the determination of tertiary structures of molecules. There is a long, multi-decade long history of such methods, which first started with lattice-based modeling approaches that discretized atomic positions on a lattice/grid and utilized combinatorial optimization, to then methods that relaxed these conditions and could handle continuous 3D space but resorted to stochastic optimization, such as Rosetta [21], and then all the way to deep learning-based methods, such as AlphaFold2 [22], leveraging the precious information in similar sequences of proteins to infer proximal atoms in 3D space. All these methods effectively sample the structure space of a protein molecule and give us a discrete (sample-based) representation of this space. Not all the structures are relevant; many may be physically unrealistic and so functionally-irrelevant.

Rosetta [21] is a popular platform that we leverage in our work to generate structures that effectively are samples of a protein's structure space. The Rosetta structure generation procedure operates in 4 subsequent phages where each one is a single trajectory Metropolis Monte Carlo (MMC) search [23], and the final structure obtained as the byproduct of each phase is utilized as the initial structure for the next one. Each pass for the MMC search is fundamentally a molecular fragment replacement operation. The four phases proceed as follows:

- Phase 1 forms an extended chain from the amino-acid sequence by setting the backbone dihedral angles (described in Section 3.2.2) to corresponding characteristic values. Then, it performs a number (20,000) of MMC moves, where each move suggests replacing a fragment of length 9 in a current structure with a fragment sampled from a library pre-compiled over known structures (from a structure database). The process is driven by the Rosetta $score0$ energy function, which penalizes self-collisions of the amino-acid chain in 3D.

- After a structure is obtained at the end of the steps mentioned above, phase 2 repeats

the entire process with a more ambitious energy function, *score*1, which includes additional energetic terms to promote the formation of secondary structure elements.

- Phase 3 switches to the *score*2 energy function, again adding more energetic terms.
- Finally, phase 4 shifts to fragment length 3 in order to make more fine-grained structural changes. It also switches to the *score*3 energy function and proceeds for $12,000$ moves to optimize the structure at the coarse-grained level [24].

The result of the process described above is a low-energy structure that may or may not be functionally-relevant. After all, the structure space is vast, and an MMC trajectory may converge to a different local minimum in the structure-energy space. Typically, one repeats the above process n times to obtain n structures, which constitute an ensemble or set that are a sample-based representation of the structure space of a molecule.

It is necessary to assess the quality/accuracy of computed structures generated by computational methods for a protein molecule. The primary assumption is that the protein molecule has a *biologically-active* state. Even though this is not an accurate assumption that ignores dynamics and the ability of many proteins to carry out many functions in the cell, it is a necessary (albeit insufficient) step in connecting structure with function. Nonetheless, analyzing a structure set and identifying one *best* structure is still a challenging problem in computational (structural) biology and bioinformatics [25]. The problem is known as the estimation of model/structure accuracy (EMA) [26].

A method addressing this problem can be characterized as selecting a subset of structures or selecting an individual structure from a given structure dataset. In the former, such methods can be evaluated in terms of purity, a metric originally introduced in [27]. In the latter, methods can be evaluated via loss, a classic machine learning metric that we adopt in [28–30].

Methods in the first category, which include clustering-based methods, organize structures in a given dataset into groups. These groups can be ranked/ordered based on characteristics that can be measured over a group. For instance, one such characteristic can

be size (number of structures in the group). After ordering the groups by rank, one can take the first l groups and offer them as the set that is most likely to contain functionally-relevant structure. Given an experimentally-known structure and a distance threshold, we can measure the purity of the set by computing the dissimilarity of each structure of the set with respect to the experimentally-known structure.

Methods in the second category select structures directly. It is worth noting that one can easily put together a pipeline that follows up a method from the first category with a method from the second category. For instance, after selecting first a subset B of structures from a given dataset, uniform random sampling can be employed to select any structure from B and offer for prediction. We propose loss to evaluate how good a selected structure is. The structure that is closest to the experimentally-known structure (according to some proximity measure) has a loss of zero. A perfect method would always find such a structure.

Currently, there is great diversity among structure selection methods. Based on the approach they follow, these methods can be roughly grouped into single-model, multi-model, and quasi-single model methods. Single-model methods work on a per structure basis [31] and employ energy functions designed specifically to aid structure selection. Some of these methods use physics-based functions relying on the physical properties of atomic interactions [18, 32, 33]. Others use knowledge-based/statistical scoring functions that rely on the statistical analysis of experimentally known structures [34–36]. The latter methods have been more successful [37,38]. Clustering-based methods, on the other hand, do not rely on energy or scoring functions. They group together similar structures and offer the largest c clusters as prediction. Some recent work has leveraged concepts, such as communities, from network science to cluster structures [39]. These methods construct clusters as communities as in social networks.

Until very recently, clustering-based methods decidedly outperformed single-model methods [40]. However, single-model methods have progressed considerably, to the point that

they can now compete with clustering-based methods [41]. Since the most successful single-model methods rely on specially-designed scoring functions that users often have to re-implement, clustering-based methods remain more popular. Clustering-based methods pose their own concerns, some of which are addressed in [42–44]. Most notably, they suffer from the curse of dimensionality [45] and carry significant computational costs with structure data of increasing size. Since they are based on consensus, they have a very hard time identifying good structures in sparse, low-quality structure datasets, where structures similar to the functionally-relevant one are significantly under-sampled by structure generation algorithms.

In recent years, quasi-single model methods and supervised learning methods have taken hold in the community. These methods currently outperform clustering-based methods. Quasi-single model methods combine concepts of single- and multi-model methods [46, 47]. They work by comparing structures to some selected, high-quality reference structures [48]. Methods based on supervised learning are currently quite diverse, leveraging SVMs [49, 50], Random Forest [51], Neural Networks [52, 53], and ensemble learning [54]. Feature sets are also diverse, derived from terms of statistical scoring functions [55, 56] and/or expert-constructed structural features [57, 58]. These methods show great promise. Inspired by outstanding performance in image recognition, structure selection research has adopted deep learning strategies. For instance, Cao et al. [53] propose DeepQA, a single-model structure selection method that utilizes energy, structural, and physio-chemical characteristics of a structure for quality prediction. Improved structure selection has also been observed with models based on convolutional neural networks (CNNs). For instance, Hou et al. [59] use a deep one-dimensional CNN (1DCNN) to build a single-model structure selection method. The authors make use of two 1DCNNs to predict the local and global quality of a structure. In [60], the authors propose Ornate, a single-model method that applies a deep three-dimensional CNN (3DCNN) for model quality estimation. 3DCNN has also been used successfully in [61]. Hou et al. observed a substantial improvement in protein model selection by using contact distance predicted via a deep CNN [62]. These methods

are very promising, but they are still challenged by the scarcity of labeled data, imbalanced data distribution, and more.

Chapter 3: Preliminaries and Background

Protein molecules feature prominently in this dissertation. So, we first summarize background information on protein structure and representation that is generally informative for other molecular modeling. In addition, since the foundational building block is the ability to compare two structures, we then provide an overview of benchmark dissimilarity and similarity measures for molecular structures.

3.1 Protein Architecture

Proteins are composed of small organic molecules called amino acids. That means the fundamental building block in a protein molecule is the amino acid. Small to medium-sized proteins can have 50-300 amino acids. An amino acid can have dozens of atoms. All amino acids have a common set of atoms known as the backbone and where they are different is in the set of atoms that hang out of the main/central alpha carbon, known as the side chain. There are 20 naturally-occurring amino acids, and one-letter, as well as three-letter codes, have been devised for them. A linear representation of a protein molecule can be a string of characters (also known as primary structure or primary sequence), each taking value over an alphabet of 20 letters. For analyzing structures, we need 3D information.

Tertiary (3D) structures of proteins are three-dimensional objects; they have shape and occupy volume in space as they are composed of atoms occupying positions. The atoms are not free-floating and connect to each other with links/bonds. Hence the obvious key question one would have to answer first is how do we represent these three-dimensional objects? What do we encode that will allow us to recognize any inherent organization?

3.2 Representation

3.2.1 Cartesian Coordinates

Generally, tertiary structures of protein molecules are represented as ordered sequences of the 3D coordinates of amino acids. If one considers a molecule of N atoms, then a naive representation of a structure of the molecule under consideration would be a point of form $x_1, y_1, z_1, \ldots, x_N, y_N, z_N$ in a space of $3N$ dimensions. The protein data bank (PDB) [6] stores structural information (coordinate file(s)) as the list of atoms in each protein molecule along with the spatial information (3D position in space) needed to reconstruct the particular structure. Besides the all-atom setting (where all types of atoms are considered), one can focus on a specific type of atom or a particular group of atoms. For instance, a protein structure can also be represented by the coordinates of the alpha-carbon (C_α/CA) atoms of its residues, discarding other atoms from the representation. This is often done to reduce dimensionality. Another way is to consider the backbone atoms: C_α, C, N, and O (backbone-atom setting). This representation is suitable for employing classic distance metrics (e.g., Euclidean distance) for comparing two structures. However, it is extremely high-dimensional as even a small protein can have hundreds of atoms. This presents what is also known as the curse of dimensionality. Finding an informative distance metric in high dimensional space remains an open problem.

3.2.2 Dihedral Angles

Instead of Cartesian coordinates, one can use the backbone dihedral/torsion angles (ϕ and ψ angles per amino-acid) as features. A dihedral angle is the angle between two planes; the plane formed by the atoms $i-2, i-1, i$ and the plane formed by the atoms $i-1, i, i+1$ where $i-2, i-1, i, i+1$ are four sequentially bonded atoms (Figure 3.1). The backbone of a protein (which links the backbone atoms) has three different torsion angles- phi (ϕ): rotation around N–C_α bond in an amino acid, psi (ψ): rotation around C_α–C bond in an amino acid, omega (ω): rotation around C–N bond linking two consecutive amino acids [63]. The

key idea here is that the changes in structures can be considered as the result of rotations around bonds that connect atoms. It has been observed that the comparison of structures (at room temperature) reveals changes in angles are due to some specific ones (dihedral angles). This representation ensures dimensionality reduction by a factor of 7 over the Cartesian coordinates. For this representation, the most intuitive distance function would be of L1-norm. However, it is necessary to go beyond the L1-norm to design more meaningful distance functions. Because, all angles are not equally important. If we interpret angles as rotations, changes in angles at the beginning of the chain of atoms cause larger changes (swept volume in 3D) than changes in angles at the end of the chain.

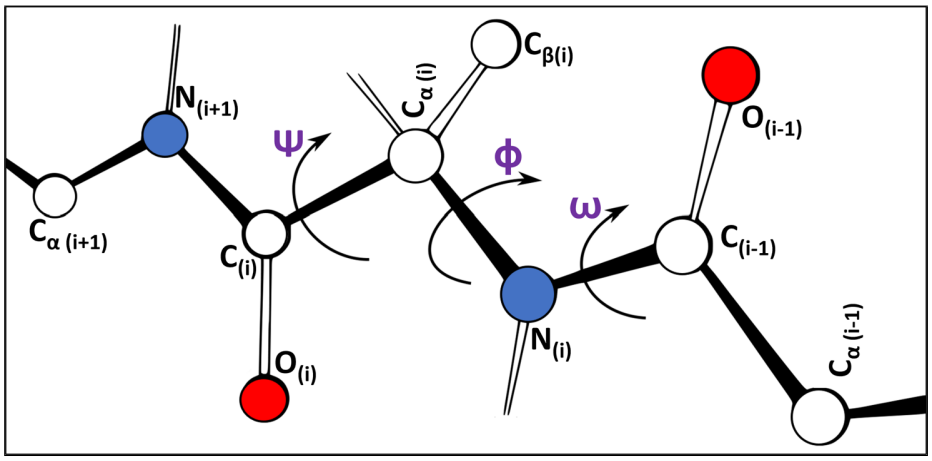


Figure 3.1: Backbone Dihedral Angles

3.2.3 Coarse-grained Representation

Ultrafast Shape Recognition (USR) metrics [64] that contribute to characterizing the three-dimensional shapes of ligands, can be utilized to featurize tertiary structures. The metrics are based on the moments of distance distribution of atoms and provide ways to compare

the molecular shapes. From a tertiary structure, USR metrics extract four reference points to characterize the distribution of distances of all atoms. These four points (Fig. 3.2) are: the molecular centroid (`ctd`), the closest atom to the molecular centroid (`cst`), the farthest atom from the molecular centroid (`fct`), and the farthest atom from `fct` (`ftf`). To capture the geometry and shape of a molecular structure, the moments of these discrete distributions are captured. The resulting distributions can be encapsulated by three measures: mean, variance, and skewness. As a result, each structure becomes a collection of 12 features [65]. The key observation for this type of representation is that changes in atomic positions or angles ultimately result in changes to the shape of the structure. Hence, it is possible to get away with a coarse representation of shape. USR metrics summarize the distance distribution of atoms from each of the four reference points via mean, variance, and skewness [64]. This encoding mechanism ensures dimensionality reduction and makes the data more suitable for applying clustering algorithms. However, this representation is too coarse to capture subtle structural changes.

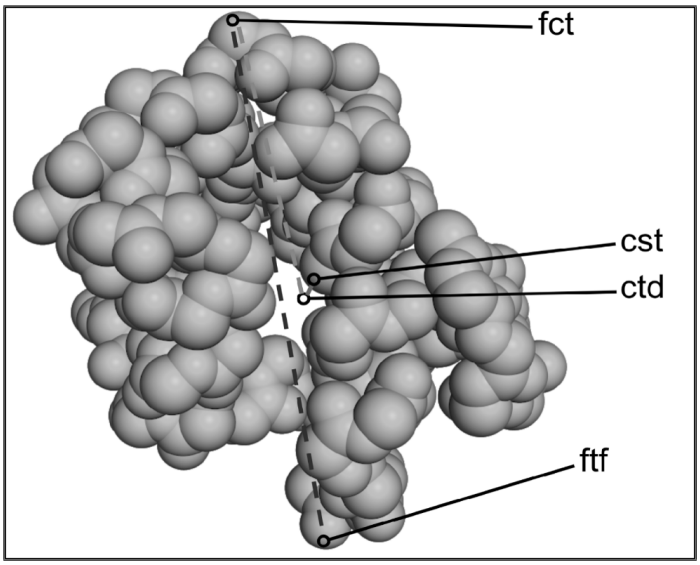


Figure 3.2: Reference Points for USR Metrics [65]

Furthermore, dimensionality reduction techniques are employed either implicitly or explicitly with these representations. For instance:- in [66], principal component analysis (PCA) is used to sample protein structure coordinates while considering the correlation among the atomic coordinates in the low-energy regions. To analyze molecular dynamics simulations, PCA is applied on dihedral angles [67] resulting in a one-to-one representation of the original angle distribution. Isometric feature mapping (ISOMAP) is used in [67] for the analysis of protein trajectories. Besides, time-lagged independent component analysis (TICA) is applied for molecular dynamics data that discovers coordinates of maximal auto-correlation at a specific lag time, whereas PCA focuses on the coordinates of maximal variance.

3.3 Domain-Specific Proximity/Distance Measure

Proximity measures or distance functions (for comparing protein tertiary structures) are primarily focused on Cartesian coordinate-based representation. To compare the tertiary structures of proteins, a number of similarity/dissimilarity measures are available [68]:

Root mean square deviation (RMSD)

The RMSD [69] between pairs of equivalent atoms is widely used to capture the degree of dissimilarity between two optimally superimposed tertiary structures of a protein. RMSD is computed by,

$$RMSD = \sqrt{\frac{1}{N}\sum_{1}^{N} \mid S_1^i - S_2^i \mid^2} \tag{3.1}$$

Here, N is the number of atoms, S_1^i and S_2^i represent the coordinate vectors for i-th atom of the structure1 and structure2 respectively (after optimal superimposition). One can consider a number of settings to compute RMSD such as: only over C_α atoms or backbone atoms or over all atomic coordinates of the structures. It is to be noted that the significance

of the RMSD value is dependent on the size of the structure e.g., a particular RMSD value between two large structures representing the degree of similarity might indicate otherwise for two smaller structures. However, RMSD is very intuitive and can be considered as a variant of euclidean distance.

Global Distance Test-Total Score (GDT-TS)

In the context of protein tertiary structures, GDT-TS determines the similarity between two structures with their corresponding superimposed residues.

$$GDT - TS(S_1, S_2) = \frac{\rho_1 + \rho_2 + \rho_4 + \rho_8}{4} \tag{3.2}$$

Here, ρ_t denotes the percentage of residues from structure S_1 to be superimposed with the corresponding residues from structure S_2 having chosen distance threshold (in terms of RMSD), t ($t \in \{1, 2, 4, 8\}$ $\mathring{A}$). GDT-TS value ranges from 0 to 1. The larger score indicates better similarity. In case of comparison between the two structures of the same protein, GDT-TS is more accurate than RMSD. However, GDT-TS score is sensitive to the lengths of the structures which can be tackled by normalization (scaling with the lengths of the structures under comparison). Moreover, a variant of GDT-TS, GDT-HA uses the distance threshold, $t \in \{0.5, 1, 2, 4\}$ $\mathring{A}$.

Template-Modeling (TM) Score

TM-Score determines the global structural similarity of a structure with respect to the reference structure in terms of the distances of each pair of residues.

$$TM - Score = max\left[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{1 + (\frac{d_i}{d_0})^2}\right] \tag{3.3}$$

Here, N denotes the number of residues, d_i represents the distance of the i-th pair of residues after alignment, and $d_0 = 1.24\sqrt[3]{N-15} - 1.8$ (Å). The range of TM-Score values is $(0, 1]$, with a higher value indicating better similarity. Nevertheless, GDT-TS and the TM-Score incur the same computational cost.

MaxSub Score

MaxSub aims at identifying the maximum superimposable subset of residues of a structure over the experimental structure and provides a single normalized score that exhibits the quality of the structure [70]. This score is sequence-dependent. MaxSub score ranges from 0 to 1, with a higher value indicating better similarity. A minor limitation of this score is that it does not take into account the fragmentation of the structures.

3.4 Internal Structure Energy: Potential Energy Functions

The energy function approximates the energy of a tertiary structure. The Rosetta energy/scoring functions offer a variety of alternatives to do so for a structure that consider different energetic terms to calculate the energy scores. We use Rosetta REF15 energy function [71]. The energy terms and their description for this energy function are provided in Table 3.1.

3.5 Interaction Energy

The performance of the computational methods that characterize protein–ligand interactions from tertiary structures relying on the laws of physics and chemistry is quantified by their capability to qualitatively describe protein–ligand interaction as well as by their ability to quantify the strength of interaction. In fact, the strength of the interaction can be determined by the free energy of binding and it is possible to measure this quantity experimentally. Computational methods strive to calculate the free energy of binding from tertiary structures and assess their performance by comparing with experimentally observed

Table 3.1: Energy Terms in REF15 Energy Function

Term	**Description**
`fa_atr`	Attractive energy between two atoms on different residues separated by distance d
`fa_rep`	Repulsive energy between two atoms on different residues separated by distance d
`fa_sol`	Gaussian exclusion implicit solvation energy between atoms in different residues
`fa_intra_rep`	Repulsive energy between two atoms on the same residues separated by distance d
`fa_intra_sol`	Gaussian exclusion implicit solvation energy between atoms in the same residue
`lk_ball_wtd`	Orientation-dependent solvation of polar atoms assuming ideal water geometry
`fa_elec`	Energy of interaction between two non-bonded charged atoms separated by distance d
`pro_close`	Penalty for an open proline ring and proline ω bonding energy
`hbond_sr_bb`	Energy of long-range hydrogen bonds
`hbond_lr_bb`	Energy of short-range hydrogen bonds
`hbond_bb_sc`	Energy of backbone-side-chain hydrogen bonds
`hbond_sc`	Energy of side-chain-side-chain hydrogen bonds
`dslf_fa13`	Energy of disulfide bridges
`omega`	Backbone-dependent penalty for cis ω dihedrals that deviate from 0° and trans ω dihedrals that deviate from 180°
`fa_dun`	Probability that a chosen rotamer is functionally relevant-like given backbone ϕ, ψ angles
`p_aa_pp`	Probability of amino acid identity given backbone ϕ, ψ angles
`yhh_planarity`	Sinusoidal penalty for non-planar tyrosine χ_3 dihedral angle
`ref`	Reference energies for amino acid types
`rama_prepro`	Probability of backbone ϕ, ψ angles given the amino acid type

free energies of binding [72]. Theoretically, the free energy of binding is determined by evaluating the properties of individual structures of the protein, ligand, the complex, or of their corresponding ensembles. In fact, binding free energy is a state function and is independent of the path taken from the protein or ligand to the protein-ligand complex [72]. Generally, the free energy of binding is decomposed into a number of additive energy components [73].

$$\Delta G_{bind} = \Delta G_{int} + \Delta G_{solv} + \Delta G_{motion} + \Delta G_{conf} \tag{3.4}$$

where ΔG_{int} denotes the free energy due to the interaction of the protein and ligand that form the protein-ligand complex, ΔG_{solv} is the free energy of solvation, ΔG_{motion} represents the free energy change associated with the changes in the motion of the protein, ligand, and the protein-ligand complex, ΔG_{conf} is the free energy due to structural changes during the formation of the complex [72]. It is worth mentioning that ΔG_{int} is dominated by enthalpic contributions from steric and electrostatic interactions upon complex formation. The steric interactions are usually captured by a pairwise Lennard-Jones (LJ) potential and the electrostatic interaction energy is usually computed via Coulomb's law using atom-centered point charges [74]. The impact of the protein environment is mimicked by scaling the Columbic interaction with a distance-dependent dielectric constant [75]. Both steric and electrostatic interactions are computed for non-bonded atoms in molecular mechanics force fields such as AMBER [15], and CHARMM [18].

3.6 Energy Landscape

The energy landscape is a fitness landscape that consists of a set of points X, a neighborhood $\mathcal{N}(X)$ of X, a distance metric on X, and a fitness function $f : X \to \mathbb{R}_{\geq 0}$ that assigns a fitness to each point in X. Neighbors are assigned to the points via a neighborhood function $\mathcal{N} : X \to \mathcal{P}(X)$. In the context of protein structure space, the points $x \in X$ represent structures and the fitness function often delegates an energy function. Effectively, the energy

landscape of a protein describes its internal or potential energy as a function of the energy-evaluated points (representing structures) $x \in X$ that constitutes the landscape. A protein energy landscape is complex. The multi-modal and high-dimensional nature of a protein energy landscape is contributed by an ensemble of structural states such as basins/wells and their separating barriers near or far from the functionally-relevant state [76]. In molecular energy landscapes, a basin corresponds to a long-lived, thermodynamically stable or semi-stable state [10]. The concept of a basin is related to a local minimum (also known as focal minimum). A focal minimum in a landscape is covered by a basin of attraction, that is the collection of points on the landscape from which the steepest ascent/descent converges to that focal optimum [77]. Barriers that comprise collections of local maxima along the path between basins isolate basins and modulate transitions of a system between different structural states corresponding to the basins in the landscape [12].

Chapter 4: Graph-embedding Methods to Organize Protein Structure Space

In this chapter, we present methods to leverage a graph-based representation of a computationally probed protein structure space in terms of the nearest-neighbor graph (nngraph), which is used to embed computed structures of a protein molecule of interest. Then, we organize the structures into groups via community detection algorithms initially devised to detect communities of users in social networks. After that, we evaluate the obtained groups and then demonstrate how various ranking-based techniques perform in automatically selecting groups that are more likely to contain functionally-relevant structures, employing experimentally-available structures as the ground truth. The work described in this chapter has been disseminated in [39].

4.1 A Graph-based Embedding of Structures

We employ a nearest-neighbor graph to represent the structure space probed via computation. The graph encodes the proximity of structures in this space. Let us denote the nngraph where the set of structures are embedded as $G = (V, E)$. The structures populate the vertex set V. A local neighborhood composition is inferred for each structure to populate the edge set E. This is based on proximity, measuring the distance between two structures via root-mean-squared-deviation (RMSD). First, each structure is superimposed over a structure selected as a reference (we arbitrarily choose the first structure as reference). The superimposition minimizes the differences due to rigid-body motions: whole-body rotation, and whole-body translation. After this superimposition, the RMSD is then measured between every pair of structures. It is to be noted that superimposing all structures to a reference structure and then performing pairwise RMSD computations

saves computation time. In contrast, seeking an optimal superimposition for each pair of structures would result in quadratic (rather than linear) running time. Once such distances are available for every pair of structures, the neighbors of each vertex $u \in V$ are other vertices $v \in V$ such that $dist(u, v) \leq \epsilon$, where ϵ is a user-defined parameter that controls the radius of the neighborhood. A vertex is connected via an edge to each of its neighbors determined in this manner. We note that proximity query data structures (such as kd-tree) allow efficiently extracting the nearest neighbors of a vertex. It is worth noting that the value of ϵ is an important consideration. A small value may result in a disconnected graph. This can be remedied by initializing ϵ to some initial value ϵ_0 and then increasing it by $\delta\epsilon$ over a maximum number of n_ϵ iterations while at the same time controlling the density of the nngraph via a parameter h. This parameter specifies the maximum number of neighbors allowed per vertex. In this way, only vertices with no more than h neighbors gain neighbors after each iterative increment of ϵ, with h controlling the density of the graph. It is worth noting that the nngraph is undirected. It is possible to convert it to a directed one by additionally including the role of energy in the embedding. An edge can be directed from a vertex corresponding to a structure with higher energy to a neighboring vertex corresponding to a structure with lower energy.

4.2 Organize Structures into Communities

After embedding the structures into an nngraph, we can employ community detection algorithms to identify communities as the groups of structures. We compare several state-of-the-art community detection algorithms, such as Girvan–Newman's Edge betweenness, which is based on hierarchical clustering; Leading Eigenvector (LE), which maximizes modularity over communities/clusters; Walktrap (WT), which implements an agglomerative approach; Label Propagation (LP), which seeks a consensus on a unique label for densely-connected vertices; Louvain (Lo), which is a heuristic-based method focusing on modularity optimization; InfoMap (IM), which is based on information flow analysis; and Greedy Modularity

Maximization (GMM), which implements hierarchical agglomeration-based clustering on a list of recommended metrics with the communities they yield, as well as on the quality of top-ranked selected communities.

4.2.1 Community Detection Methods

We provide here a high-level overview of the community detection algorithms that have been taken into consideration:

Edge betweenness (Girvan-Newman): This approach was introduced to sidestep the drawbacks of hierarchical clustering. It operates based on the intuition that edges linking the communities are anticipated to possess a high edge betweenness, which generalizes Freeman's betweenness centrality [78] from vertices to edges. To reveal the underlying community structure of the network, the Girvan-Newman method successively removes edges with high edge betweenness. Measuring edge betweenness takes $O(|E| \cdot |V|)$ time. Since this step has to be carried out repeatedly (for each edge), the entire approach runs in $O(|E|^2 \cdot |V)$ time.

Leading Eigenvector (LE): The prime objective of this method is modularity maximization (in terms of the eigen-spectrum of the modularity matrix) across possible subdivisions of a network [79]. With repeated divisions, the method discovers a leading eigenvector that partitions the graph into two subgroups; the goal of maximal improvement of modularity is achieved at every step. This process terminates when modification of modularity in the sub-network starts being negative. In fact, the method is associated with additional outcomes: a spectral measure of bipartite architecture in the network and a centrality measure to detect the vertices holding nuclear positions in communities. In general, the partitioning step takes $O(|V|(|E| + |V|))$ time.

Walktrap (WT): This method employs random walks to take into account the architectural resemblance between vertices (or groups of vertices). The underlying intuition is that vertices that are within the same community are supposed to have shorter distances for random walks [80]. The method administers an agglomerative approach that starts from $|V|$

communities (reduced to singleton clusters) and hierarchically merges two adjacent communities at each step. This is an effective approach to handling dense subgraphs of sparse graphs, which is most often the case for complex real-world networks. The method runs in time $O(|E||V|^2)$ and space $O(|V|^2)$ in the worst case.

Label Propagation (LP): This method is based on the intuition that each vertex in the network is supposed to follow the majority of its neighbors while joining a community [81]. The method aims for the robust use of the network infrastructure instead of a predefined objective function (to optimize) or *a-priori* information on the communities. Initially, a unique label is assigned to each vertex; that is, the method initializes $|V|$ singleton communities. In progressive steps, the adoption of a label comes into play for each vertex depending on the label possessed by the majority of its neighbors at that instant. This iterative process effectively performs the task of label propagation through the network and helps to form a consensus on a unique label for densely connected vertices. The process halts when each vertex and most of its neighbors have an identical label. The algorithm takes linear time in the number of edges ($O(|E|)$).

InfoMap (IM): This method identifies communities by using random walks along with information flow analysis [82]. The vertices and their connections are decomposed into modules to represent the network in such a way that maximizes the amount of information in the actual network. The method tries to assign codewords to vertices; the process is efficient in terms of the dynamics of the network. A signal is transmitted to a decoder (via a limited capacity channel) who tries to decode the message, as well as to form viable candidates for the actual network. The lower the number of candidates, the more information about the actual network has been transmitted. The method runs in $O(|E|)$ time.

Louvain (Lo): This heuristic-based method focuses on modularity optimization. The method consists of an iterative repetition of two stages. The first stage deals with the initial partition, where each vertex is assigned to a unique community (singleton communities). Modularity gain is measured by assigning a vertex to a neighborhood community so as to exclusively search for a way to maximize positive gain. The order in which vertices

are explored does not affect modularity but may increase computation time. The second stage commences with the construction of a new weighted network, whose vertices are the communities generated by the first phase. This process continues until maximum modularity is achieved [83].

Greedy Modularity Maximization (GMM): This is a hierarchical agglomeration method that makes use of a greedy optimization approach. The underlying assumption is that high modularity values are associated with good communities. Initially, each vertex itself forms a community. Then, the vertices of the two communities are combined together in a way that yields maximum modularity gain. This step is repeated $(|V| - 1)$ times. The process is represented as a hierarchical tree-like structure (a dendrogram), whose end-nodes represent the vertices of the actual network, and the internal vertices correspond to the connections; that is, the dendrogram shows a hierarchical decomposition (level-wise) of the network into communities. The method runs in $O(|E|\ d_d \log |V|)$ time [84], where d_d is the depth of the dendrogram representing the network's community architecture.

4.2.2 Metrics for Evaluating Community Detection Methods

A comprehensive list consisting of 15 community-recommended metrics has been considered to assess the community detection methods [85]. We note that the following metrics are scoring functions that perform mathematical formalization of the community-wise connectivity structure of a provided set of vertices and identify communities as high-scored sets. To summarize these metrics, let us consider a graph $G(V, E)$ with $n = |V|$ vertices and $m = |E|$ edges, and a community is defined as a set S of n_S vertices and m_S edges.

Fraction Over Median Degree (fomd): Let the degree of u for each vertex $u \in S$ be denoted by $d(u)$, and let d_m be the median across the degrees $d(u)$. Then, fomd is determined as the fraction of vertices in S with an internal degree greater than d_m; that is, $fomd(S) = \frac{|\{u:u \in S, |\{(u,v):v \in S\}| > d_m\}|}{n_S}$. The denser and more cohesive the communities, the higher the associated fomd scores.

Max odf (out degree fraction): Max odf evaluates the maximum ratio of edges of a

vertex in community S which point outward from S. That is, $odf_{max}(S) = \max_{u \in S} \frac{|\{(u,v) \in E : v \notin S\}|}{d(u)}$. According to Max odf, a community is characterized as a set of vertices that connect to more vertices within the set than to vertices outside of it. As a result, better communities are associated with lower Max odf scores.

Triangle(Triad) Participation Ratio (tpr): Let T_c denotes the number of vertices which form a triangle in S. The tpr metric measures the ratio of vertices belonging to a triangle and can be formulated as: $tpr(S) = \frac{|\{u : u \in S, \{(v,w) : v,w \in S, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \emptyset\}|}{n_S}$. Better community clustering yields higher tpr scores.

Internal Edge Density: For a set S, let us denote the maximum number of possible edges by $m_{Smax} = n_S(n_S - 1)/2$. The internal edge density is the ratio of the edges that are actually in S, denoted by m_S, over m_{Smax}; that is, $ied(S) = \frac{m_S}{n_S(n_S-1)/2}$. This metric represents the internal connectivity of a cluster (community) and a higher score indicates that there are more connections within the vertices of that community.

Average Internal Degree: This metric determines the average internal degree of the members of set S and can be formulated as: $aid(S) = \frac{2m_S}{n_S}$. The denser a community, the higher its average internal degree score.

Cut Ratio: Let C_S denotes the edges that are going outward from a set S. The cut ratio score measures the ratio of C_S over all possible edges and is defined as: $cr(S) = \frac{C_S}{n_S(n-n_S)}$. Better communities are associated with lower scores.

Expansion: This metric calculates the number of edges (for each vertex) going out of a set S and can be formulated as: $ex(S) = \frac{C_S}{n_S}$. Lower scores correspond to better communities.

Edges Inside: This metric measures the internal connectivity of a set S as $ei(S) = m_S$. Better communities are related with higher scores.

Conductance: This metric is based on the combination of internal and external connectivity and is measured as: $cnd(S) = \frac{C_S}{(2m_S+C_S)}$. Lower scores relate with well-separated communities.

Normalized Cut: This metric is defined as: $nc(S) = \frac{C_S}{(2m_S+C_S)} + \frac{C_S}{2(m-m_S)+C_S}$. The metric has the special property that concurrently meets the two following objectives: maximization of dissimilarity across communities and minimization of overall similarity (eschewing the unnatural bias for breaking up small sets). Lower values of normalized cut maintain balance between these two objectives.

Coverage: This metric measures the ratio of the number of intra-community edges to the number of edges in the graph and is defined as: $cvg(S) = \frac{\omega(C)}{\omega(G)}$. Here, $\omega(C) = \sum_{i=1}^{k} \omega(E(v_x, v_y)); v_x, v_y \in C_i$. Higher coverage values indicate that there are more connections within communities rather than edges linking various communities. In fact, the ideal scenario is that communities are completely separated from one another, which would correspond to a coverage of 1 (the maximum possible value).

Average odf This metric provides the average ratio of edges that point outward of S over vertices in S and is defined as: $odf_{avg}(S) = \frac{1}{n_S} \sum_{u \in S} \frac{|\{(u,v) \in E : v \notin S\}|}{d(u)}$. Lower values of average odf relate with better communities.

Modularity: This metric is based on the network model and determines the difference between the number of edges within S and the expected number of such edges in a random graph of with identical degree sequence, $E(m_S)$ [85]. Modularity can be defined as: $md(S) = \frac{1}{4}(m_S - E(m_S))$. Higher values of modularity correspond to denser connections within a community than anticipated at random.

Flake odf: This metric combines internal and external connectivity and determines the fraction of the number of vertices with fewer connections within the community than with the outside. Flake odf is defined as: $fodf(S) = \frac{|\{u : u \in S, |\{(u,v) \in E : v \in S\}| < d(u)/2\}|}{n_S}$. Better communities are associated with higher values.

Separability: This is a community-goodness metric [85] based on the intuition that good communities are well-separated (have relatively few edges from set S to the rest of the network). Separability finds the ratio between edges pointing in and outside of the set S and is defined as: $sp(S) = \frac{m_S}{C_S}$. Higher values indicate better communities.

We note that these metrics can be grouped into four major classes [85]: metrics based on internal connectivity (fraction over median degree, triangle participation ratio, internal edge density, average internal degree, edges inside), metrics based on external connectivity (cut ratio, expansion), metrics based on internal-external connectivity (conductance, normalized cut, max odf, average odf, flake odf), and metrics based on the network model (modularity).

4.3 Selection Strategies to Rank the Groups/Communities

We utilize group-level characteristics associated with the communities to rank them. They fall into three categories: size, energy, and hybrid characteristics. The size of a group is the number of structures/vertices in it. Energy can also be associated with a group. We note that the structures we consider here are generated from template-free methods, which pursue an optimization approach that seeks to minimize the inter-atomic energy in a structure via a selected energy function. Hence, each structure has an associated energy value. Given the energies of structures in a community, the energy of the same can be defined as the minimum energy over all the structures in it or the average over the energies of the structures in it.

Whether size or energy, a selection technique ranks (via sorting) the groups and selects the c top-rank communities, offering them as prediction for where functionally-relevant structures reside. It is worth mentioning that considering only energy would promote a significant number of false positives, as it is well known that protein energy functions are inherently inaccurate. Therefore, we consider both size and energy together in a second selection strategy (S+E); we consider the $l > c$ largest groups and then re-sort them from lowest to highest energy, selecting the top c of them for prediction. Hybrid characteristics consider both size and energy but additionally take into account the possibility that size and energy are possibly conflicting optimization criteria. Since solutions minimizing all conflicting objectives simultaneously are typically non-existent, Pareto-optimal solutions

are sought. A Pareto-optimal solution cannot be improved in one objective without sacrificing the quality of at least one other objective. That is, a solution Sol_1 *Pareto-dominates* another solution Sol_2, if the following two conditions are satisfied:

(1) For all optimization objectives i, $score_i(Sol_1) \geq score_i(Sol_2)$;

(2) For at least one optimization objective i, $score_i(Sol_1) > score_i(Sol_2)$.

Based on this concept of Pareto optimality, two additional quantities, Pareto Rank (PR) and Pareto Count (PC), can be associated with each group C. These two quantities employ the concept of dominance, summarized above. PR(C) is the number of communities that dominate C. PC(C) is the number of communities that C dominates. It is now straightforward to use these two new, hybrid characteristics, in the same ranking-based manner. It is worth noting that while ranking with PR, the groups are sorted by low to high PR values; in PR+PC, groups with the same PR value are additionally sorted from high to low PCs. Taken all together, we consider four selection strategies: Sel-S, Sel-S+E, Sel-PR, Sel-PR+PC.

4.4 Evaluation Dataset and Metric

Our evaluation focuses on ten target proteins of different folds and lengths (number of amino acids), listed in Table 4.1 where column 2 shows the PDB ID of an experimentally-available, functionally-relevant structure for each test case, and columns 3 and 4 show the fold (* indicates structures with a predominant β fold and a short helix) and the length (number of amino acids), respectively, whereas column 5 shows the size of the structure set Ω generated via the Rosetta *ab-initio* protocol [21], and column 6 shows the lowest lRMSD from the experimentally-known structure over the structure ensemble. The targets selected have experimentally known structures to aid the evaluation. The Protein Data Bank identifier (PDB ID) of the (crystallographic) experimentally known structure of each target is shown in Column 3 in Table 4.1. The targets listed in Table 4.1 are divided

into three categories (easy, medium, and hard) to indicate the quality of the Rosetta-generated structure ensembles. This categorization emerges from analysis in terms of the lowest distance (measured via least root-mean-squared distance–lRMSD) of all Rosetta-computed structures from the corresponding experimentally-available structure of a target. Specifically, if the lowest lRMSD (over all structures) `min_dist` ≤ 0.7, these are considered as the easy cases. For medium-difficulty targets, the range is 0.7 Å $<$ `min_dist` $<$ 2Å), and the `min_dist` > 2.0 for the hard cases. This distance is shown as min_dist in Column 6 in Table 4.1.

Table 4.1: Rosetta Generated Structure Ensemble Dataset of Proteins

	PDB ID	**Fold**	**Length**	$\|\Omega\|$	**min_dist (Å)**
Easy	1dtdb	$\alpha + \beta$	61	$57,839$	0.51
	1tig	$\alpha + \beta$	88	$52,099$	0.60
	1dtja	$\alpha + \beta$	74	$53,526$	0.68
Medium	1hz6a	$\alpha + \beta$	64	$57,474$	0.72
	1c8ca	β^*	64	$53,322$	1.08
	1bq9	β	53	$53,663$	1.30
	1sap	β	66	$51,209$	1.75
Hard	2ezk	α	93	$50,192$	2.56
	1aoy	α	78	$52,218$	3.26
	1isua	*coil*	62	$60,360$	5.53

4.4.1 Evaluation Metric

As stated earlier in Chapter 1, a limitation of experimental techniques is that they reveal a limited number of functionally-relevant structures for a protein; in the majority of cases, we often have only one such structure, indicated above in Table 4.1 with one PDB ID entry. Operating within this limitation, we expand the notion of ground-truth structures to include additional structures around one given experimental structure to at least account for small

(RMSD) fluctuations expected under physiological conditions (at room temperature). So, in our metric described below that allows us to evaluate the quality of a group, we refer to plurality of functionally-relevant structure(s) for a given protein.

Specifically, we leverage the purity metric p, which keeps track of the number of functionally-relevant structures relative to the size of a group of structures. The purity of a group C is,

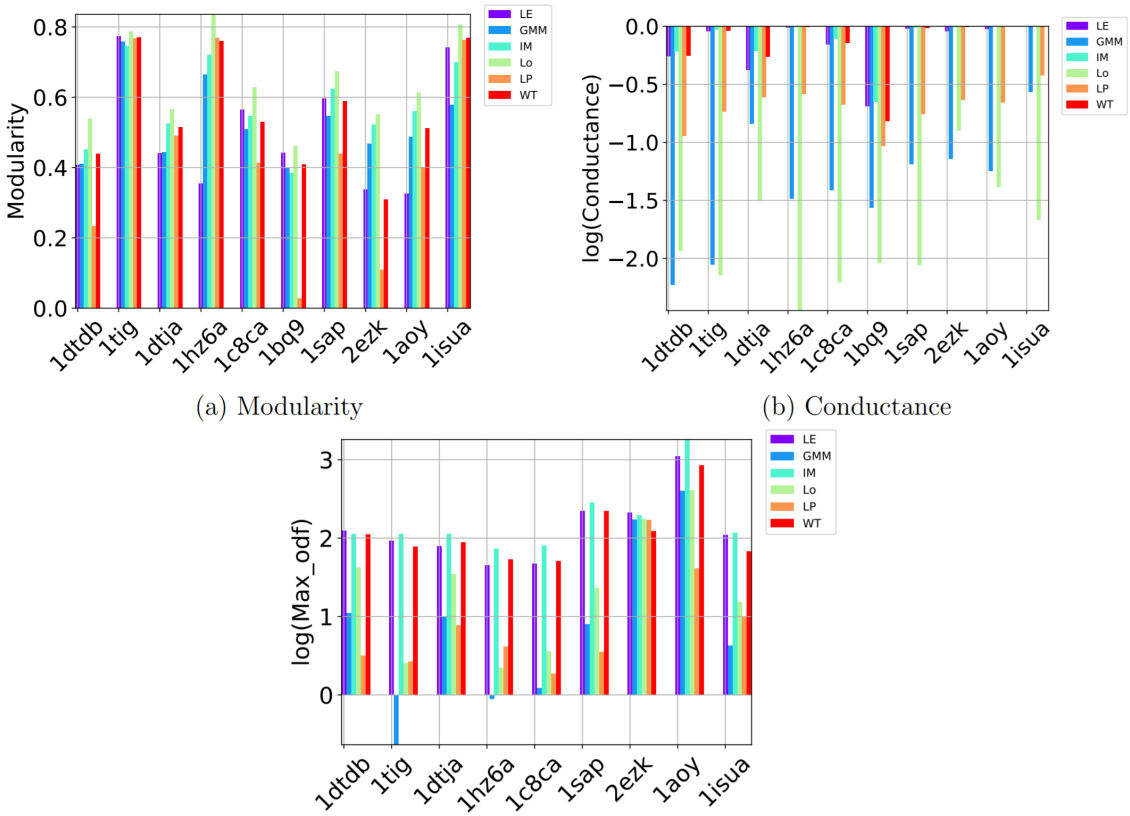
$$p_c = \frac{\text{number of functionally-relevant structures in C}}{|C|} \tag{4.1}$$

where $|C|$ denotes the size of group C.

Purity is related with the precision metric in machine learning. If we consider the functionally-relevant structures in a group as true-positives (TP) and all other structures as false-positives (FP), then purity is $\frac{TP}{TP+FP}$. Here, we are less concerned about the false negatives, as our objective is to maximize the possibility of selecting a functionally-relevant structure from a group uniformly at random, which is equivalent to maximizing true positives and minimizing false positives in a group. The purity metric p penalizes a group by the number of false positives present in that group. Therefore, a group populated with a large number of false positives will result in a low purity (p) regardless of the number of true positive population present in that group.

4.5 Evaluation Results

As mentioned earlier in Section 4.2, we have evaluated the community detection algorithms using a number of recommended metrics; Figure 4.1 shows the comparison along three of those metrics that evaluate communities over undirected nngraphs and demonstrates that Louvain and GMM yield better communities in comparison to the others as higher modularity represents denser connections in a community, lower values for conductance and max odf correspond to better separated communities.



(a) Modularity

(b) Conductance

(c) Max odf

Figure 4.1: Comparison of Community Detection Algorithms on Modularity (a), Conductance (b), and Maximum out degree fraction-Max odf (c)

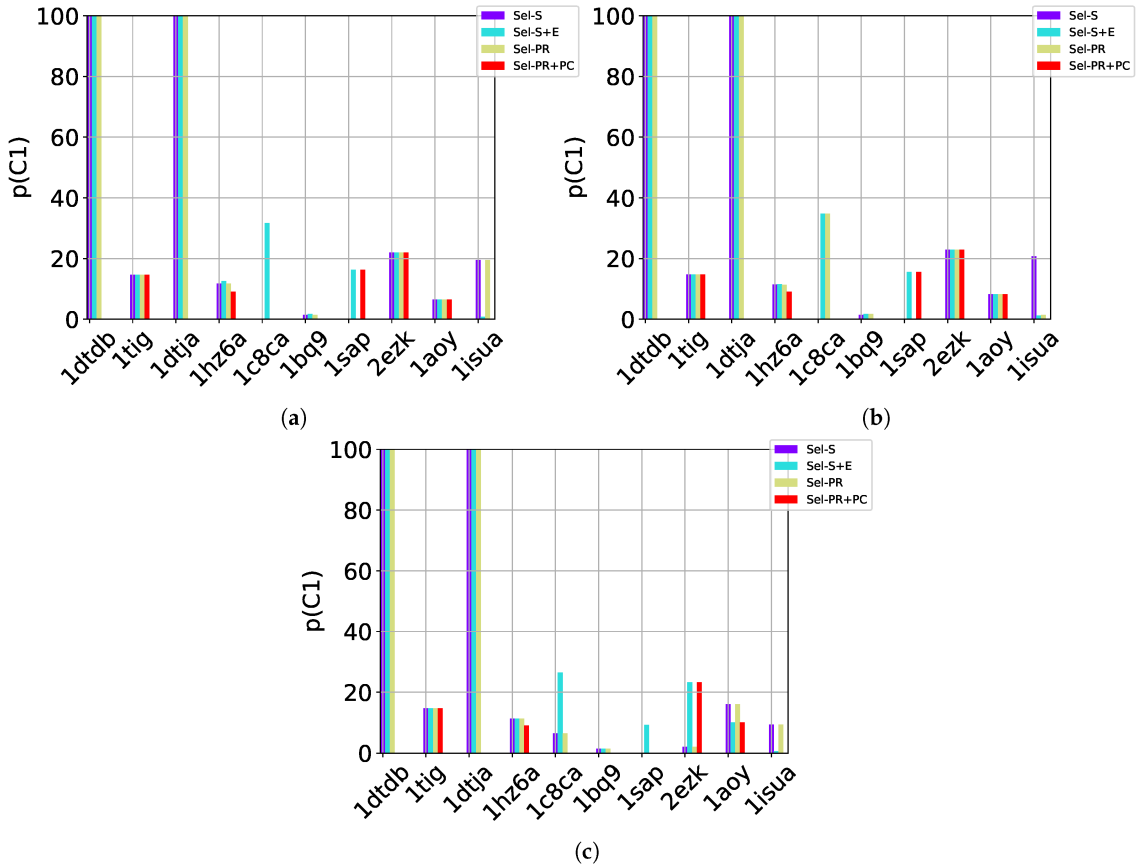


Figure 4.2: Comparison of the various selection strategies on the purity of the top community(C1) selected over communities detected with the (a) Louvain method on directed nngraph, (b) Louvain method on undirected nngraph, and (c) GMM method on undirected nngraph embeddings of the structures

Fig. 4.2 and 4.3 show the comparison of the selection strategies in terms of the purity of the topmost and the top 3 communities detected with the Louvain method on directed nngraph embeddings of structure data in (a), the Louvain method on undirected nngraph embeddings of structure data in (b), and the GMM method on undirected nngraph embeddings of the structure data in (c). These comparisons imply that, of the four selection strategies, Sel-S and Sel-S+E consistently yield good results. And taking all together, Sel-S+E is better than the others.

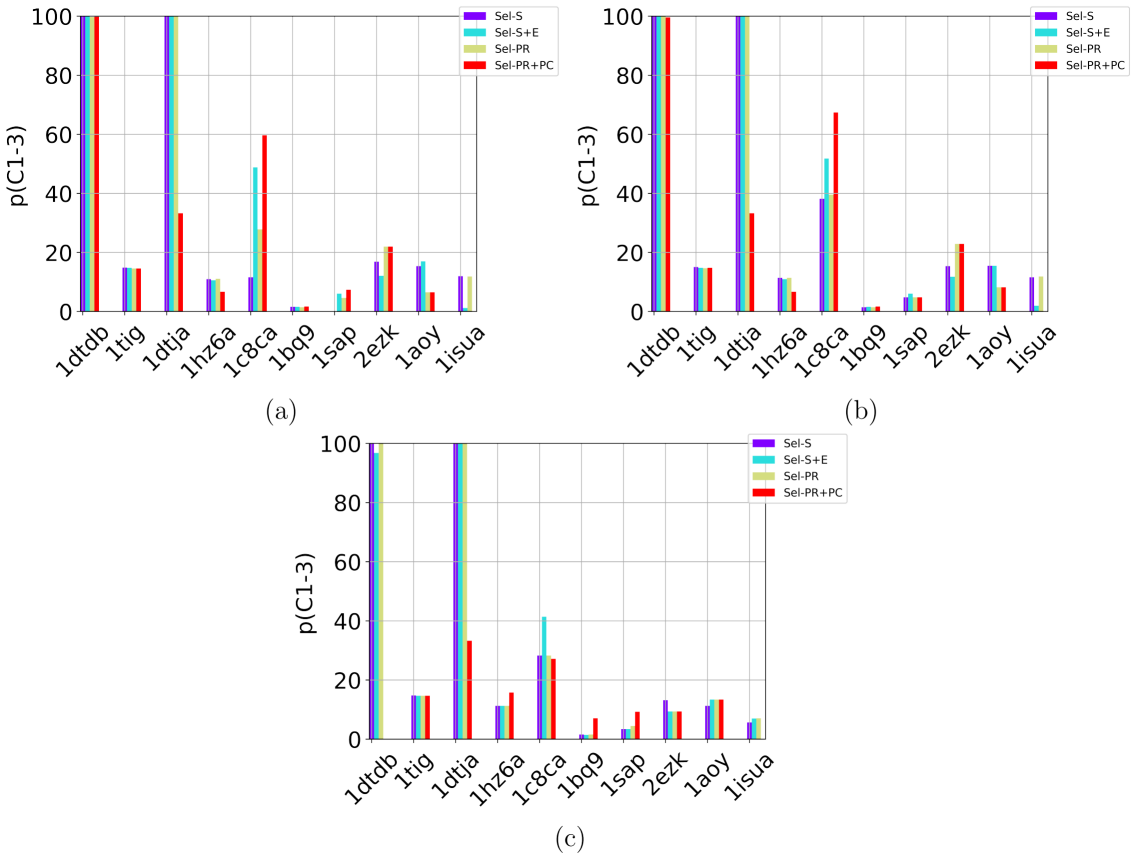


Figure 4.3: Comparison of the various selection strategies on the purity of the top three communities (C1-3), selected over communities detected with the (a) Louvain method on directed nngraph, (b) Louvain method on undirected nngraph, and (c) GMM method on undirected nngraph embeddings of the structures

In fact, we bolster the above comparison of selection strategies via two sets of analyses. First, we compare the four selection strategies based on the rank/position of the purest community in the sorted order they impose on detected communities. Table 4.2 reports the rank of the top-selected community (by each selection strategy) in a purity-based ordering (from high to low purity). The lowest rank over the selection strategies is highlighted in bold font. The results in Table 4.2 show that the lowest rank is obtained overall by Sel-S+E, which selects communities by size and energy. Moreover, on the medium- and hard-difficulty datasets, the Louvain on directed and undirected and GMM behave comparably, with GMM outperforming the two other methods on the easy datasets.

Table 4.2: Rank (by **Size, Size and Energy, Pareto rank, Pareto rank and Pareto count**) of the community with the highest purity among those identified by Louvain (Lo), Louvain$_{Directed}$ (Lo$_D$) and GMM.

	Rank by (Lo) **S, S+E, PR, PR+PC**	**Rank by (Lo$_D$)** **S, S+E, PR, PR+PC**	**Rank by (GMM)** **S, S+E, PR, PR+PC**
1dtdb	3, 4, **1**, 9	**1**, **1**, **1**, 3	**1**, **1**, **1**, 8
1tig	691, **396**, 7069, 7073	229, **112**, 2287, 2289	283, **44**, 962, 963
1dtja	71, **64**, 26735, 26736	**1**, 7, **1**, 12	**1**, 3, **1**, 9
1hz6a	647, **639**, 10160, 10166	280, **49**, 673, 670	337, **70**, 748, 740
1c8ca	818, **572**, 9700, 9736	42, **31**, 540, 542	15, **1**, 4, 2
1bq9	1230, **267**, 4816, 4836	1223, **268**, 4810, 4827	1271, **269**, 4826, 4853
1sap	3301, **137**, 538, 541	3298, **137**, 538, 551	3369, **142**, 566, 566
2ezk	6, **5**, 13, 12	**3**, 9, 14, 16	3, **1**, 2, **1**
1aoy	3, **2**, 12, 11	**3**, **3**, 14, 13	**1**, 3, **1**, 3
1isua	135, **117**, 1519, 1527	136, **117**, 1520, 1525	194, **193**, 1236, 1241

Second, we conduct 1-sided and 2-sided statistical significance analysis via Fisher's [86] and Barnard's [87] exact tests on 2×2 contingency matrices. The analysis compares Sel-S+E to the other three selection strategies on the rank of the top-selected community in a purity-based ordering (from high to low purity). Over each of the 10 structure datasets

corresponding to 10 target proteins, the rank of the top-community selected over those identified by Louvain on directed and undirected nngraphs and GMM on undirected nngraphs. Fisher's exact test is conditional and widely adopted for statistical significance. Barnard's test is unconditional and generally considered more powerful than Fisher's test on 2×2 contingency matrices. We use 2-sided tests to determine which algorithms do not have similar performance and 1-sided tests to determine if Sel-S+E performs significantly better than the other selection strategies.

Table 4.3: Comparison of **Sel-S+E** to other selection strategies on best rank via **1-sided** Fisher's and Barnard's tests. Top panel evaluates the null hypothesis that **Sel-S+E** does not provide the best rank (based on reported p-values), considering each of the other three selection strategies in turn. Similarly, the lower panel evaluates the null hypothesis that **Sel-S+E** does not provide a better rank with respect to another particular selection strategy, considering each in turn.

Best Rank			
Test	**Sel–S**	**Sel–PR**	**Sel–PR+PC**
Fisher's	6.621×10^{-7}	1.626×10^{-7}	9.388×10^{-12}
Barnard's	2.314×10^{-7}	6.33×10^{-8}	2.128×10^{-12}
Better Rank			
Test	**Sel–S**	**Sel–PR**	**Sel–PR+PC**
Fisher's	0.0001154	7.744×10^{-7}	4.194×10^{-15}
Barnard's	6.738×10^{-5}	3.811×10^{-7}	8.075×10^{-16}

The top panel in Table 4.3 evaluates the null hypothesis that **Sel-S+E** does not provide the best rank, considering each of the other three selection strategies in turn. The bottom panel evaluates the null hypothesis that Sel-S+E does not provide a better rank with respect to another particular selection strategy, considering each of the other three in turn. The results in Table 4.3 show that the null hypothesis is rejected in both cases.

Table 4.4 shows a similar comparison for a 2-sided test. The results in Table 4.4 show

that the null hypothesis is rejected in both cases. Taken together, the analysis confirms that Sel-S+E is the top performing selection strategy with regards to the rank of the the top-selected community in a purity-based order of detected communities.

Table 4.4: Comparison of **Sel-S+E** to other selection strategies on best rank via **2-sided** Fisher's and Barnard's tests. The tests evaluate the null hypothesis (based on reported p-values) that **Sel-S+E** (or, **Size+Energy**) provides similar ranking in comparison to other selection strategies.

Best Rank			
Test	**Sel–S**	**Sel–PR**	**Sel–PR+PC**
Fisher's	1.324×10^{-6}	3.252×10^{-7}	1.878×10^{-11}
Barnard's	4.629×10^{-7}	1.266×10^{-7}	4.255×10^{-12}
Better Rank			
Test	**Sel–S**	**Sel–PR**	**Sel–PR+PC**
Fisher's	0.000231	1.549×10^{-6}	8.388×10^{-15}
Barnard's	0.0001348	7.621×10^{-7}	1.615×10^{-15}

4.6 Summary

The work summarized in this chapter shows the utility of embedding computationally-sampled tertiary structures of a protein molecule in a graph and leveraging the graph embedding to elucidate the organization of the structure space. Prior work has shown that other clustering algorithms not based on graphs perform poorly in comparison [88]. However, while we show here that embedding structures in a graph is informative, it comes with high memory demands due to the storage of the vertex and edge lists. This motivates us to explore alternative frameworks and do away with explicit storing of the graph data structure. In later chapters (Chapter 6-8) we directly leverage the adjacency matrix via matrix (and tensor) decomposition-based methods that also prove more powerful for a

variety of application settings. Before we proceed to relate our work in this direction, we take a short detour and show how the organization of protein structure space is powerful and essential in obtaining discrete summarization of protein dynamics, a key application of the methodological work described here.

Chapter 5: From Organization of Structure Space to Markov State Models of Dynamics

In this Chapter, we demonstrate how the organization of the structure space of a protein molecule can be leveraged to build discrete models of protein structure dynamics. We leverage work presented in the previous chapter that embeds structures in a graph to find the inherent organization. We compare in this setting the community-detection formulation presented in the previous chapter and the top algorithms shown, Lo and GMM, with work in [88], which additionally utilizes energies of structures to group structures into *basins*, a concept we relate in detail later in this chapter. We make the connection between structure groups or structure basins and structural states and then build Markov State models (MSMs) over the states to summarize protein dynamics. We compare various methodological settings, effectively conducting an ablation study, and demonstrate the utility of the work on a highly-flexible peptide, Met-Enkephalin. The work described in this chapter has been disseminated in [89, 90].

5.1 Application Setting

The setting we investigate here is the following. We are provided structures accessed by a molecule in a physics-based simulation via Molecular Dynamics (MD) platforms. The structures are organized into trajectories. Each trajectory is a list of structures accessed consecutively during the simulation, in time steps of magnitude δt. So, for instance, a trajectory T_j is a list $\{S_{t_0}, S_{t_1}, \ldots, S_{t_k}\}$, where $t_{i+1} - t_i = \delta t$, and S_{t_i} is a structure accessed by a molecule in simulation at time t_i. An MD simulation provides a local view of the structure space of a protein, which can be considered a biased random walk, with the potential energy providing the bias towards lower-energy structures. However, since the

energy landscape is vast and rich in local minima, an MD simulation often converges to a local minimum near the structure initializing the simulation. That is the reason why many MD simulations are typically conducted so that one obtains a list of trajectories T_j.

Then, the problem is how to integrate these trajectories. Many of them may be intersecting paths in the structure-energy space. Many may contain structures that populate the same or nearby local minima in the space. Therefore, our goal is to integrate these trajectories, so that structures are grouped into structural states, and then to utilize the temporal information in the trajectories to build an MSM over the states, which gives us a high-level but quantitative summarization and view of the structure-energy space and tells us, for instance, what the major states are, what are the transitions between states, and whether there are "absorbing" states that would indicate important, functionally-relevant regions of the structure-energy space.

Computational platforms that build MSMs of dynamics are now becoming increasingly popular tools in computational biology. There are two main ones, MSMBuilder [91] and EMMA [92]; the Python implementation of the latter is known as PyEMMA. These platforms do not spend much thought on how to organize structures first into groups and rely on simple clustering algorithms, such as k-means. They also allow some reduction of the structures via dimensionality reduction algorithms, such as Principal Component Analysis (PCA) or Independent Component Analysis (ICA) [93]. These platforms, however, are important in providing us with the functionality we need to construct MSMs, as well as to interrogate the quality of a constructed MSM with rigorous statistical analysis.

In this Chapter, our hypothesis is that more careful thought into how to organize structures first into groups or states will lead to better-quality MSMs. We utilize our prior work on graph-clustering and additionally consider energy in order to better capture the organization of structure-energy space. Once the states are obtained in this manner (as we will also show in more detail later in this Chapter), we then utilize PyEMMA to build an MSM over the identified states. We build a full computational pipeline that converts given MD trajectories into an MSM of the fragmented dynamics probed in simulation.

5.2 Organizing Structures in MD Trajectories

We first provide an overview of the computational pipeline that constructs an MSM from various MD trajectories. We then describe the approaches that organize the structures accessed in MD simulations into structural states. Finally, we summarize the statistical techniques we employ for evaluation of the various MSMs one can obtain.

5.2.1 From MD Trajectories to a Markov State Model (MSM)

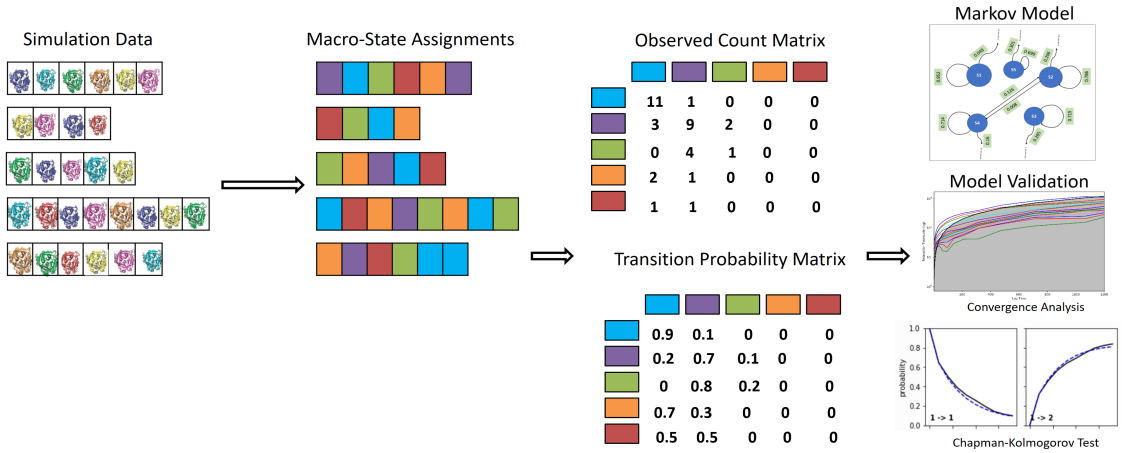


Figure 5.1: Schematic of the computational pipeline that converts the information available in MD trajectories probing the structural dynamics of a molecular system of interest into an MSM of the system's dynamics.

Fig. 5.1 provides a schematic overview of the computational pipeline that takes MD trajectories and returns a transition probability matrix. As described above, the input to the pipeline is a list of MD trajectories. Each MD trajectory is a series of structures accessed consecutively in an MD simulation. Generally, the structures are described in terms of the Cartesian coordinates of the atoms of the molecular system under investigation. The time

interval/step between two successive structures is also available. Generally, the time step is a user-defined parameter in an MD simulation. Its range can vary from 1 femtosecond (fs) in a detailed MD simulation to several femtoseconds in coarse-grained, expedited simulations.

It is not necessary (and sometimes not feasible) to analyze all the structures in an MD trajectory. Most of the MSM construction software, such as MSMBuilder [91] and EMMA [92] allow the user to select a lag time. This can be a multiple of the original time step/gap between two successive structures in an MD trajectory. The selection of lag time can be viewed as a data reduction strategy. It is generally assumed that structures within the same state inter-convert faster than the chosen lag time, but it is important to verify this assumption by examining the properties of the resulting MSMs for varying lag times.

After selecting a suitable lag time (which is a user-defined parameter and can range from the time interval between two successive structures in MD simulation to several multiples of it), the construction of the MSM proceeds as follows. Before organizing structures into states, structures are "prepared" via a two-step process.

First, features of interest (such as coordinates of all the atoms or only a specific group of atoms) are extracted from structures that are subjected to reduction via dimensionality reduction techniques, and it is the reduced representations of the structures that are fed to clustering algorithms for identification of structural states. In fact, studies [94] suggest that the time-lagged Independent Component Analysis (TICA) [93] is preferable over PCA (or other dimensionality reduction techniques) for MD trajectory data. Moreover, the fundamental distinction between the PCA and TICA is that PCA captures coordinates of maximal variance, while TICA captures coordinates of maximal auto-correlation for a given lag time. The reduced structures are then subjected to a clustering algorithm in order to group structures into clusters, which are referred to as structural states. This process is also known as state-space discretization. The baseline algorithms provided in EMMA are k-means, uniform time clustering, regular space clustering; but one can apply a different clustering algorithm. This is where we evaluate two alternative approaches: one with community detection algorithms and another with the identification of energy landscape

basins.

5.2.2 State Identification via Clustering over a Graph Embedding of Structures

We employ the strategy of embedding the structures obtained from various MD simulations in a nearest-neighbor graph (nngraph). The nngraph encodes the proximity among the structures in the MD-probed structure space. Specifically, consider a set of structures to be embedded in an nngraph $G = (V, E)$, where the vertex set V contains the structures, and the edge set E is generated by inferring a local neighborhood over each vertex. A vertex is connected via edges (undirected) to its nearest neighbors, which are identified by pre-specifying a distance threshold. We construct the undirected nngraph as described in Section 4.1.

It is worth noting that we do not make use of any sophisticated features or any dimensionality reduction and instead compute the distance between two structures in the original structure space probed by the various MD trajectories in terms of RMSD. The application of RMSD requires the structures under comparison to being first superimposed so as to remove differences due to rigid-body motions (translations and rotations in three dimensions). This is a computationally expensive step. Instead, we first align all the structures to a reference structure (arbitrarily selected to be the first in an MD trajectory). This pre-alignment to a reference has been employed in time-aware RMSD-based analysis of molecular structures [27, 88]. Moreover, the identification of nearest neighbors does not have to rely on a brute-force approach to performing all possible comparisons. Instead, we make use of a proximity query data structure (k-d tree) that organizes structures so as to efficiently answer proximity queries.

Clustering can be carried out over the nngraph. Specifically, we leverage community detection algorithms originally introduced for applications that encode relations among entities or individuals in social networks. Such algorithms are effective in aggregating structures based on their organization in the nngraph, identifying cohesive groupings among

the structures/vertices. As suggested by our work summarized in Chapter 4, the top two community detection algorithms are Louvain (Lo) and Greedy Modularity Maximization (GMM); we consider these two algorithms to cluster structures into states for the purpose of constructing the MSM models.

5.2.3 State Identification via Detection of Energy Landscape Basins

An alternative strategy is to utilize (rather than ignoring) the energies of structures obtained in MD simulations. This strategy considers that the MD simulations have probed an underlying energy landscape, which adds the energy as an additional dimension to the structure space. In this landscape, a point is a structure-energy pair, and an MD simulation has probed this landscape one point at a time; effectively, an MD trajectory leaves footprints that are nearby points in the landscape. The landscape itself proves an organization of the structure space, as it groups together structures that are geometrically and energetically similar. The landscape contains information on how structures with similar energies inter-convert into one another, thus providing an opportunity for quantitative understanding of the underlying dynamics of a molecule of interest [11]. Specifically, a thermodynamically-stable (or semi-stable) state does not directly rely on structural similarity but instead corresponds to basins/wells in the energy landscape [9].

This understanding of the rich information in an energy landscape inspires us to pursue statistical spatial analytics capable of identifying basins in the landscape and so organizing structures into basins. Identification of basins again relies on first embedding the structures in a nngraph (as described in Section 4.1), but the graph is now equipped with energies, as well; that is, the energy of each structure is additionally recorded in the vertex that encodes it. The first step is to identify vertices that represent a distinct local minimum. A vertex u is a local minimum if $\forall v \in N(u)$ and $\forall v \in V, e(u) \leq e(v)$, where e denotes energy and N denotes neighborhood.

The identification of local minima is important because a basin is tied to a unique local minimum. This is the deepest point in a basin and is also referred to as the *focal minimum*.

In this sense, a basin is a basin of attraction. Other structures/vertices near to the local minimum are attracted to the local minimum. A procedure identifies the vertices drawn to the same local minimum and associates them with the corresponding basin. Specifically, each vertex u is associated with a negative gradient estimated by selecting the edge (u, v), maximizing the ratio $[e(u) - e(v)]/dist(u, v)$ ($dist$ is the distance between two vertices measured via RMSD). From each vertex u that is not a local minimum, the negative gradient is iteratively followed (the edge that maximizes the above ratio is selected and followed) until a local minimum is reached. Vertices that reach (by this mechanism) the same local minimum are assigned to the (same) basin associated with that minimum.

We note that this approach, which takes into account energy as summarized above and identifies basins, has first appeared in [27] but was used to advance a problem known as estimation of model accuracy (model refers to a structure in this term), which we address in later chapters and utilized as an evaluation setting. In this chapter, we leverage our understanding that the detected basins can be treated as the states and so can be employed to construct an MSM, as we now describe in detail.

5.2.4 MSM Construction

After the assignments of structures into states, state-to-state transition probabilities are computed by utilizing the MD trajectories. In a given MD trajectory, suppose a structure S_A is followed by a structure S_B. Now, let us assume that clustering has revealed that structure S_A maps to state St_i and structure S_B maps to state St_j. Then, the transition from S_A to S_B contributes one count to the transition from state St_i to state St_j. Since many structures may map to the same state, the various MD trajectories increase such counts of transitions between states. These counts are normalized to obtain transition probabilities between the clustering-identified states. In summary, let us consider that clustering has yielded Y disjoint states St_1, St_2, ..., St_Y. A matrix of conditional transition probabilities between these states is estimated from the simulation trajectories x_t [20]. The transition matrix, $\mathbf{T} \equiv (\mathrm{P}_{ij})$: $\mathrm{P}_{ij}(\tau) = Probability\ (\mathrm{x}_{t+\tau} \in St_j \mid \mathrm{x}_t \in St_i)$, where τ is the chosen lag

time. The resultant MSM is subjected to rigorous analysis that focuses on model selection, estimation, and validation. The purpose of this analysis is to verify whether the constructed model is capable of making reliable predictions regarding the dynamics of the system under observation.

5.3 Evaluation Dataset

We carry out the evaluation of various MSMs constructed over a dataset of $30,000$ molecular structures of the Met-enkephalin (Met-Enk) peptide. Met-Enk is a naturally-occurring opioid (5 amino acids long) that mediates pain and opiate dependency by interacting with opioid receptors [95]. Interest in Met-Enk dynamics is due to a hypothesis that the peptide is highly flexible and possibly involved in many more interactions than presently known. The $30,000$ structures are obtained from three MD trajectories. Each trajectory starts from a different experimentally-known structure found in the Protein Data Bank (PDB) under entry with identifiers 1PLW, 1PLX, and 2LWC, respectively. The MD simulations were carried out at 300K (room temperature) and at standard atmospheric pressure in the AMBER [15] simulation package, using all-atom detail and immersing the peptide in explicit solvent. The time step in each simulation was 1 fs, and each simulation was run for 10 million steps. Structures were saved every 1000 fs (1 ps), resulting in $10,000$ structures collected from every MD simulation (a total of $30,000$ structures collectively in 3 trajectories).

5.4 Evaluation of MSM

5.4.1 Convergence Analysis

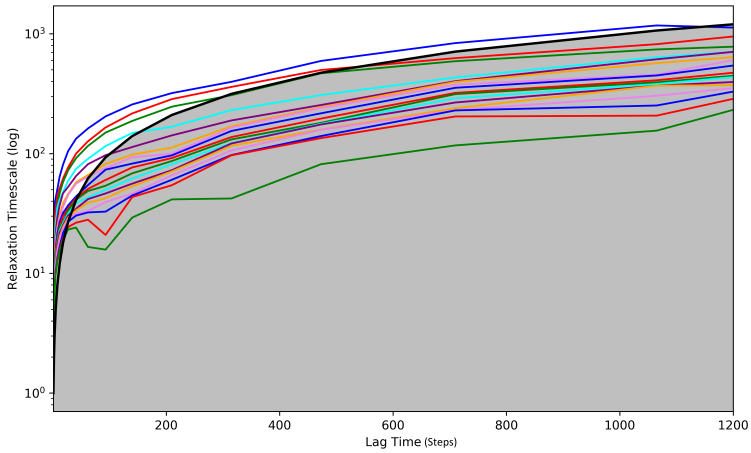
The quality of MSMs obtained can be evaluated to determine the impact of the various state identification techniques. First, we do so via the convergence analysis that tests whether the duration of the lag time is sufficient to guarantee that the state space discretization maintains the Markov property that the system is memory-less; that is, the conditional probability distribution of future states depends only upon the current state and not on

prior states [19]. According to this property, if the state space decomposition is accurate, structures within a state inter-convert on timescales faster than the lag time and transition to other states on slower timescales.

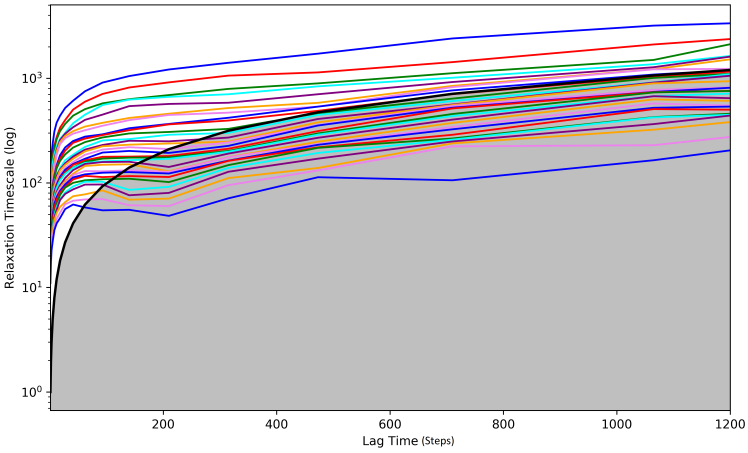
The standard practice is to verify whether an MSM satisfies the Markov property by visually interpreting the generated implied timescale plot of the model relaxation timescale versus model lag time. The desired property is to have an exponential decay in the plot to system equilibrium. With relaxation timescales being physical properties of the system, the ideal case is for the implied timescales to be independent of the lag time. For an ideal model with good discretization, the implied timescales plot exhibit convergence within fewer steps.

Out of the clustering algorithms readily available in PyEMMA, the k-means outperforms uniform time clustering and regular space clustering, whereas the community detection algorithm GMM also outperforms them (results can be found in [90]). We consider k-means as the baseline. Figure 5.2 shows the models obtained when states are identified by (a) Louvain's community detection algorithm, (b) PyEMMA's k-means, and (c) basin-based method. Both k-means and Louvain-based models fail to exhibit convergence even after 1000 steps. In contrast, the model obtained when states are extracted via the basin identification-based approach reaches convergence early (after about 700 steps). This indicates that the best MSM of the Met-Enk equilibrium dynamics is the one obtained when states are identified as basins in the energy landscape.

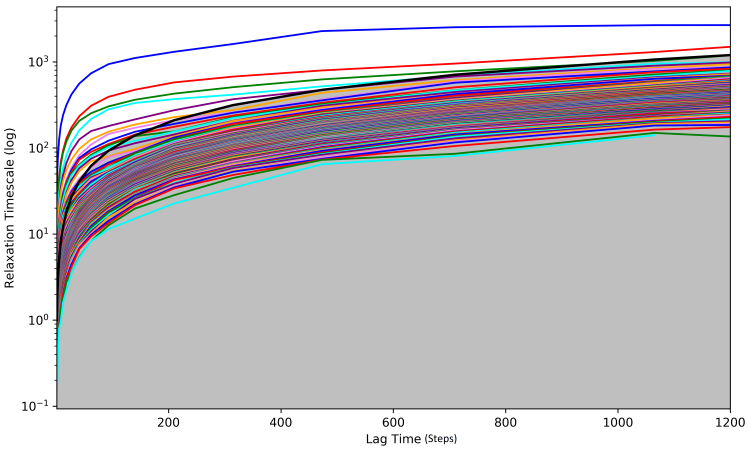
Another way to test for the Markov property is to conduct the Chapman-Kolmogorov (CK) test which compares the transition probability of different states for increasing lag times. Ultimately, the goal is to establish whether the lag time is sufficiently large to make the selected state decomposition Markovian. We conduct both evaluations to determine which state-space decomposition strategy results in a higher-quality MSM of molecular dynamics.



(**a**) Louvain-based



(**b**) k-means-based



(**c**) Basin-based

Figure 5.2: Implied timescale plot obtained when carrying out state space discretization via (a) the Louvain community detection algorithm, (b) PyEMMA's k-means, and (c) the basin identification algorithm. The cutoff region (above which any curve representing a good discretization should be) is shown in gray.

5.4.2 MSM Model Visualization

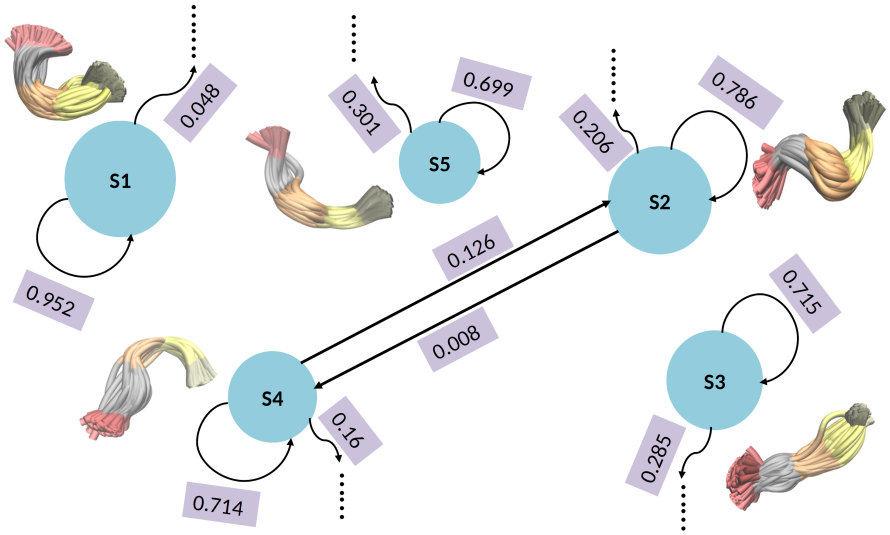


Figure 5.3: Visualization of the best model of dynamics, limiting the visualization to the top five states. Structures in each of the top five states of the best model are shown superimposed over one another (Red: N terminus, Silver: C terminus).

The best MSM of the Met-Enk equilibrium dynamics is obtained when states are identified as basins in the energy landscape, which is visualized in Fig. 5.3. To make the visualization uncluttered, rather than showing all the 173 states (173 basins are identified with the basin identification approach), the visualization is limited to the five states with the highest self-transition probabilities. Altogether these states contain approximately 16% of the $30,000$ Met-Enk structures. Fig. 5.3 shows these states as disks, with the sizes of disks indicating the relative differences in sizes (number of structures) of the corresponding states. Transitions among these states and others not shown states are drawn as arcs, with transition probabilities annotated, as well. Fig. 5.3 makes it clear that the peptide visits several long-lived states; each of the shown states has self-transition probabilities > 0.69.

State 1 (S1) has a very high self-transition probability of over 0.9. The other states have slightly lower self-transition probabilities and comparatively-low probabilities of transition to other states. Fig. 5.3 also manifests the actual structures in each of these five states. All the structures in a state are superimposed over one another. Each amino acid is color-coded according to its position (red denotes the N terminus and silver the C terminus).

5.4.3 Relating Long-lived States to Experimentally-known Structures

Table 5.1: Structures in each model-identified state are compared to the first structure in each of the three experimentally-identified NMR ensembles (PDB IDs shown) deposited in the PDB. Average and minimum RMSDs (Å) are reported. Entries in bold highlight the lowest RMSD.

	1PLW		**1PLX**		**2LWC**	
	avg	min	avg	min	avg	min
S1	2.813	1.181	1.920	**0.712**	2.023	**0.543**
S2	1.018	**0.516**	1.905	1.411	1.895	1.539
S3	1.540	1.167	1.904	**0.845**	1.376	**0.364**
S4	2.229	1.244	1.748	1.214	1.817	1.199
S5	1.511	**0.741**	1.575	1.321	1.480	1.280

Finally, the long-lived states identified by the model are compared with structures identified in the wet laboratory. This analysis provides insight into which states capture already known ones, and which may constitute new, unknown states of the Met-Enk peptide. Known structures are under PDB IDs 1PLW, 1PLX, and 2LWC. Table 5.1 compares each identified state to each known structure. It is worth noting that each of the PDB entries indeed contains a small NMR ensemble of 20-80 models (NMR refers to Nuclear Magnetic Resonance). So, all structures in a model-identified state are compared via RMSD (after optimal superposition removes rigid-body differences) to the first structure in each NMR ensemble.

Table 5.1 reports the minimum and average RMSD and highlights in bold RMSDs under 1Å. RMSDs below 1Å can be used to establish a correspondence between model-identified states and experimentally-identified structures. For instance, S2 can be considered to capture 1PLW, S1 captures both 1PLX and 2LWC, S3 captures 1PLX, 2LWC, and S5 captures 1PLW. With a more stringent cutoff of 0.72Å, only S1, S2, and S3 capture the three experimentally-known structures, whereas S4 and S5 constitute new states. This type of analysis suggests that the MD simulation probes novel states of Met-Enk; at least one of them, S4, as shown in Fig. 5.3, transitions to a state (S2) captured in the wet laboratory.

5.5 Summary

The work described in this chapter relates to the importance of leveraging the structural energetics readily available from MD simulations (rather than ignoring it) to identify states in the crucial step of state-space discretization in MSM construction. The evaluation also suggests that such states, tied to the concept of basins in the energy landscape probed by MD simulations yield MSMs of better quality and thus more accurate models of structural dynamics. It is worth mentioning that as our application has been on a small peptide, it is important to make a case for the generalization of this approach by extending to larger and richer biological systems. In Chapter 9, we elaborate on employing this strategy for organizing structures of antibody-antigen bound molecular systems for which MD data has been provided to us by computational and molecular biology collaborators.

Chapter 6: Feature-oriented Factorization Methods to Organize Protein Structure Space

This chapter investigates a new direction of research that improves upon graph embedding-based methods in both time cost and memory demand issues, as well as upon the challenge of identifying functionally-relevant structures from sparse datasets while still utilizing adjacency matrix information. We demonstrate here the capability of a matrix factorization approach in organizing the protein structure space and allowing the detection of functionally-relevant structures, even when data are sparse. In particular, we show that a feature-based, factorization-based method outperforms basin-based and other state-of-the-art methods. Work described in this chapter has been disseminated in [29].

6.1 Non-negative Matrix Factorization (NMF)-based Framework to Organize Protein Structure Space

We first describe the main ingredients of the NMF-based framework at a high level. Fig. 6.1 shows that the framework essentially assigns provided structures of a protein molecule into groups, selects a best group, and finally selects the best structure from the selected group. The inspiration for the framework is the classic setting, where a protein molecule is active in one structural state, and we seek to determine this state via a representative structure selected with no prior information from the structure data.

The various structures illustrated in Fig. 6.1 are colored in red, green, and blue, indicating their corresponding energy levels; red indicates higher energy, blue indicates lower energy, and green indicates a level between red and blue. First, the framework extracts features from the given structures and stores them in an initial feature matrix, X. In the next step, the NMF-based framework decomposes the feature matrix into two non-negative

matrices, W and H. The factor matrix W contains the basis patterns. Linear combinations of these basis patterns describe and reconstruct each structure in the initial matrix. These basis patterns define different structure groups/clusters. The framework assigns a structure s to a structure-group G if s is closest to the basis pattern representing the structure-group G. Next, it selects a structure-group via structure selection strategies and evaluates how functionally-relevant the selected group is. Finally, the framework selects a representative structure from the selected group/cluster to represent the best functionally-relevant/biologically-active structural state in the given set of structures.

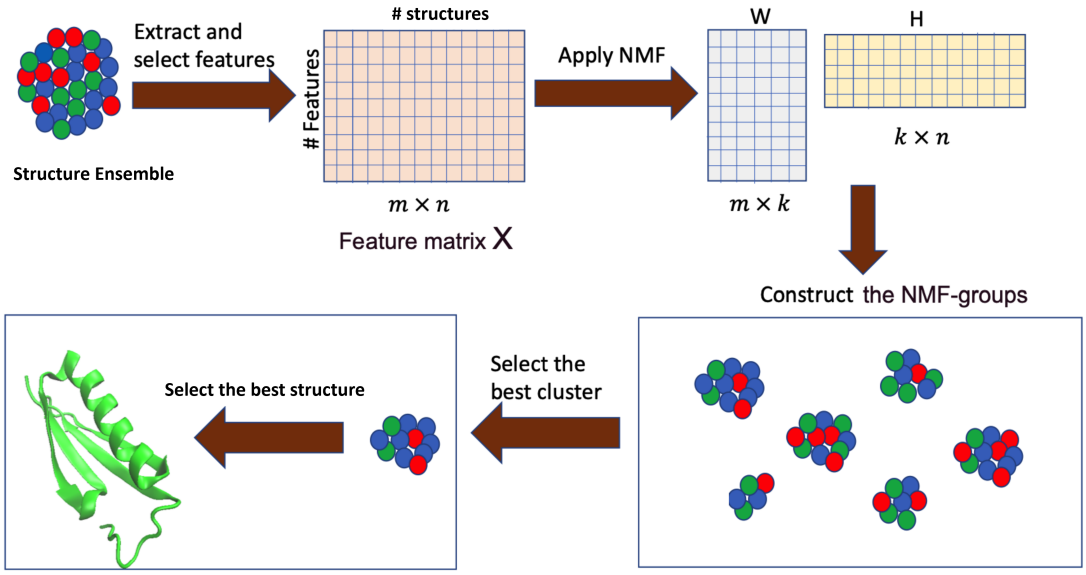


Figure 6.1: Schematic of the NMF-based framework. At a high level, the framework groups structures, selects a best group, and then selects a best structure from the selected group.

6.1.1 Non-negative Matrix Factorization (NMF)

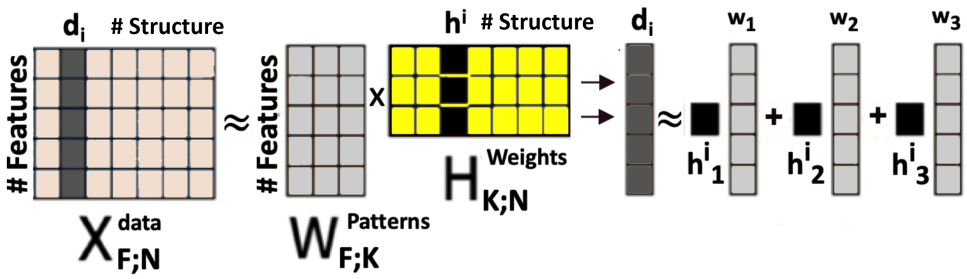


Figure 6.2: Illustration of NMF decomposition of the feature matrix X. The decomposition produces two factor matrices W and H. The columns of W represent basis patterns. Each structure d_i (column in feature matrix X) is expressed as a linear combination of the basis patterns with coefficients found in the corresponding column matrix H.

NMF is a widely-used unsupervised learning method for dimensionality reduction and feature extraction. The non-negative data, a matrix of dimensions features $\times$ samples, is factorized into two non-negative low-rank matrix factors, W and H, with a small inner dimension K. For a given data $X \in \mathbb{R}_{+}^{F \times N}$ (features $\times$ samples), NMF approximates X with the product of W and H, by minimizing the Frobenius norm (indicated by $||.||_{\mathcal{F}}$),

$$\epsilon = min||X - WH||_{\mathcal{F}}^{2} \tag{6.1}$$

or, $X_{ij} = \sum_{s=1}^{K} W_{is}H_{sj} + \epsilon_{ij}$, where, ϵ_{ij} is the error of the approximation, which is normally distributed. In this way, each column of X (representing a sample) is expressed as a linear combination of the basis latent patterns (the columns of W) and its weights (the corresponding column of H) as in Fig. 6.2. The non-negativity forces NMF to learn local parts of the object (described in X) [96], hence, to extract easily interpretable and sparse latent features, which makes NMF a preferable technique when explainability is important.

NMF is underpinned by a statistical model of superimposed components (the number of these components is equal to the size of the small dimension K) that can be treated as latent features in Gaussian, Poisson, or other mixture models. NMF minimization (with a specific distance metric $||...||_{dist}$) is equivalent to the expectation-minimization (EM) algorithm. In this probabilistic interpretation of NMF, the manifested variables are the columns $d_1, ..., d_N$, of the matrix, X, generated by the latent variables, $h_1, ..., h_K$, that are the columns of the matrix, H. Specifically, each observable x_i is generated from a probability distribution with mean $\langle d_i \rangle = \sum_{s=1}^{K} W_{is} h_s$, where K is the number of the latent variables [96]. Thus, the influence of h_s on d_i is through the basis patterns represented by the columns of the matrix W, $w_1, ..., w_K$.

In our case, the basis patterns, represented by the columns of the matrix W, can be thought of as pseudo-structures (not necessarily in the structure ensemble) whose linear combinations span the entire ensemble space. Then, each structure is a linear combination of these pseudo-structures with coefficients given by the corresponding columns of matrix H. The NMF optimization problem, $min||X - WH||_{dist}$, can be solved by various algorithms, such as the multiplicative update [96], block principle pivot [97], and projected gradient methods [98]. All these methods follow alternating non-negative least squares [99], where for each iteration, one of the factors is fixed and the other updates.

6.1.2 Feature Extraction

We encode each structure with 39 energy-based features covering three different categories. Of these 39 features, 9 are potential energy-based, 17 are Rosetta REF2015 energy terms, 9 features are based on the consistency between the actual and predicted values of structures, and 3 are contact-based scores. One more feature comes from Rosetta Score12's total energy. The potential energy-based and consistency-based terms are used in a support vector machine-based single structure selection method [50].

1. Features Based on Energy Functions (27 features): Eighteen of these features are collected from Rosetta REF2015 and Score12 energy functions. We use raw values

of 17 energy terms from Rosetta REF2015 energy function. We also use REF2015 and Score12 total energy. The total energy values are computed using the weighted energy terms. The 9 potential energy terms are the following: we use a side-chain orientation-dependent potential *RWplus* [100]; a distance-dependent potential DFIRE [101]; dDFIRE [102] adds orientation-dependency to DFIRE; three features from GOAP [56] that are distance and orientation-dependent all-atom potential. GOAP contains DFIRE and an angle-dependent term. We use the overall GOAP potential, the DFIRE term, and the angle-dependent term as three features. OPUS-PSP [103] is an orientation-dependent all-atom potential that includes an orientation-dependent packing energy term and a Lenard-Jones repulsive energy term. Both of these energy terms and the total energy constitute three more features.

2. Features Based on Structural Consistency (9 features): We use nine features based on structural consistency. Four of these are secondary structure-based features. We use PSIPRED to compute the secondary structure of each target from its primary sequence. We extract the secondary structure of the structures for each protein using DSSP. We keep track of the number of matches in secondary structure elements (beta sheets, alpha helices, and coils) between PSIPRED and DSSP calculations. We normalize the match counts for the three secondary structure elements by the length of the corresponding sequence and use the normalized match counts as features. The fourth feature is computed as follows. When there is a match between the PSIPRED and DSSP calculations, we store the score of each secondary element computed by PSIPRED. We add these scores and use the total score as a feature. We compute 5 features based on solvent accessibility. To determine the buried (B) or exposed (E) state of a residue (with respect to solvent accessibility), we use RaptorX, which is a Deep Convolutional Neural Fields (DeepCNF)-based webserver [104]. We specify each residue as either buried or exposed based on the probabilities computed by RaptorX. We also calculate the relative solvent accessibility of a residue of each structure using DSSP. To specify a residue as buried or exposed, we divide the calculated solvent accessibility by the total solvent accessibility [105]. A cut-off value of 25% has been used in this process. Two more features are computed from the number of B and E matches

computed by RaptorX and DSSP. We use the length of the corresponding sequence to normalize these features. When there is a mismatch between RaptorX and DSSP results, we note the corresponding probabilities. We use the combined probability as another feature. We also compute Pearson's correlation, and cosine similarity between the number of secondary structure elements and solvent accessibility states computed from the primary sequence and from the structures, and used them as two more features.

3. Features Based on Residue Contacts (3 features): We use three features based on contact scores. We use relative contact order which is defined as the average sequence distance between all pairs of contacting residues and normalized by the total sequence length [106]. We extract two more features by following the process mentioned in [107]. We use RaptorX-contact [104] to predict the contacts from the amino-acid sequence. We treat the top 10 RaptorX-predicted contacts as references. We determine true positives (TP), false positives (FP), and False negatives (FN) from the structures using the top 10 pairs of amino acids. If these 10 pairs are also found in contact with a structure, we have a true positive. False negatives increase when the top 10 pairs are not found in contact with a structure. Finally, false negatives are found if the contacts in a structure are not found in the reference structures. We calculate precision and recall and use them as features; precision is defined by $\frac{TP}{TP+FP}$, and recall is defined as $\frac{TP}{TP+FN}$.

6.1.3 NMF-backed Structure Groups

We construct structure clusters/groups using the factors of NMF on the structure matrix. The structure matrix is essentially a feature matrix with features extracted from the structures. In the structure matrix X, the rows correspond to the features of structures and the columns represent distinct structures. Therefore, each cell $X(i, j)$ represents the i-th feature of the j-th structure. To satisfy the non-negativity, we shift the negative values of any feature in the feature matrix X into positive space, and then apply NMF to the feature matrix. The basis patterns, i.e., the columns of the matrix W define the structure-groups backed by NMF. To identify the basis pattern that a structure d_i belongs to, we note the

maximum value of the corresponding hidden variable h_s^i (Fig. 6.2). A structure d_i belongs to the group/cluster defined by the basis pattern w_i; basis pattern w_i is associated with the maximum value of the column h_s^i corresponding structure d_i.

6.1.4 Structure-Group Selection

Once the membership of each structure to a given pattern w_i is established, we select a structure group/cluster. The structure-groups are characterized by their associated metrics. We compute two metrics. The first metric, median absolute deviation (MAD), measures the spread of data within a group. Lower MAD value indicates less variability. Another desirable property of MAD is its robustness against outliers. As our methods rely on the principle of consensus, we choose MAD for one of our metrics, described below.

NMF-MAD

Median Absolute Deviation (MAD) measures the variability of data samples while not considering any application-oriented characteristics of the data samples. This metric is also resilient to outliers. Let C be a group and x_i be a structure from C, then the MAD measure for C is defined as,

$$\text{MAD}(C) = b \cdot median(\{dist(x_i, median(C)) \mid x_i \in C\}) \tag{6.2}$$

where b is a constant scale factor that depends on the probability density of the observed samples [108]. For each group C, we compute the MAD score for all structures in the group and take the average of the scores. Considering the average MAD score as a characteristic of the group, they are ranked based on increasing MAD score and the top group (highest MAD score) is selected.

NMF-Rank

The second selection method is based on structure-specific characteristics. The average of the energies of structures of a group is defined as the *average energy* of the group. The *minimum energy* of a group is defined as the minimum of the energies of all structures assigned to that group. The size of a group is the number of structures populating the group. Three stages of rankings are performed on the groups to identify the group that represents the structures that are similar to the experimentally-known structure. The groups are ranked based on increasing size and the top 5 groups are selected. We rank the selected groups again based on minimum energy and select the top 3 groups. Finally, these 3 groups are sorted in ascending order of average energy, where the top group is offered as prediction.

6.1.5 Functionally-relevant Structure Selection

We use the concept of density score of a structure [109] to select the best structure from the top structure-group. Let a structure-group consists of n structures and a structure x_i belongs to this group. The density score DS_i of structure x_i is defined as follows,

$$DS_i = \frac{\sum_{j=1}^{n} r_{ij}}{n} \tag{6.3}$$

The term r_{ij} denotes the pairwise root-mean-squared-deviation (RMSD) between structure x_i and structure x_j $(1 \leq i, j \leq n)$. We normalize the density scores so that the scores are in a range between -1 and 1. We compute the normalized density score $DS_i^{'}$ as following.

$$DS_i^{'} = \begin{cases} \frac{(DS_i - DS_{median})}{DS_{median} - DS_{min}} & \text{if } DS_i < DS_{median} \\ 0 & \text{if } DS_i = DS_{median} \\ \frac{(DS_i - DS_{median})}{DS_{max} - DS_{median}} & \text{if } DS_i > DS_{median} \end{cases} \tag{6.4}$$

The terms DS_{min}, DS_{max}, and DS_{median} denote the minimum, maximum, and median density scores, respectively. We assign weight W_i to each structure based on its normalized density score. The weight W_i is defined by $W_i = e^{-r(DS_i^{'})}$, where r is a constant (we use $r = 5$). We rank the structures in decreasing order of their weights, and offer the top structure as the best. If a structure set is of sparse distribution, then a group might consist of only two structures. In such a scenario, we select the structure with the lower energy.

6.2 Evaluation Dataset

Table 6.1: Benchmark dataset (* denotes proteins with a predominant β fold and a short helix). The chain extracted from a multi-chain PDB entry is shown in parentheses.

Difficulty	#	**PDB ID**	**Fold**	**Length**	$\lvert\Omega\rvert$	**min_dist** (Å)
Easy	1	1ail	α	70	53,544	0.50
	2	1dtd(B)	$\alpha+\beta$	61	57,810	0.51
	3	1wap(A)	β	68	51,810	0.60
	4	1tig	$\alpha+\beta$	88	52,071	0.61
	5	1dtj(A)	$\alpha+\beta$	74	53,497	0.68
Medium	6	1hz6(A)	$\alpha+\beta$	64	57,449	0.72
	7	1c8c(A)	β^*	64	53,297	1.08
	8	2ci2	$\alpha+\beta$	65	52,187	1.22
	9	1bq9	β	53	53,629	1.31
	10	1hhp	β^*	99	52,128	1.52
	11	1fwp	$\alpha+\beta$	69	53,103	1.56
	12	1sap	β	66	51,182	1.75
Hard	13	2h5n(D)	α	123	51,450	2.05
	14	2ezk	α	93	50,167	2.56
	15	1aoy	α	78	52,189	3.27
	16	1cc5	α	83	51,666	3.95
	17	1isu(A)	*coil*	62	60,329	5.53

We evaluate our methods on two datasets. First, we evaluate on 17 benchmark proteins of different folds and lengths (number of amino acids, Table 6.1). We use Rosetta *template-free* protocol to generate $50,000$ to $60,000$ structures per target. The 4-letter PDB id for each protein is in column 3. These proteins represent *easy*, *medium*, and *hard* cases for Rosetta. The difficulty levels (*easy*, *medium*, *hard*) are informed by the performance of an incremental clustering-based structure selection [27]. Further specification is also available in Section 4.1. The size of structure ensemble, $|\Omega|$ for each protein is in column 6. Column 7 is the minimum distance, *min_dist*, between the structures generated by Rosetta and an experimentally-known structure in corresponding PDB entries. The *min_dist* informs about the varied performance that Rosetta achieves for each protein.

Table 6.2: CASP dataset. CASP target IDs are shown in Column 2. PDB ID, Length, and Minimum RMSD over structure dataset to corresponding experimentally-known structure are shown for each target. Experimentally-known structures only available in the CASP website [110] are marked by asterisks.

#	**Target ID**	**PDB ID**	**Length**	$\|\Omega\|$	**min_dist** (Å)
1	T1008-D1	6msp	77	$55,000$	1.54
2	T0886-D1	5fhy	69	$55,000$	4.92
3	T0953s1-D1	6f45	67	$55,000$	5.81
4	T0960-D2	6cl5	84	$55,000$	5.98
5	T0898-D2	**	55	$43,435$	6.0
6	T0892-D2	5nv4	110	$36,860$	6.62
7	T0953s2-D3	6f45	77	$55,000$	7.52
8	T0957s1-D1	6cp8	108	$45,000$	4.91
9	T0897-D1	**	138	$25,000$	8.30
10	T0859-D1	5jzr	113	$40,000$	9.06

Besides, we consider 10 free modeling targets from CASP 12 and 13 (Table 6.2). Several of these targets such as T0953s2, T0957s1, T1008 are determined as hard targets [111,112].

6.3 Evaluation Results

6.3.1 Group Selection Results

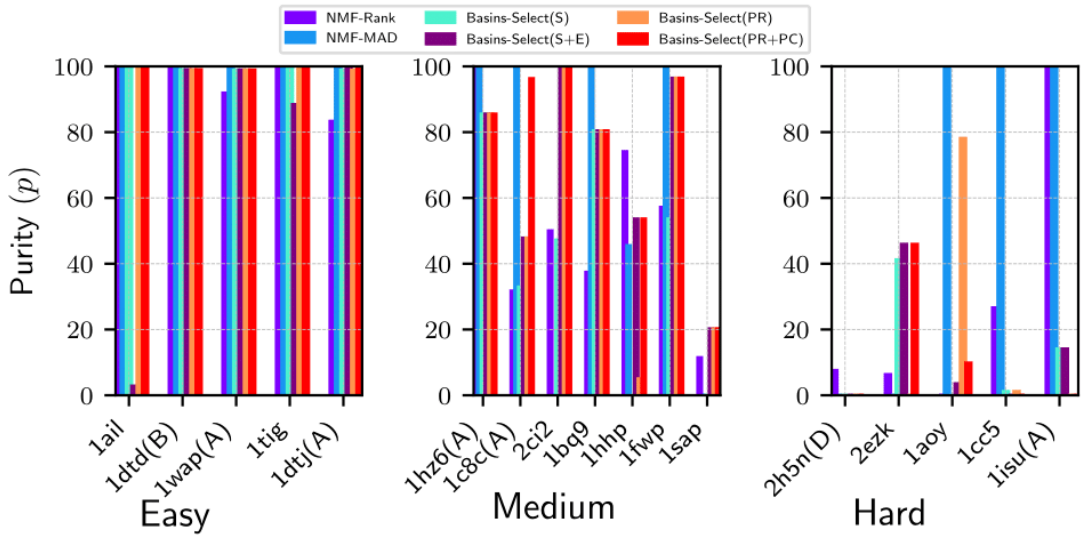


Figure 6.3: Comparison of four unsupervised basin-based and two NMF-based structure selection methods (y-axis tracks the purity of the top basin/group/cluster detected by each selection method, while x-axis tracks the PDB ID of each target protein)

We compare (in terms of purity as described in Section 4.4.1) the NMF-based methods: NMF-MAD and NMF-Rank, with four unsupervised basin-based methods presented in [27] that leverage the concept of energy landscape to construct structure-groups. At first, the basins are extracted from the underlying energy landscape of a protein structure space, and then structure selection is performed by ranking and selecting the basins based on their size (Basins-Select(S)), and size and energy (Basins-Select(S+E)). A basin consists of structures and is considered a structure-group/cluster. Specifically, Basins-Select(S) ranks the basins

based on decreasing basin-size and selects the top basin. Basins-Select(S+E) ranks the basins first by decreasing size and selects top b basins where b is user-defined. Then, the b basins are further ranked based on increasing energy and the top basin is selected. Since the goals of obtaining lower energy and larger size pose two conflicting objectives, two Pareto-based selection methods (Basins-Select(PR) and Basins-Select(PR+PC)) are devised by utilizing the concept of dominance described in Section 4.3.

Fig. 6.3 compares NMF-MAD and NMF-Rank with four basin-based unsupervised structure selection methods on 17 proteins (5 easy, 7 medium-difficulty, 5 hard). All methods perform comparably well on the easy test cases. For all the 5 test cases, NMF-MAD achieves 100% purity. NMF-Rank shows more than 90% purity in 4 test cases, and more than 80% purity for the remaining. The four basin-based methods achieve good purity scores (from 88% to 100%). However, one method, Basins-Select(S+E), shows poor performance (2.8% purity) on an easy test case (*1dtd(B)*).

For medium-difficulty proteins, NMF-MAD performs better than basin-based methods in 4 out of 7 cases. The structure sets for medium-difficulty proteins contain comparatively lower number of structures that are similar to the experimentally-known structure. For instance: *1bq9* with a structure set of size $53,629$ contains only 1.6% of such structures. Similarly, the percentage is 2.5% for the structures in a structure set of size $52,128$ (*1hhp*). Even in those cases, the best that the basin-based methods achieve for the protein under PDB ID *1bq9* is 80.4% purity, whereas NMF-MAD achieves 100% purity. In the other instance, NMF-Rank scores 74.1% for the protein under PDB ID *1hhp*. For the same protein, the best purity score by any basin-based method is 53.6%. NMF-Rank and NMF-MAD both outperform the basin-based structure-group selection methods.

The utility of NMF-based structure selection methods can be better realized when we consider the hard test cases. The structure sets for hard proteins exhibit the highest level of sparsity. The number of structures that are close to the experimentally-known one found in these structure sets is below 6%. Additionally, the best quality structure that Rosetta could sample for these proteins is further away from the experimentally-known one compared to

the best structures under the category of easy and medium-difficulty proteins. A lack of enough good quality structures and the resulting sparsity in the structure set for hard proteins make the task of functionally-relevant structure selection more challenging. For these challenging cases, NMF-based methods significantly outperform basin-based methods in 4 out of 5 test cases. For instance, NMF-MAD achieves 100% purity for both the proteins with PDB IDs *1cc5* and *1isu(A)*, whereas the basin-based methods can achieve 1.14% and 14.1% purity at best. On the hardest test case, the protein with PDB ID *2h5n(D)*, NMF-Rank achieves 7.54% purity whereas the basin-based methods capture 0% purity. Such an outstanding performance by NMF-based methods on the hard test cases emphasizes the utility of NMF for grouping structures for functionally-relevant structure selection.

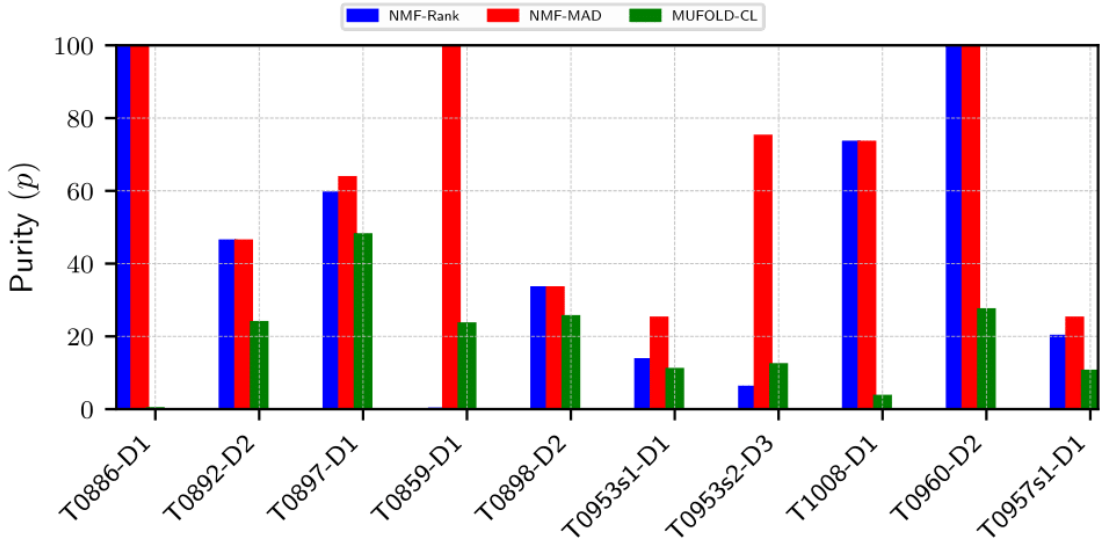


Figure 6.4: Comparison of two NMF-based structure selection methods and MUFOLD-CL on CASP targets (y-axis tracks the purity of the top basin/group/cluster predicted by each selection method, while x-axis tracks the ID of each target protein).

Fig. 6.4 compares NMF-MAD, NMF-Rank and MUFOLD-CL (a state-of-the-art Estimation of Model Accuracy (EMA) method) on the CASP targets. NMF-MAD and NMF-Rank outperform MUFOLD-CL in 10/10 and 8/10 test cases, respectively.

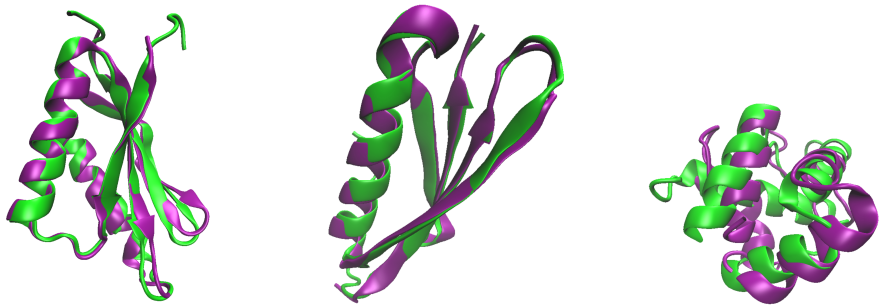
6.3.2 Single Structure Selection Results

Table 6.3: Results for Quantitative Comparison on CASP targets. Columns 2 and 3 show loss in lRMSD, GDT-TS, TM-score for the best structure selected by NMF-MAD and MUFOLD-CL, respectively.

Targets	**NMF-MAD lRMSD Loss (Å), GDT-TS loss, TM Loss**	**MUFOLD-CL lRMSD Loss (Å), GDT-TS loss, TM Loss**
T0886-D1	$2.77, 0.029, 0.03$	$8.51, 0.03, 0.02$
T0892-D2	$5.5, 0.007, 0.03$	$6.32, 0.025, 0.032$
T0897-D1	$3.5, 0.0, 0.003$	$6.5, 0.014, 0.017$
T0859-D1	$4.44, 0.011, 0.032$	$8.62, 0.022, 0.014$
T0898-D2	$5.2, 0.027, 0.01$	$5.1, 0.027, 0.007$
T0953s1-D1	$5.4, 0.011, 0.017$	$8.34, 0.019, 0.017$
T0953s2-D3	$4.88, 0.029, 0.039$	$4.2, 0.032, 0.024$
T1008-D1	$0.42, 0.012, 0.02$	$6.2, 0.019, 0.009$
T0960-D2	$3.51, 0.011, 0.016$	$5.6, 0.012, 0.02$
T0957s1-D1	$4.68, 0.008, 0.027$	$6.64, 0.074, 0.091$

We compare NMF-MAD and MUFOLD-CL for best structure selection in terms of lRMSD loss, GDT-TS loss, and TM-Score loss. We report the loss incurred with random selection and the average lRMSD of the selected group as baselines. Table 6.3 shows a quantitative comparison between NMF-MAD and MUFOLD-CL on the CASP dataset in terms of lRMSD loss, GDT-TS loss, and TM-score loss. NMF-MAD outperforms MUFOLD-CL in 8/10 cases in terms of lRMSD loss, in 9/10 cases in terms of GDT-TS loss. In terms

of TM-score loss, MUFOLD-CL performs better than NMF-MAD in 5/10 cases, NMF-MAD outperforms MUFOLD-CL in 4/10 cases, while both perform somewhat similar in the remaining case.



1tig (0.32Å) **1hz6(A) (0.35Å)** **1cc5 (1.93Å)**

Figure 6.5: Structures under each difficulty category (easy, medium, hard) selected by NMF-MAD are shown superimposed over known wet-laboratory structures under PDB ID 1tig, 1hz6(A), and 1cc5. The corresponding experimentally-known structure is colored in purple, and the best structure selected by NMF-MAD is colored green (with the RMSD loss reported in parentheses).

Fig. 6.5 shows NMF-MAD-selected structures (colored in green) for each difficulty level (easy, medium, hard) superimposed over the corresponding experimentally-known structures (colored in purple) resolved in the wet laboratory and deposited in Protein Data Bank. The best structure selected (Fig. 6.5) by NMF-MAD for the easy protein target with PDB ID *1tig* and for the protein with PDB ID *1hz6(A)* under the medium-difficulty category are structurally similar to their experimentally-known structures. For the hard protein target (PDB ID *1cc5*), the selected structure, albeit not quite close to the experimentally-known one as for the easy and medium-difficulty cases, but is not significantly deviant too.

6.4 Summary

The work described in this chapter marks the opening of a new avenue in organizing the structure space of dynamic molecules like proteins and also demonstrates the potentiality of factorization-based methods for organizing as well as detecting functionally-relevant structures. However, the method summarized in this chapter relies heavily on features and parameters. For instance, one needs to provide the number of groups as a parameter. Chapters 7 and 8 provide ways to extend factorization-based frameworks that promise to resolve these issues and so support a variety of applications for advancing knowledge in molecular biology.

Chapter 7: Non-parametric, Feature-free Factorization-based Method to Organize Protein Structure Space

The work presented in this chapter capitalizes on the decomposition-based framework and formulate a novel method relying on matrix-factorization that is both feature-free and non-parametric. Specifically, we describe a novel symmetric non-negative matrix factorization (SNMF)-based framework. The work described here has been disseminated in [30].

7.1 Symmetric Non-negative Matrix Factorization (SNMF)-based Framework to Organize Protein Structure Space

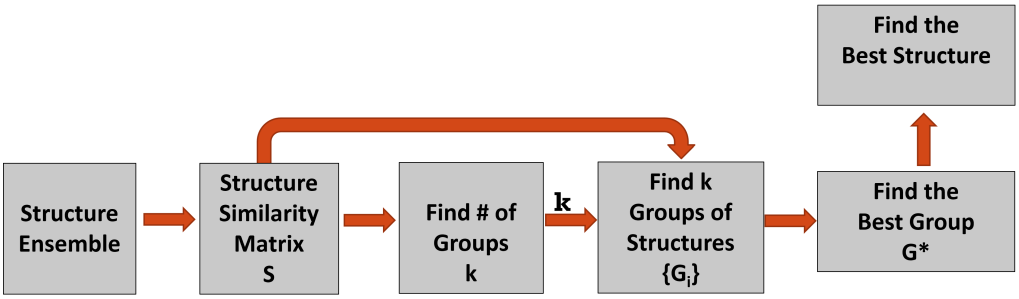


Figure 7.1: The schematics of the framework operationalized by SNMF-DS

We propose a novel method, SNMF-DS, that utilizes symmetric non-negative matrix factorization (NMF) in the graph embedding setting for structure selection. The method is fully non-parametric and employs the eigen-gap statistic to automatically determine the

number of components for matrix factorization. The framework that SNMF-DS operationalizes for structure selection proceeds in three stages.

- In the first stage, given the tertiary structures of a target protein, they are organized into groups $\{G_i\}$.
- The second stage utilizes structure energies to discriminate among the groups and select a best group G^* from $\{G_i\}$.
- In the third stage, a weighting scheme associates weights with structures in the best group to select a best structure from the best group.

The best structure is selected as the closest approximation to the functionally-relevant structure available in the given structure ensemble. Conceptually, the framework is related in Figure 7.1.

Figure 7.1 shows that the input to SNMF-DS are the Cartesian coordinates of the structures. These are utilized to construct a structure similarity matrix, which is then subjected to an eigen-gap heuristic in order to determine k, the number of groups. We use this information of k to perform symmetric non-negative matrix factorization. The resultant factor (W) is used to elucidate k groups in which structures are organized by finding the group membership from the factor matrix, W. The method is non-parametric, as the value for k is determined automatically by exploiting the eigen-gap heuristic.

7.1.1 From Structure Ensemble to Structure Similarity Matrix

SNMF-DS takes as input the structures of a given protein target whereas each structure is a tertiary structure. Each tertiary structure is stripped down to its main-chain carbon atoms (CA atoms), discarding side-chain atoms and other backbone atoms. This reduction improves the cost of computing the similarity matrix $S_{n\times n}$ (in Figure 7.1) which is symmetric and contains at entry $S_{i,j}$ the similarity between two structures i and j in the given structure set. Specifically, $S_{i,j} = \frac{1}{\text{RMSD(i,j)}+\epsilon}$ where RMSD refers to the root-mean-squared-deviation that measures the dissimilarity between two structures, and ϵ refers to an

infinitesimally-small constant. While different metrics other than RMSD can be used, we elect to use RMSD due to its popularity in comparing molecular structures. RMSD averages the Euclidean distance over the number of atoms (CA atoms in our case). The structures are first optimally superimposed over an arbitrarily-chosen structure (we use the first in the set of given structures) to minimize differences due to rigid-body motions (whole-body translation and rotation). In this way, the RMSD values capture the internal structural differences rather than differences due to whole-body motions in space. The reason for using ϵ is to guard against, in principle, a division by 0 in the case of two identical structures.

7.1.2 From Structure Similarity Matrix to number of Structure Groups

In this step, SNMF-DS finds the number of structure groups in a non-parametric manner using eigen-gap heuristic [113]. The pairwise structure similarity matrix S is used to search for the m nearest neighbors of each structure. Specifically, Algorithm 1 demonstrates the eigen-gap heuristic to find k [114]. Some suggestions about the value of m are available in literature [115], such as $log(n) + 1$, $\sqrt{n}$, $2n^{1/d}$ where, n is the number of structures in our setting, and d is the number of coordinates in a structure. For our implementation, we pick $m = \sqrt{n}$.

Finding the m nearest neighbors of each structure in the structure set is instantiating a nearest-neighbor graph (nngraph), where structures are vertices, and edges connect structures to their nearest-neighbors. We note that this graph is not explicitly constructed. Instead, SNMF-DS constructs an adjacency matrix A and a degree matrix D. The entries of A indicate whether pairs of vertices/structures are adjacent or not in the nngraph. Since the nngraph is a finite simple graph, A is a binary matrix with *zeros* on its diagonal (we do not allow a structure to be considered a nearest neighbor of itself). The degree matrix D is a diagonal matrix that contains the degree of each vertex; $D_{i,j} = deg(i, j)$ if $i = j$ and 0 otherwise. A and D are used together to construct the Laplacian matrix, $L = D - A$ of the nngraph and obtain the optimal number k of groups in which to organize structures.

Algorithm 1 EigenGap Heuristic

Require: S, m

$A \leftarrow$ Adjacency Matrix from S, m
$D \leftarrow$ Degree Matrix from A
$L \leftarrow D - A$ //Laplacian Matrix
$L_S \leftarrow D^{-1/2} L D^{-1/2}$ //Symmetric normalized Laplacian Matrix
$\{\lambda, U\} \leftarrow$ Eigen-decompose(L_S) //λ: eigenvalues, U: eigenvectors
$\lambda_d \leftarrow$ diagonal(λ) //extract eigenvalues
sort the elements of λ_d in ascending order
for $k \leftarrow 1 \ldots \lambda_d.length - 1$ **do**
  EigenGap$(k) \leftarrow \lambda_d[k+1] - \lambda_d[k]$
end for
return k that corresponds to the highest peak of the EigenGap(k)-curve

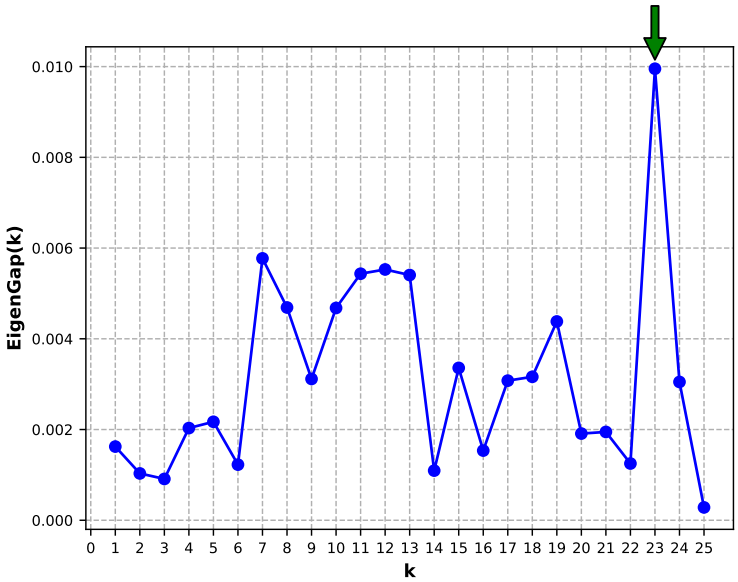


Figure 7.2: The EigenGap curve computed over structures of CASP Target `T1008-D1`. The highest peak is obtained at $k = 23$ (indicated by an arrow), thus identifying 23 groups among the structures. The curve is only shown for the first 25 components out of 500 in the interest of clarity.

Figure 7.2 shows the EigenGap curve computed as described over the structures of a protein target in one of our evaluation datasets. The highest peak is obtained at $k = 23$; so, 23 is the number of groups that is utilized by SNMF-DS to organize the structures as described next.

7.1.3 Organizing Structures into Groups

Conceptually, as shown in Figure 7.3, in this step the SNMF-DS method approximates the similarity matrix S by a lower-rank factorization WW^T. The matrix W is interpreted by SNMF-DS as the cluster membership indicator matrix, which reveals the groups to which structures belong.

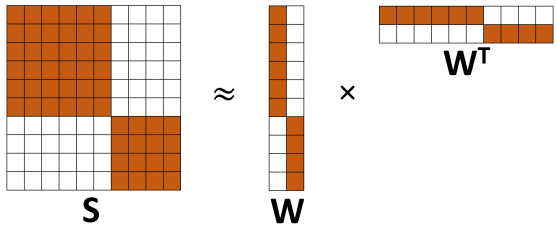


Figure 7.3: An illustration of the factorization of a symmetric non-negative similarity matrix: $S \approx WW^T$ as in Eq. (7.1). Colored region has larger values than white region

We recall that NMF is an unsupervised method which approximates a given non-negative data-matrix, $X \in \mathbb{R}_+^{n \times d}$ by factoring it into two non-negative factor matrices, $W \in \mathbb{R}_+^{n \times k}$, and $H \in \mathbb{R}_+^{k \times d}$ such that, $X = WH$ [116] where k is identified via Algorithm 1. Symmetric NMF is a special case of NMF having completely positive and identical non-negative factor matrices [117]. In symmetric NMF [118], we solve the following equation for the cluster

membership indicator matrix $W \in \mathbb{R}_{+}^{n \times k}$,

$$min_{W \geq 0}\ f(W) = ||S - WW^T||_F^2 \tag{7.1}$$

where similarity matrix $S \in \mathbb{R}_{+}^{n \times n}$ (S is symmetric and hence, $S = S^T$), and $||.||_F$ indicates the *Frobenius norm*-based minimization. Typically, $k << n$. To solve the optimization problem in Eq. (7.1) for W, we apply alternating non-negative least squares (ANLS) optimization (with block principal pivoting) that converges to stationary points [119]. To initialize the factor W, we apply non-negative double singular value decomposition (NNDSVD) [120] which is based on approximations of the positive sections of the partial SVD factors of the similarity matrix so that symmetric NMF optimization attains better convergence. The largest entry in each row of the W matrix indicates the clustering assignments [121].

7.1.4 Determining the Best Structure-Group

After determining the group composition using the matrix W as described above, we then identify the best group as follows. Each group of structures is associated a score that is computed as the average over the potential energies of structures in the group. The groups are then ranked, and the one with the lowest score is selected as the best group.

7.1.5 Determining the Best Structure in a Group

Once the best group of structures is determined, the structures in the group are evaluated so as to determine a best structure. We make use of the strategy recently introduced in [28], which employs a structure density score [109]. Let a structure x_i belong to a group comprised of l structures. The density score ds_i of structure x_i is given by $ds_i = \frac{\sum_{j=1}^{l} r_{ij}}{l}$; where r_{ij} denotes the pairwise root-mean-squared-deviation (RMSD) between structure x_i and structure x_j ($1 \leq i, j \leq l$). The structure density scores are normalized to be in the

range -1 and 1. The normalized density score $ds_i^{'}$ is given by

$$ds_i^{'} = \begin{cases} \frac{(ds_i - ds_{median})}{ds_{median} - ds_{min}} & \text{if } ds_i < ds_{median} \\ 0 & \text{if } ds_i = ds_{median} \\ \frac{(ds_i - ds_{median})}{ds_{max} - ds_{median}} & \text{if } ds_i > ds_{median} \end{cases} \tag{7.2}$$

where ds_{min}, ds_{max}, and ds_{median} denote the minimum, maximum, and median density scores respectively. Using these normalized scores, we then assign weight w_i to each structure as in: $w_i = e^{-ds_i^{'}}$. Once the structures in a group are weighted in this manner, the maximum-weight structure is then selected as the best structure.

7.2 Evaluation Dataset

SNMF-DS is evaluated on two datasets. The first, shown in Table 7.1, contains 18 benchmark proteins of different folds and lengths (number of amino acids). The second dataset, shown in Table 7.2, contains 10 targets selected from the free modeling category in CASP12 and CASP13; the list includes several hard targets [111, 112]. For each protein target, we use the Rosetta *AbInitio* protocol to generate 12,000 structures. Tables 7.1-7.2 provide additional details for each structure dataset. For instance, Table 7.1 shows the entry id of an experimentally-known structure (ground truth) for each target in the Protein Data Bank (PDB). The fold of the experimentally-known structure, and the number of amino acids in the corresponding target are shown, as well. The minimum RMSD to the experimentally-known structure in a structure dataset is shown in Column 6. This value is utilized to estimate the difficulty of a structure dataset for structure selection. Targets where this value does not exceed 1Å are considered easy; those where this value does not exceed 3Å are considered medium; the rest are considered hard.

Table 7.1: Benchmark dataset (* denotes proteins with a predominant β fold and a short helix). The chain extracted from a multi-chain PDB entry is shown in parentheses. PDB ID, Fold, Length, and minimum RMSD over structure dataset to corresponding experimentally-known structure are shown for each target.

Difficulty	**#**	**PDB ID**	**Fold**	**Length**	**Min RMSD (Å)**
Easy	1	1ail	α	70	0.573
	2	1dtd(B)	$\alpha+\beta$	61	0.565
	3	1wap(A)	β	68	0.568
	4	1tig	$\alpha+\beta$	88	0.623
	5	1dtj(A)	$\alpha+\beta$	74	0.701
	6	1hz6(A)	$\alpha+\beta$	64	0.827
Medium	7	1c8c(A)	β^*	64	1.331
	8	2ci2	$\alpha+\beta$	65	1.581
	9	1bq9	β	53	1.308
	10	1hhp	β^*	99	1.761
	11	1fwp	$\alpha+\beta$	69	1.568
	12	1sap	β	66	2.031
	13	2h5n(D)	α	123	2.053
Hard	14	2ezk	α	93	3.475
	15	1aoy	α	78	3.496
	16	1aly	β	146	9.179
	17	1cc5	α	83	4.654
	18	1isu(A)	*coil*	62	5.912

Moreover, Table 7.2 lists similar information for the CASP targets. We note that in two cases, marked by asterisks, the experimentally-known structure has not been deposited yet in the PDB and is only available on the CASP website [110]. The minimum RMSDs shown in Column 5 indicate the level of difficulty Rosetta experiences on the CASP targets and convey the variability of the quality of structure datasets over which a structure selection method has to perform in general.

Table 7.2: CASP dataset. CASP target IDs are shown in Column 2. PDB ID, Length, and Minimum RMSD over structure dataset to corresponding experimentally-known structure are shown for each target. Experimentally-known structures only available in the CASP website [110] are marked by asterisks.

#	**Target ID**	**PDB ID**	**Length**	**Min RMSD (Å)**
1	T1008-D1	6msp	77	1.542
2	T0886-D1	5fhy	69	5.102
3	T0953s1-D1	6f45	67	6.344
4	T0960-D2	6cl5	84	6.402
5	T0898-D2	**	55	6.598
6	T0892-D2	5nv4	110	6.950
7	T0953s2-D3	6f45	77	7.607
8	T0957s1-D1	6cp8	108	7.677
9	T0897-D1	**	138	9.638
10	T0859-D1	5jzr	113	10.268

7.3 Evaluation Results

We compare SNMF-DS with three representative, state-of-the-art methods, (1) Feature-based NMF method [122] that outperforms the basin-based [27] methods (which outperform community-based graph-clustering methods [28, 39]), (2) SBROD [123], an energy-based method, and (3) MUFOLD-CL [124], a clustering-based method. The comparative evaluation is carried out on two datasets via rigorous metrics.

7.3.1 Running Time Comparison

We compare SNMF-DS, NMF-MAD [29], MUFOLD-CL, and SBROD on five selected structure sets (two from the CASP targets and three from the benchmark targets) that are representative of our observations with regards to running time. Table 7.3 indicates that MUFOLD-CL is the fastest, followed by SNMF-DS, and SBROD. NMF-MAD is the most expensive due to its costly feature selection and extraction steps.

Table 7.3: Running Time Comparison

ID	SNMF-DS	MUFOLD-CL	SBROD	NMF-MAD
T0898-D2	16m 46s	1m 17s	33m 55s	2h 41m 53s
T0897-D1	39m 9s	3m 14s	1h 25m 37s	5h 7m 12s
1ail	17m 57s	55s	43m 39s	3h 9m 15s
1tig	18m 26s	1m 6s	52m 17s	3h 48m 8s
1aly	25m 26s	2m 33s	1h 36m 15s	6h 22m 43s

7.3.2 Group Purity Comparison

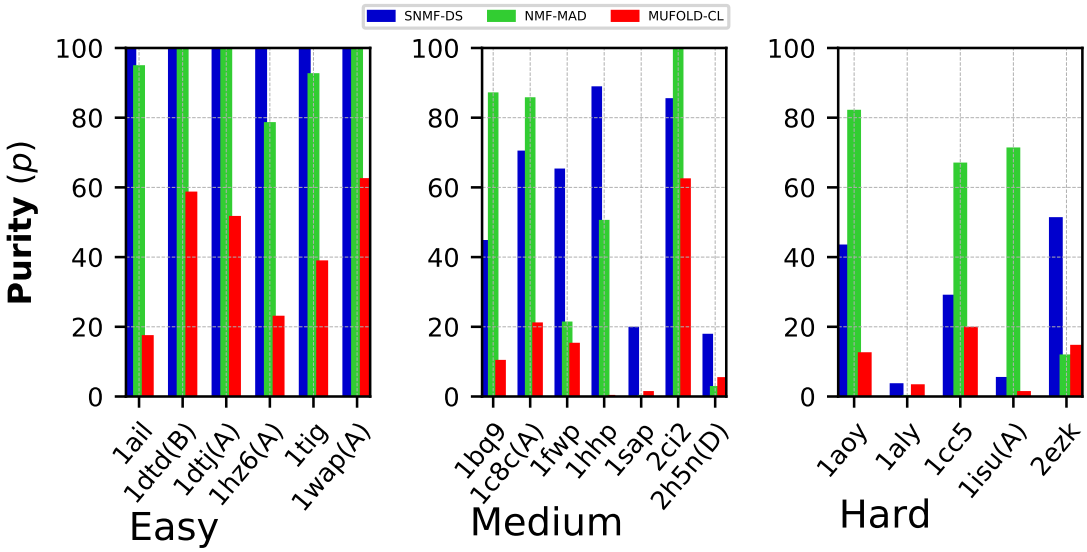


Figure 7.4: The purity of the group/cluster selected by SNMF-DS, NMF-MAD, and MUFOLD-CL are shown over the benchmark targets.

We compare the purity of the group/cluster selected by SNMF-DS, NMF-MAD, and

MUFOLD-CL. It is worth noting that SBROD ranks structures by energies and so does not organize them into groups. Figure 7.4 compares purities over the benchmark targets, whereas Figure 7.5 does so over the CASP targets.

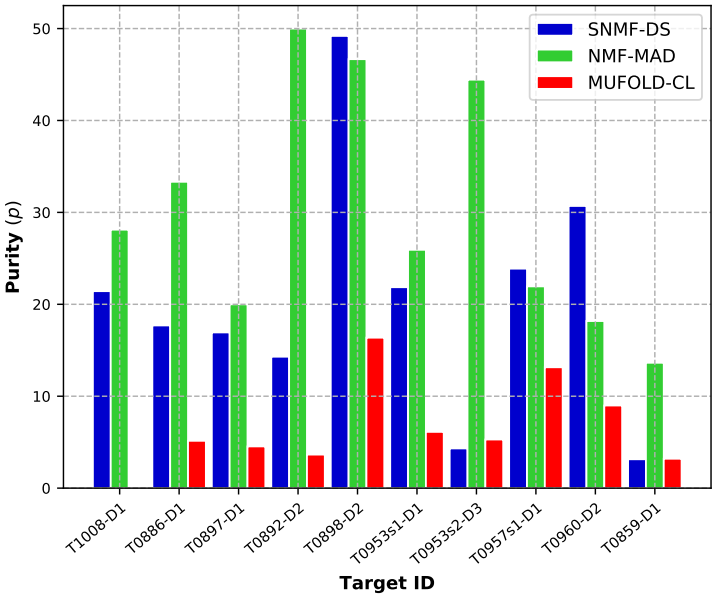


Figure 7.5: The purity of the group/cluster selected by SNMF-DS, NMF-MAD, and MUFOLD-CL are shown over the CASP targets.

Figures 7.4-7.5 show that SNMF-DS and NMF-MAD largely outperform MUFOLD-CL. Specifically, for the easy benchmark targets, the purity values obtained by MUFOLD-CL range from 17% to 62%, whereas NMF-MAD attains 78% to 100% purity, and SNMF-DS dominates with 100% purity in each target. For the medium benchmark targets, SNMF-DS achieves better purity than NMF-MAD on 4/7 cases; MUFOLD-CL is inferior to SNMF-DS on all the medium benchmark targets (and with only two marginal wins over NMF-MAD). On the hard benchmark targets, NMF-MAD does particularly well, reaching purity values from 11% to 81% (except for `1aly`); for SNMF-DS, purity values over these targets range

from 3% to 51%. MUFOLD-CL does not perform better than SNMF-DS on any target; it only beats NMF-MAD on one target (`2ezk`). These observations are further confirmed over the CASP dataset. On the 10 CASP targets, MUFOLD-CL reaches purity values ranging from 3% to 16% (on T1008-D1, purity is 0%) and is inferior to both NMF-MAD and SNMF-DS; it performs as well or slightly better than SNMF-DS on only 2/10 targets. Specifically, in 7/10 targets, NMF-MAD outperforms SNMF-DS with purity values ranging from 13% to 50%; in the remaining 3 targets, SNMF-DS performs better than NMF-MAD with purity values ranging from 3% to 49%. Altogether, these results demonstrate that SNMF-DS is as competitive as NMF-MAD in terms of the quality of the selected group.

7.3.3 Loss Comparison

Table 7.4: SNMF-DS, MUFOLD-CL, SBROD, and NMF-MAD are compared in terms of RMSD, TM-Score, and GDT-TS loss on the benchmark targets. Lowest loss per PDB ID in any metric (RMSD, TM-Score, or GDT-TS) is highlighted in bold.

PDB ID	RMSD Loss, TM-Score Loss, GDT-TS Loss			
	SNMF-DS	MUFOLD-CL	SBROD	NMF-MAD
`1ail`	**0.5084**, **0.0655**, **0.072**	1.447, 0.1676, 0.1336	2.937, 0.314, 0.3478	0.971, 0.1604, 0.1357
`1dtj(A)`	0.1941, **0.0048**, 0.0296	**0.036**, 0.0198, **0.0066**	0.69, 0.006, 0.0329	0.3345, 0.0782, 0.1081
`1dtd(B)`	0.3528, **0.0042**, **0.0041**	0.49, 0.0052, 0.0043	**0.12**, 0.005, 0.0082	0.5915, 0.0329, 0.0451
`1wap(A)`	0.3425, 0.0288, **0.0166**	**0.263**, **0.0242**, 0.0233	1.242, 0.1107, 0.1	0.6219, 0.0531, 0.04
`1tig`	**0.0717**, **0.003**, **0.0053**	0.749, 0.004, 0.008	0.709, 0.0134, 0.016	0.6569, 0.0469, 0.0483
`1hz6(A)`	**0.0936**, **0.002**, **0.0034**	0.405, 0.0037, 0.0036	0.191, 0.0145, 0.0382	0.809, 0.0415, 0.0352
`1bq9`	**1.1992**, 0.1677, 0.1389	2.02, 0.2115, 0.1759	1.337, 0.1331, 0.1065	1.3089, **0.1167**, **0.0755**
`1c8c(A)`	**0.7991**, 0.1092, 0.086	1.012, 0.135, 0.1016	1.531, 0.1465, 0.1328	1.092, **0.0596**, **0.0429**
`1fwp`	**0.5085**, **0.0034**, 0.0036	0.724, 0.0074, **0.0018**	1.039, 0.0589, 0.1332	0.5319, 0.0471, 0.0616
`1hhp`	**2.1971**, 0.0601, 0.0707	10.919, 0.6326, 0.6161	2.76, **0.0533**, **0.0606**	2.6835, 0.2939, 0.2828
`1sap`	**0.5592**, **0.074**, **0.0417**	1.61, 0.0831, 0.0492	1.873, 0.141, 0.1136	2.075, 0.0989, 0.125
`2ci2`	**0.3118**, **0.007**, **0.006**	3.202, 0.1155, 0.1114	3.083, 0.1334, 0.1175	1.7897, 0.3246, 0.3462
`2h5n(D)`	3.7028, 0.3178, 0.3215	7.806, 0.2479, 0.2576	3.883, 0.0856, 0.094	**3.3498**, **0.0805**, **0.0732**
`1aoy`	2.7896, **0.1136**, **0.093**	5.246, 0.1856, 0.1635	**2.047**, 0.1286, 0.1218	2.9788, 0.2918, 0.2788
`1aly`	5.7842, 0.0167, 0.0368	**3.467**, **0.0155**, **0.024**	4.373, 0.029, 0.0325	7.9939, 0.1411, 0.1635
`1cc5`	**0.4732**, **0.048**, 0.0452	1.159, 0.0831, **0.0392**	1.949, 0.0501, 0.0551	2.1843, 0.0565, 0.0573
`1isu(A)`	2.9928, 0.2182, 0.2299	6.357, 0.2106, 0.242	5.32, 0.1603, 0.2137	**2.5552**, **0.081**, **0.0887**
`2ezk`	2.9154, 0.0188, 0.0177	**1.172**, **0.003**, **0.0076**	3.142, 0.0178, 0.0244	3.5136, 0.0229, 0.0296

We compare SNMF-DS, NMF-MAD, MUFOLD-CL, and SBROD in terms of RMSD loss, TM-Score loss, and GDT-TS loss of the selected structure. This comparison is in

Table 7.4 for the benchmark targets and in Table 7.5 for the CASP targets.

Table 7.5: SNMF-DS, MUFOLD-CL, SBROD, and NMF-MAD are compared in terms of RMSD, TM-Score, and GDT-TS loss on the CASP targets. Lowest loss per target in any metric (RMSD, TM-Score, or GDT-TS) is highlighted in bold.

Target ID	RMSD Loss, TM-Score Loss, GDT-TS Loss			
	SNMF-DS	MUFOLD-CL	SBROD	NMF-MAD
T1008-D1	**0.3656, 0.007, 0.0011**	3.305, 0.0137, 0.065	0.398, 0.0086, 0.0032	1.0238, 0.0156, 0.0162
T0886-D1	3.6714, **0.03**, 0.0362	4.94, 0.0403, 0.0435	**2.12**, 0.034, 0.0326	2.5984, 0.0331, **0.029**
T0953s1-D1	2.9398, **0.02**, 0.0112	2.947, 0.055, 0.0187	3.032, 0.084, **0.0037**	**2.613**, 0.0225, 0.0223
T0960-D2	1.8595, 0.0307, 0.0268	**0.53**, 0.0384, 0.0328	0.67, 0.0505, 0.0417	2.6181, **0.0182**, **0.0178**
T0898-D2	1.4889, 0.003, **0.0071**	0.468, 0.008, 0.0091	**0.162**, **0.001**, 0.0137	2.3824, 0.0108, 0.0181
T0892-D2	**0.9038**, **0.0119**, **0.004**	1.787, 0.0129, 0.0069	1.51, 0.0134, 0.0114	2.8416, 0.0242, 0.009
T0953s2-D3	1.4223, **0.01**, 0.011	2.137, 0.0187, 0.0162	**0.326**, 0.0109, **0.0033**	1.8621, 0.0256, 0.0153
T0897-D1	3.471, 0.0263, 0.0108	1.137, 0.0064, 0.018	**0.236**, **0.0032**, **0.0055**	2.9413, 0.0158, 0.009
T0957s1-D1	1.18, **0.0027**, 0.0047	0.709, 0.008, 0.0023	**0.423**, 0.0079, **0.001**	1.6803, 0.018, 0.0076
T0859-D1	2.3755, 0.056, 0.045	**0.421**, 0.0094, **0.0023**	0.518, **0.0088**, 0.0044	3.5967, 0.0329, 0.0132

Tables 7.4-7.5 make clear the superiority of SNMF-DS over the other methods. For instance, Table 7.4 shows that the RMSD loss incurred by SNMF-DS is below $1\mathring{A}$ for 11/18 of the benchmark targets. Table 7.4 also shows that for 14/18 of these targets, the best structure selected by SNMF-DS incurs the minimum loss compared to the other methods in terms of at least one of the three quantities (RMSD loss, TM-Score loss, and GDT-TS loss). Table 7.5 shows that the RMSD loss incurred by SNMF-DS is below $2\mathring{A}$ for 6/10 of the CASP targets. For 7/10 CASP targets, the best structure selected by SNMF-DS incurs the minimum loss compared to the other methods in terms of at least one of the three measures (RMSD loss, TM-Score loss, and GDT-TS loss).

7.3.4 Statistical Significance Analysis

Finally, we conduct a statistical significance analysis on both purity and RMSD loss results combined over the benchmark and CASP. We report the results of Friedman statistical tests with Hommel's post-hoc analysis [125]. We note that Friedman's test is ideal for conducting statistical significance of multiple methods contending over multiple test cases.

The test is non-parametric and evaluates the null hypothesis. The null hypothesis states that there is negligible difference between the contending methods. Once the null hypothesis is rejected, we conduct Hommel's post-hoc analysis to justify the performance of SNMF-DS with respect to the other methods. The statistical tests are performed on all the 28 targets at $\alpha = 0.05$. The results are shown in Table 7.6. The lowest average rank reported in Columns 2 and 5 for RMSD loss and purity, respectively, indicates the best method (marked with an star); we note that SNMF-DS is reported as the best on either RMSD loss or purity. We note that a method is said to be significantly different from the best one, when the p-value of the corresponding method is less than that of the p-Hommel at $\alpha = 0.05$ (values are indicated in bold). Therefore, Table 7.6 clearly shows that SNMF-DS is the best method irrespective of the performance measure, purity or RMSD loss. In fact, for RMSD loss, SNMF-DS significantly outperforms all the other methods; for purity, SNMF-DS outperforms MUFOLD-CL, while there is insignificant difference with NMF-MAD.

Table 7.6: Statistical significance of different methods over all 28 targets (benchmark and CASP) determined through Friedman's tests with Hommel's post-hoc analysis at $\alpha = 0.05$. The best method is marked with an star ($\star$), while boldface indicates that the corresponding method is significantly different in comparison to the best method. Note that SBROD does not produce a group, hence no purity analysis can be provided.

Method	RMSD Loss			Purity		
	Avg. Rank	p value	p Hommel	Avg. Rank	p value	p Hommel
NMF-MAD	**2.929**	**0.002**	**0.006**	1.679	0.593	0.593
MUFOLD-CL	**2.679**	**0.017**	**0.034**	**2.786**	**2.91E-6**	**5.8E-6**
SBROD	**2.536**	**0.049**	**0.05**			
SNMF-DS*	1.857	-	-	1.536	-	-

7.4 Summary

The work described in this chapter shows that SNMF-DS is a powerful method that outperforms state-of-the-art methods. The presented results are encouraging, as exploiting non-negative matrix factorization is a relatively new thread of research for molecular structure modeling and further support extending matrix factorization to tensor factorization by utilizing a collection of proximity measures (rather than a single one as in SNMF-DS). In chapter 8, we describe a hybrid framework capable of not only organizing structures into groups but also scoring individual structures, thereby additionally addressing the EMA problem we laid out in chapter 2.

Chapter 8: Tensor Decomposition-based Method to Organize, Score, and Select Structures

In this chapter, we suggest a shift from matrix-factorization to tensor factorization-based method that can support further utility alongside the organization of tertiary structures of a protein into groups. In fact, the method doubles as a non-parametric clustering technique and so can broadly support various application settings. We investigate its efficacy on the hallmark EMA problem. The framework falls in the category of multi-structure methods, as it can extract information from multiple structures of a given protein. Furthermore, as we want to demonstrate the method as a complete EMA framework, it should additionally allow us to obtain an individual score/distance for each structure that can serve as a proxy of the quality/accuracy of a structure in the set it belongs to. By analyzing such scores and the structure with the best score over many protein targets, we can evaluate the described method with state-of-the-art methods, including single-structure methods that are currently considered the best-performing ones in EMA. The work described here has been disseminated in [126].

8.1 Non-negative Tensor Factorization (NTF)-based Framework to Organize Protein Structure Space

We refer to the method as NTF-REL (non-negative tensor factorization with RESCAL) which can proceed in four stages. Stage I organizes given structures into groups $\{G_i\}$ via tensor factorization. Stage II utilizes energies to rank the groups. Stage III partitions each group into subgroups and ranks them. Stage IV utilizes all this information to compute a score for each structure. A schematic is related in Fig. 8.1.

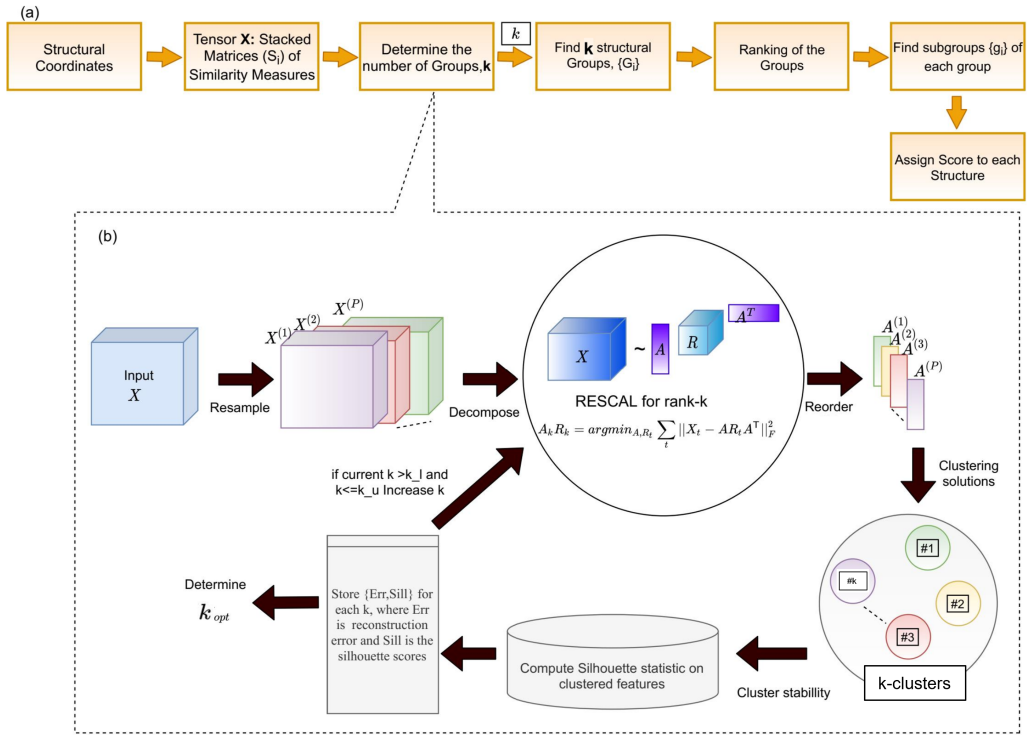


Figure 8.1: (a) Schematic of NTF-REL, (b) Finding the number of latent features with non-negative RESCAL factorization.

8.1.1 Stage I: From Structures to Groups

Different metrics for comparing two structures can capture different aspects and often provide complementary information [127]. This concept motivates us to go from matrix to tensor. In fact, we can form a tensor X by stacking (symmetric) similarity matrices $S^i_{n,n}$ obtained on n structures, where i refers to a particular metric. Entry (a, b) in S^i measures the similarity according to metric i between two structures at positions a, b in a list of n structures; $S^i_{a,a} = 1$. We utilize 5 popular domain-specific metrics, RMSD, TM-score, GDT-TS, GDT-HA, and MaxSub Score. Since RMSD is a dissimilarity measure, we turn it into a similarity one as in $S_{a,b} = \frac{1}{\text{RMSD(a,b)}}$. Now, Each S^i is a slice of the tensor. Each structure

can be stripped down to its main-chain carbon atoms (CA atoms) thereby reducing the cost of computing the tensor.

As Fig. 8.1 represents the schematics of the framework, we employ the RESCAL tensor factorization approach [128] integrated with an automatic latent dimension determination method [129]. RESCAL was initially intended for extracting latent communities in relational data coming from dynamic networks. However, there exists recently proposed tensor-based frameworks that support both stationary and non-stationary systems [130]. Specifically, RESCAL factorizes tensors formed by a set of m stacked matrices of graphs (each graph has n nodes), $X^{n\times n\times m}$ into a factor matrix $A^{n\times k}$ and a core tensor $R^{k\times k\times m}$, where k is the latent dimension (or number of the latent communities/groups). The factorization solves the following optimization problem:

$$\text{argmin}_{A,R} \|X - R \times_1 A \times_2 A\|_F^2 \tag{8.1}$$

where $\times_i$ denotes the mode$-i$ product [131]. The extracted factors are interpretable; each column of A represents a latent community/group of objects, and each slice R_m of the core tensor R captures the relations among the groups at instance m. Considering the non-negativity of the data, we employ non-negative RESCAL [132]. The optimization with non-negativity constraints is given by,

$$\text{argmin}_{A,R_m} \sum_m \left\|X_m - AR_mA^\top\right\|_F^2 \quad \textit{subject to } \sum_j A_{ij} = 1, \text{ for } 1 \leq j \leq k; A, R \geq 0$$

Fig. 8.1(b) demonstrates the mechanism for the adaptation of RESCAL integrated with an algorithm to find the k latent groups, to which we refer as RESCAL-k [129]. RESCAL-k consists of the following components:

(1) Custom Resampling: We generate an ensemble of tensors from X, $[X^{(q)}]_{q=1,\ldots,P}$; where the mean of these tensors is equal to the original tensor X. Each of these tensors $X^{(q)}$ is built by perturbing each of the elements using random uniform noise, such that $X^{(q)} = X(\odot)\Delta_q$ (further details can be found in [133]).

(2) RESCAL Minimization: We use Frobenius norm-based multiplicative updates [128] to explore various numbers of latent features; k in an interval $[k_{min}, k_{max}]$, for each of the P generated random tensors $X^{(q)}$. The decomposed component A corresponds to the samples in reduced latent dimension $n \times k$ denoting the groups, whereas R is the $k \times k \times m$ relational tensor representing the group interactions.

(3) Custom Clustering: For each $k \in [k_{min}, k_{max}]$, we cluster the set of the $n \times k$ latent components. To extract the latent dimension, we determine the dependency of the stability of the obtained clusters and the improvement of the reconstruction error on the latent dimension k. The final latent components, $\widetilde{A}$, are the medoids of the obtained stable clusters, with $\widetilde{R}$ denoting the corresponding mixing coefficients. The latent group estimation pipeline is based on the pyDNMFk toolbox [134, 135].

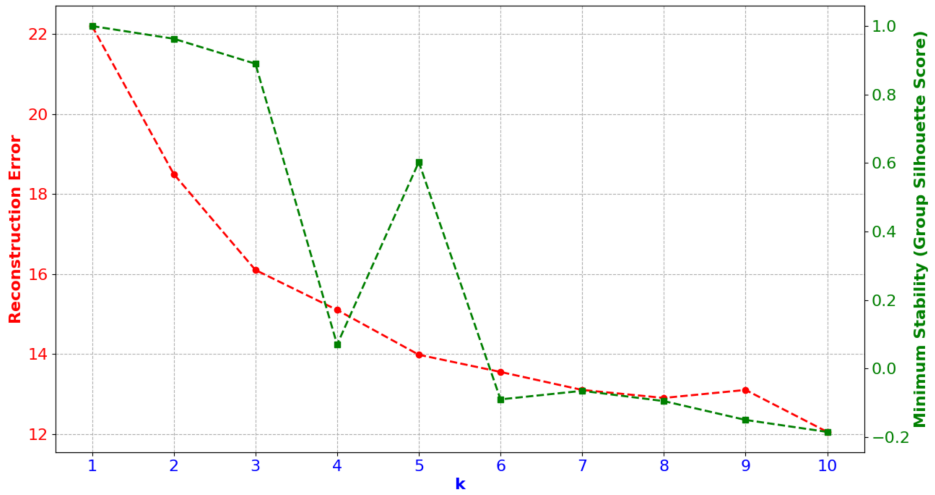


Figure 8.2: The stability analysis for one of our protein targets from CASP, `T1008-D1`. Candidate values of k contain considerably larger gaps between the relative reconstruction error and the silhouette statistics with silhouette score $\geq$ 0.6. Out of the candidates (2, 3, and 5), we choose the one with the lowest reconstruction error (i.e., $k_{opt} = 5$ here).

RESCAL-k employs Silhouette statistics [136] to determine the cluster stability for each k. The Silhouette parameter that quantifies the cluster stability is in the range $[-1, 1]$, where -1 corresponds to a bad clustering and 1 to perfect clustering. Fig. 8.2 shows how using both Silhouette and reconstruction norm, we can determine the optimal k. The final representative A corresponding to the RESCAL factorization of X for k_{opt} estimated by RESCAL-k is then used to identify the best composition of the k groups identified, as illustrated in Fig. 8.2.

8.1.2 Stage II: Ranking Groups

After determining the group composition using the matrix A as described above, we can then rank the groups. Each group (of structures) may be associated with the average value over energies of the structures in the group. The groups can then be ranked in ascending order of the group energy score; the group with the lowest score is the best-ranked group. The rank of a group G is denoted by R_G.

8.1.3 Stage III: Partitioning Groups into Subgroups

We hybridize the tensor-based approach above with graph clustering. We utilize work in [27] which embeds structure-energy pairs in a nearest-neighbor graph (using RMSD to identify nearest neighbors), over which it identifies local energy minima representing different energy basins and groups vertices into basins. The methodological details are described in section 5.2.3. Our adaptation here is not to apply this approach over all n structures, but instead to apply it to each group identified via tensor factorization in order to partition each of them into "basins"; to which we refer more generally as subgroups.

8.1.4 Stage IV: Scoring each Structure

To score each structure, we modify the strategy proposed in [28], which employs a model density score [109]. Let a structure x_i belong to a group G comprised of l structures. As in [28], we associate a density score dt_i with x_i as: $dt_i = \frac{\sum_{j=1}^{l} \text{TM-Score}_{i,j}}{l}$, where $\text{TM} - \text{Score}_{i,j}$ is

the TM-Score between x_i and x_j $(1 \leq i, j \leq l)$. In principle, different metrics can be measured; unlike work in [28], which uses RMSD, we use TM-score; its $[0, 1]$ range easily translates via the above formula into density scores in the range $[0, 1]$.

Our density score additionally utilizes information from Stages II-III. Let the number of subgroups in group G be z. The subgroups are first sorted and ranked in descending order of size. Let the rank of each subgroup $g \in G$ be r_g. The last d' $\left(d' = \lceil \frac{z}{3} \rceil\right)$ subgroups are further sorted (in ascending order) by the average potential energy of the structures in a subgroup, resulting in ${r_g}'$.

The modified structure density score $dt_i^{'}$ is then:

$$dt_i^{'} = \begin{cases} \frac{dt_i}{max(R_G)+{r_g}'} & \text{if } x_i \in \texttt{last } d' \texttt{ subgroups} \\ \frac{dt_i}{R_G+r_g} & \texttt{otherwise} \end{cases}$$

Using these modified scores, we then assign weight/score w_i to each structure as in: $w_i = e^{dt_i^{'}}$. Once the structures are weighted in this manner, the highest-weight structure is considered as the *best* structure.

8.2 Experimental Setup and Evaluation Results

We compare NTF-REL to representative SOTA methods: (1) Single-structure methods ProQ2 [137], ProQ3 [138], ProQ3D [139], and ProQ4 [140], and (2) recent NMF-based methods [29, 30] which were shown to outperform MUFOLD-CL and other multi-model methods. We evaluate NTF-REL on the same two datasets (described in section 7.2) that we have employed for our symmetric NMF-based method, SNMF-DS. These NMF-based methods were shown to outperform MUFOLD-CL and other multi-model methods. We present three sets of results: comparison with SOTA methods on target-wise correlation with respect to the true TM-Score, structure loss, and an analysis of statistical significance.

8.2.1 Evaluation Metrics

Since NTF-REL assigns a score to each structure and hence can also select a single structure as the best one, we evaluate its performance as an EMA method, as well as a single-structure selection method. First, we evaluate the quality of the scores assigned to structures by measuring the Pearson correlation between these scores and the true TM-Score from the *ground truth* (the experimentally-known structure for each target). We also measure loss as the difference in quality between the structure selected by a method and the best-quality structure in a dataset, with quality assessed by any of the three following metrics, RMSD, TM-Score, and GDT-TS, with respect to the experimentally-known structure.

8.2.2 Comparative Evaluation on Correlation with TM-Score

Table 8.1: Target-wise Pearson correlation with respect to true TM-Score. Top two values are highlighted in boldface font.

Benchmark Targets					
Target-ID	NTF-REL	ProQ2	ProQ3	ProQ3D	ProQ4
1ail	**0.7821**	0.683	0.699	0.743	**0.787**
1dtj(A)	**0.802**	0.701	0.707	0.762	**0.807**
1dtd(B)	**0.7713**	0.675	0.683	0.733	**0.776**
1wap(A)	**0.7432**	0.658	0.665	0.713	**0.747**
1tig	**0.715**	0.624	0.634	0.689	**0.719**
1hz6(A)	**0.741**	0.647	0.657	0.714	**0.745**
1bq9	**0.688**	0.616	0.607	0.655	**0.697**
1c8c(A)	**0.728**	0.636	0.643	0.693	**0.733**
1fwp	**0.733**	0.641	0.646	0.702	**0.728**
1hhp	**0.679**	0.602	0.605	0.645	**0.683**
1sap	**0.7066**	0.617	0.621	0.673	**0.711**
2ci2	**0.746**	0.655	0.661	0.709	**0.741**
2h5n(D)	**0.7204**	0.623	0.637	0.685	**0.724**
1aoy	**0.686**	0.599	0.604	0.652	**0.677**
1aly	**0.652**	0.596	0.592	0.639	**0.661**
1cc5	**0.709**	0.627	0.625	0.684	**0.714**
1isu(A)	**0.6938**	0.607	0.611	0.679	**0.698**
2ezk	**0.667**	0.602	0.607	0.653	**0.675**
CASP Targets					
Target-ID	NTF-REL	ProQ2	ProQ3	ProQ3D	ProQ4
T0859-D1	**0.7031**	0.619	0.642	0.689	**0.717**
T0886-D1	**0.6972**	0.624	0.634	0.684	**0.704**
T0892-D2	**0.7044**	0.643	0.638	0.691	**0.719**
T0897-D1	**0.6896**	0.637	0.628	0.676	**0.703**
T0898-D2	**0.7203**	0.638	0.656	0.707	**0.734**
T0953s1-D1	**0.701**	0.632	0.603	0.652	**0.708**
T0953s2-D3	**0.718**	0.627	0.617	0.678	**0.725**
T0957s1-D1	**0.738**	0.602	0.635	0.696	**0.745**
T0960-D2	**0.7161**	0.624	0.646	0.667	**0.731**
T1008-D1	**0.7533**	0.643	0.648	0.701	**0.761**

Table 8.1 relates the comparison on the benchmark and CASP targets. The top two predictions on each target are highlighted in boldface font. Table 8.1 shows that NTF-REL and ProQ4 outperform ProQ2, ProQ3, and ProQ3D on all benchmark and CASP targets. NTF-REL performs comparably to ProQ4, with differences often observed in the third digit after the decimal sign. In particular, both NTF-REL and ProQ4 achieve a Pearson correlation higher than 0.7 on 12/18 of the benchmark targets, and 8/10 and 10/10 of the CASP targets, respectively (with strictly no rounding). In many targets, both methods achieve or come very close to a Pearson correlation of 0.8.

8.2.3 Loss-based Comparison

Table 8.2: NTF-REL, SNMF-DS, ProQ3D, ProQ4, and NMF-MAD are compared on RMSD, TM-Score, and GDT-TS loss. Lowest loss per metric per target is highlighted in boldface font.

ID	**Benchmark Targets** RMSD Loss, TM-Score Loss, GDT-TS Loss				
	SNMF-DS	ProQ3D	ProQ4	NMF-MAD	NTF-REL
1ail	0.5084, 0.0655, 0.072	0.357, 0.042, 0.034	0.486, 0.063, 0.057	0.971, 0.1604, 0.1357	**0.1527**, **0.03**, **0.012**
1dtj(A)	0.1941, 0.0048, 0.0296	**0.125**, 0.0118, **0.0057**	0.21, 0.0179, 0.0089	0.3345, 0.0782, 0.1081	0.166, **0.0043**, 0.026
1dtd(B)	0.3528, 0.0042, 0.0041	0.245, 0.0026, 0.0022	0.75, 0.0061, 0.0091	0.5915, 0.0329, 0.0451	**0.117, 0.0015, 0.0016**
1wap(A)	0.3425, 0.0288, 0.0166	0.352, 0.0277, 0.0245	0.423, 0.0311, 0.029	0.6219, 0.0531, 0.04	**0.2285, 0.021, 0.0123**
1tig	**0.0717**, **0.003**, **0.0053**	0.496, 0.0035, 0.0065	0.479, 0.0032, 0.0061	0.6569, 0.0469, 0.0483	0.72, 0.13, 0.091
1hz6(A)	**0.0936**, **0.002**, **0.0034**	0.397, 0.0031, 0.0042	0.291, 0.0025, 0.0039	0.809, 0.0415, 0.0352	0.1248, 0.005, 0.006
1bq9	1.1992, 0.1677, 0.1389	0.745, 0.112, 0.0896	**0.673, 0.103, 0.0749**	1.3089, 0.1167, 0.0755	1.6362, 0.226, 0.1875
1c8c(A)	0.7991, 0.1092, 0.086	0.521, 0.0953, 0.077	**0.444**, 0.0887, 0.069	1.092, **0.0596**, **0.0429**	0.7991, 0.1092, 0.086
1fwp	0.5085, 0.0034, 0.0036	0.473, 0.0048, **0.0013**	0.491, 0.0059, 0.0019	0.5319, 0.0471, 0.0616	**0.2658, 0.0019**, 0.0023
1hhp	2.1971, 0.0601, 0.0707	0.928, 0.073, 0.069	**0.77**, **0.0467**, **0.0503**	2.6835, 0.2939, 0.2828	2.3188, 0.0634, 0.075
1sap	0.5592, 0.074, 0.0417	0.719, 0.0637, 0.0398	0.875, 0.0714, 0.0416	2.075, 0.0989, 0.125	**0.3229, 0.045, 0.0248**
2ci2	0.3118, 0.007, 0.006	**0.213, 0.0056, 0.0042**	0.831, 0.013, 0.015	1.7897, 0.3246, 0.3462	0.3656, 0.01, 0.008
2h5n(D)	3.7028, 0.3178, 0.3215	0.917, 0.0475, 0.0276	**0.839, 0.0465, 0.0315**	3.3498, 0.0805, 0.0732	2.987, 0.267, 0.2708
1aoy	2.7896, 0.1136, 0.093	1.265, 0.0567, **0.0428**	**1.074**, **0.0511**, 0.0431	2.9788, 0.2918, 0.2788	2.346, 0.107, 0.089
1aly	5.7842, 0.0167, 0.0368	**2.674**, 0.0162, **0.027**	2.733, 0.0193, 0.0325	7.9939, 0.1411, 0.1635	2.912, **0.015**, 0.0345
1cc5	**0.4732**, **0.048**, 0.0452	1.161, 0.0791, **0.0388**	1.045, 0.0602, 0.0441	2.1843, 0.0565, 0.0573	0.539, 0.054, 0.0509
1isu(A)	2.9928, 0.2182, 0.2299	1.036, **0.072, 0.0705**	1.124, 0.0733, 0.0717	2.5552, 0.081, 0.0887	**0.8689**, 0.112, 0.135
2ezk	2.9154, 0.0188, 0.0177	0.729, 0.0027, 0.0063	**0.625, 0.0019, 0.0042**	3.5136, 0.0229, 0.0296	2.986, 0.021, 0.023

Target ID	**CASP Targets** RMSD Loss, TM-Score Loss, GDT-TS Loss				
	SNMF-DS	ProQ3D	ProQ4	NMF-MAD	NTF-REL
T1008-D1	0.3656, 0.007, **0.0011**	0.2838, 0.04, 0.035	0.326, 0.091, 0.088	1.0238, 0.0156, 0.0162	**0.2717**, **0.006**, 0.005
T0886-D1	3.6714, **0.03**, 0.0362	0.983, 0.12, 0.112	**1.147**, 0.172, 0.153	2.5984, 0.0331, **0.029**	2.9813, 0.038, 0.034
T0953s1-D1	2.9398, **0.02**, **0.0112**	**0.564**, 0.053, 0.041	1.179, 0.022, 0.019	2.613, 0.0225, 0.0223	3.3869, 0.0293, 0.0289
T0960-D2	1.8595, 0.0307, 0.0268	0.765, 0.13, 0.125	**0.634**, 0.067, 0.081	2.6181, **0.0182**, **0.0178**	1.519, 0.031, 0.033
T0898-D2	1.4889, **0.003**, 0.0071	1.186, 0.019, 0.078	**0.917**, 0.0159, 0.068	2.3824, 0.0108, 0.0181	1.0799, 0.004, **0.0053**
T0892-D2	**0.9038**, **0.0119**, 0.004	1.257, 0.189, 0.076	1.391, 0.263, 0.097	2.8416, 0.0242, 0.009	1.5471, 0.021, **0.0038**
T0953s2-D3	1.4223, **0.01**, 0.011	0.954, 0.161, 0.136	**0.818**, 0.0685, 0.078	1.8621, 0.0256, 0.0153	1.3667, 0.0218, **0.0108**
T0897-D1	3.471, 0.0263, 0.0108	0.973, 0.033, 0.013	**0.849**, 0.029, 0.011	2.9413, **0.0158, 0.009**	3.1845, 0.025, 0.0102
T0957s1-D1	1.18, 0.0027, 0.0047	1.161, 0.031, 0.096	1.294, 0.078, 0.173	1.6803, 0.018, 0.0076	**0.7426, 0.002, 0.0045**
T0859-D1	2.3755, 0.056, 0.045	**1.925**, 0.0752, 0.0853	1.972, 0.0734, 0.0771	3.5967, 0.0329, 0.0132	2.3317, **0.0265, 0.0124**

The above analysis suggests that ProQ3D and ProQ4 decidedly outperform ProQ2 and ProQ3, confirming findings reported in [111]. Therefore, we narrow further comparisons to ProQ3D and ProQ4. Since NTF-REL is a decomposition-based methods, like SNMF-DS and NMF-MAD, we add the latter two to the comparative evaluation on loss. As described earlier, we compute RMSD, TM-Score, and GDT-TS loss and relate these results in Table 8.2 on both benchmark and CASP targets.

Table 8.2 shows the superiority of NTF-REL over other methods. For instance, on the benchmark targets, the RMSD loss incurred by NTF-REL is below $1\mathring{A}$ for 12/18 of the benchmark targets and under $2\mathring{A}$ for 6/10 of the CASP targets. The structure selected by NTF-REL has the minimum loss compared to the structure selected by other methods in terms of at least one of the three metrics (RMSD, TM-Score, and GDT-TS) on 8/18 of the benchmark targets and 6/10 of the CASP targets.

8.2.4 Statistical Significance Analysis

Table 8.3: Statistical significance of various methods over all 28 targets (benchmark and CASP) determined through Friedman's tests with Hommel's post-hoc analysis at $\alpha = 0.05$. The best rank on either TM-Score or GDT-TS loss is highlighted in boldface.

Method	TM-Score Loss			GDT-TS Loss		
	Avg. Rank	p value	p Hommel	Avg. Rank	p value	p Hommel
`NMF-MAD`	3.107	0.063	0.0167	3.607	0.0425	0.0125
`SNMF-DS`	3.249	0.0562	0.0125	2.911	0.7037	0.025
`ProQ3D`	2.923	0.151	0.025	2.768	0.9663	0.05
`ProQ4`	2.893	0.177	0.05	2.965	0.6121	0.0167
`NTF-REL`	**2.322**	–	–	**2.75**	–	–

We carry out a statistical significance analysis on both TM-Score loss and GDT-TS loss

combined over the benchmark and CASP datasets. We report the results of Friedman's statistical tests with Hommel's post-hoc analysis [125]. We note that Friedman's test is ideal for conducting statistical significance of multiple methods contending over multiple test cases. The test is non-parametric and evaluates the null hypothesis (The null hypothesis states that there is negligible difference between the contending methods). Then, we conduct Hommel's post-hoc analysis to fully evaluate the performance of NTF-REL in comparison to other methods. The statistical tests are performed on all the 28 (benchmark and CASP) targets at $\alpha = 0.05$. The results are related in Table 8.3. The lowest average rank are reported in Columns 2 and 5 for TM-Score loss and GDT-Score loss, respectively. The best rank is achieved by NTF-REL on either TM-Score loss or GDT-TS loss. These results conclusively demonstrate the superiority of NTF-REL.

8.3 Summary

The work described in this chapter presents a complete EMA framework that leverages a novel, tensor factorization-based method. The framework associates a score with an individual structure, so it has attributes of single-model EMA method. In addition, the method organizes structures into groups, so it has attributes of a multi-model method. The hybrid framework is shown to outperform various SOTA methods, including distance-based methods currently considered to be the most accurate. Furthermore, this tensor factorization-based method doubles as a non-parametric clustering method and so can support various structure-function studies requiring identification of structural macrostates.

Chapter 9: Summarizing Structural Dynamics of Antibody-Antigen bound System

This chapter demonstrates the ability of some of the algorithms described in this dissertation in handling systems that go beyond a single molecule. In particular, we focus on organizing structures of antibody-antigen bound molecular system obtained via MD simulation to generalize our approach summarized in Chapter 5. The work described here has been disseminated in [141].

9.1 Application Setting and Objective

The specificity and affinity of antibody-antigen recognition are mainly decided by the variable domains and, in particular, the complementarity-determining regions (CDRs). The recognition process involves structure transitions mediated by the antibody's inherent flexibility [142, 143]. Markov state models (MSMs) can be utilized to organize the structure space and to summarize the structural dynamics. As MD simulation is utilized to navigate the energy landscape of the free antibody and the energy landscape of the antigen-antibody complex, we can employ the approach (described in Chapter 5) for this complex system. A computational method that takes into account both structure geometry and energetics can detect energy basins and make the connection between basins and macrostates, the method organizes collected structures into macrostates. That means, this method handles the state space discretization. Then, an MSM construction tool can be utilized to reveal the inter-conversions between the so-identified macro-states and compute the precise state-to-state transition probabilities that provide a quantitative view of the structure dynamics, allowing a more detailed understanding of the impact of the bound antigen on the structural dynamics of the antibody.

9.2 Organizing Structures of Antibody-antigen bound Molecular System

We first provide an overview of the computational pipeline and then provide further details on the MD trajectories employed to obtain conformations of the free and antigen-bound antibody, as well as on the various settings employed to represent and prepare such conformations for analysis.

9.2.1 From MD Trajectories to an MSM of Dynamics

At its core, the pipeline takes in MD trajectories and returns an MSM transition probability matrix, where each entry specifies the probability of transition between a pair of identified macrostates. The first objective is to identify such macrostates. Once such states are identified, the second objective is to use the temporal information available in the MD trajectories to extract information on the accessibility of macrostates in terms of state-to-state probabilities of transition. The result is an MSM that summarizes the dynamics captured in the MD trajectories. Various statistical techniques then inform on the quality of the obtained MSM.

Each MD trajectory is a series of conformations accessed consecutively in an MD simulation. An important decision with implications for the ability of the pipeline to identify macrostates relates to the degree of the retained conformation representation detail for the purpose of assigning conformations to macrostates. We address it empirically, by considering several reasonable representations and analyzing their impact on the resulting MSMs.

The second important decision concerns the lag time. It is often not feasible to analyze all the conformations obtained via MD simulation. Most MSM construction tool allow the user to select a lag time. This can be a multiple of the original time step between two successive conformations in an MD trajectory. Selection of the lag time is in its essence a data reduction strategy, as it allows skipping over conformations; as it can result in loss of temporal and spatial resolution of the constructed MSM, we analyze its impact on the

quality of the resulting MSM, as well.

From MD Trajectories to Macrostates (Basins in the Landscape)

As per our earlier studies [89, 90], a better strategy to identify macrostates is to utilize the energies of conformations obtained via MD simulation. In fact, the energy landscape contains information on how conformations with similar energies inter-convert into one another and so provides an opportunity to quantitatively understand the underlying dynamics of a system of interest [144]. Our definition of macrostates utilizes this energy landscape view of dynamics. A macrostate, which is a thermodynamically-stable (or semi-stable) state does not directly rely on geometric similarity but instead corresponds to basins/wells in the energy landscape. So we leverage our earlier work on summarizing sampled conformation-energy pairs via energy basins [27]. Identification of basins relies on embedding the conformations collected over all MD trajectories into a nearest-neighbor graph [39], where each vertex is a conformation-energy pair. The methodological details are available in sections 5.2.2 and 5.2.3.

In summary, the methodology groups conformations into distinct, non-overlapping energy basins. The conformation that sits at the very bottom of a basin is denoted as its *focal minimum* and is used as a unique identifier of a basin. Since basins contain actual conformations, a basin can be summarized in terms of energies of the conformations in it (minimum, mean, maximum energy), as well as the geometric dissimilarity among conformations, measured, for instance, via the maximum pairwise root-mean-squared-deviation (RMSD) [145] between conformations in a basin. We consider the identified basins to be the macrostates.

From Macrostates to State-to-State Transition Probabilities (The Construction of the MSM)

The next step is to compute the state-to-state transition probabilities. The MD trajectories are utilized for this purpose. In a given MD trajectory, a conformation a is followed

by a conformation b. Let us consider that the process of assigning the conformations into macrostates, has assigned some conformation a to some macrostate St_i and some conformation b to some macrostate St_j. The observed transition from a to b is the evidence of the transition from macrostate St_i to macrostate St_j and thus contributes one count to the total counts of transitions from St_i to St_j. In this way, MD trajectories contribute to the "count" matrix of transitions between macrostates. The counts are normalized to turn them into probabilities.

Let us assume that the basin identification process above has resulted in M disjoint states St_1, St_2, ..., St_M. A matrix of conditional transition probabilities between these states is estimated from the simulation trajectories x_t [20]. The transition matrix, $\mathbf{T} \equiv (\mathrm{P}_{ij})$: $\mathrm{P}_{ij}(\tau) = Prob\ (\mathrm{x}_{t+\tau} \in St_j \mid \mathrm{x}_t \in St_i)$, where τ is the chosen lag time. This transition matrix is the tangible product of what is referred to as the construction of the MSM. It contains all the information needed, as every entry in the transition matrix specifies the probability with which two states inter-convert into one another, thus summarizing the dynamics of the system under investigation.

The transition matrix can provide more information about the system under investigation through its eigen-decomposition into eigenvectors and eigenvalues. The highest eigenvalue (with a value of 1) and its corresponding eigenvector represents the equilibrium/stationary distribution. The higher the population of a macrostate, the more thermodynamically-stable the macrostate is.

Statistical Evaluation

The MSM resulting from the computational process described above is subjected to rigorous analysis in order to evaluate whether the constructed MSM is reliable to utilize for making predictions regarding dynamics. As in other studies [90, 146], we employ two main tests, the convergence analysis and the Chapman-Kolmogorov (CK) test. Both check for the Markov property that the MSM is memory-less; that is, the conditional probability distribution of future macrostates depends only upon the current macrostate and not on prior

macrostates [19]. In our analysis, we conduct both evaluations whether the state-space decomposition (assignment of conformations into macrostates) results in a high-quality MSM.

Convergence Analysis: The convergence analysis tests whether the duration of the lag time is sufficient to guarantee that the state space discretization maintains the Markov property. If the state space decomposition is accurate, conformations within a state interconvert on timescales faster than the lag time and transition to other states on slower timescales. It is standard practice to verify this property visually, via interpretation of the generated implied timescale plot of the model relaxation timescale versus model lag time. One expects to observe an exponential decay in the plot to system equilibrium. With relaxation timescales being physical properties of the system, ideally, the implied timescales need to be independent of the lag time. According to the variational principle of conformation dynamics [147], it is desirable for the model to have a longer timescale. For an ideal model with good discretization, the implied timescales plot exhibit convergence within fewer steps.

CK Test: Discretization error can result in a deterministic fluctuation of the MSM dynamics from the actual dynamics that remains persistent even when excluding statistical error by means of excessive sampling [148]. The propagation error on the discrete space is measured by checking whether the approximation, $[\hat{\mathbf{T}}(\tau)]^k \approx \hat{\mathbf{T}}(k\tau)$ holds within statistical uncertainty where, $\hat{\mathbf{T}}(\tau)$ is the transition matrix estimated from the data at lag time τ, and $\hat{\mathbf{T}}(k\tau)$ is the transition matrix estimated from the same data at longer lag times kτ. The Python library we use for this purpose, PyEMMA, allows testing different models. Given a model estimated at lag time τ, a prediction can be made of a model quantity for lag time $k\tau$; the prediction can then be compared to an independently-estimated model at $k\tau$. The CK test computes the transition probability between meta-stable states for increasing lag times. The determination is made visually; ideally, plots show that the estimated and the predicted model exhibit negligible deviation.

9.2.2 Conformation Generation and Preparation

MD Simulations for Conformation Generation

The free antibody molecule (PDB ID: `1IGT`) contains $20,544$ atoms ($1,322$ amino acids). The antigen-antibody complex contains $21,092$ atoms ($1,356$ amino acids). Initial antibody random conformations are generated by adjusting three sets of torsion angles: 231C-232N-232CA-232C, 232N-232CA-232C-233N, and 232CA-232C-233N-233CA (numbering in `1IGT`), each step with $60°$ rotation. During the conformation randomization, the `Fc` domain is kept fixed, whereas the `Fab` domain moves freely, leading to 216 conformations. Excluding conformations with closed `Fab` domain or with `Fc` domain clashes, 12 conformations are selected as the starting points for the MD simulations. That is, we perform 12 independent MD simulations of the antibody with 12 different initial conformations. Each MD trajectory is $53-54$ nanoseconds long, with a time step of 4 picoseconds between two consecutive frames/conformations. Thus, a total of $160,000$ conformations are generated for both the free antibody and the antibody-antigen complex in this manner. The MD simulations are conducted using the NAMD software [17] with CHARMM force field [142]. Further details regarding the process of conformation generation can be found in [142].

9.2.3 Preparation of Conformations for Analysis

From a given trajectory file (`.dcd` format), conformations are extracted using the *mdconvert* command-line script from the *MDTraj* python library. Considering a time lag of 128 picoseconds, which corresponds to selecting every 32nd frame in a trajectory file, we obtain $5,000$ conformations for the free antibody and the antigen-antibody complex, respectively. We consider several options for representing conformations of each system.

Cartesian Coordinate-based Representations

Setting 1: In this setting, each conformation is simplified to a high-dimensional point ($CA_1.x$, $CA_1.y$, $CA_1.z$, $CA_5.x$, $CA_5.y$, $CA_5.z$, $CA_{10}.x$, $CA_{10}.y$, $CA_{10}.z$, $\ldots$), effectively

skipping every 4 consecutive CA atoms. This makes it computationally feasible to compare conformations, which is a central step to embedding conformations in a nearest-neighbor graph for the purpose of identifying basins.

Setting 2: In this setting, we consider all CA atoms; it is more computationally costly to compare conformations via this representation and identify basins but one does retain more detail.

Principal Component-based Representations

We rely on Principal Component Analysis (PCA) to identify a few variance-preserving dimensions along which to project collected conformations and obtain "reduced" coordinates for them. PCA is popular to analyze protein conformations [88, 149–151]. Specifically, we construct a matrix $A_{3k\times n}$, where $n = 5,000$ conformations and k is the number of CA atoms in the molecule of interest. All conformations are first optimally superimposed over the first conformation (chosen arbitrarily to be the reference conformation). The reference conformation is then "subtracted" from each of the conformations, and the matrix stores the resulting deviations. The purpose for this is to capture internal conformation changes rather than differences due to rigid-body motions (translations and rotations in three-dimensional space). A singular value decomposition of $\frac{1}{n-1} \cdot A$ is then carried out to obtain the eigen-decomposition $\frac{1}{n-1} \cdot A_{3k\times n} = U_{3k\times n} \cdot \mathcal{E}_{n\times n} \cdot V^T_{n\times n}$. The eigenvectors/PCs are $3k$-dimensional vectors stored in the columns of U, in order of corresponding highest-to-lowest singular values; these are stored in the diagonal of the $\mathcal{E}$ matrix. The singular values σ_i are square roots of the eigenvalues e_i, which provide the variance of the original (displacement) data over the corresponding eigenvector PC_i.

The main decision after utilizing PCA is to determine how many (projection) coordinates to be employed for representing a conformation. Typically, an accumulation of variance analysis is conducted. After ordering the PCs by corresponding eigenvalue (highest to lowest), at each PC_i, the cumulative variance of $\{PC_1, \ldots, PC_i\}$ is plotted. The individual variance of each PC is its eigenvalue, normalized over all eigenvalues (of all n obtained PCs).

For the free antibody, the first five PCs cumulatively cover 80.89% of the total variance. The first eight PCs cover 90.51% of the variance, and the first 23 PCs cover 99.05% of the variance. Similarly, for the antigen-antibody complex, the first five PCs cumulatively cover 82.01% of the variance; the first nine PCs cover 91.77% of the variance, and the first 24 PCs cover 99.09% of the variance.

Setting 3: We set our goal at capturing no lower than 90% of the total variance. This means that for the free antibody, each conformation is represented with 8 coordinates (projections of a conformation on the top 8 PCs); for the antigen-antibody complex, this threshold means that each conformation is represented with 9 coordinates.

Setting 4: We set the goal at capturing no lower than 99% of the variance. This means that for the free antibody, each conformation is represented with 23 coordinates; for the antigen-antibody complex, this threshold means that each conformation is represented with 24 coordinates.

Time-lagged independent component-based Representations

TICA is a linear transformation method. In contrast to PCA, which finds coordinates of maximal variance, TICA finds coordinates of maximal auto-correlation at the given lag time. TICA is useful to find the slow components in a dataset and a reasonable choice to transform MD data.

Setting 5: In this setting, our goal is to capture no lower than 90% of the total variance. This means that for the free antibody, each conformation is represented with 2273 coordinates (projections of a conformation on the top 2273 components); for the antigen-antibody complex, this threshold means that each conformation is represented with 2341 coordinates.

Setting 6: We now set the goal at capturing no lower than 99% of the variance. This means that for the free antibody, each conformation is represented with 3189 coordinates; for the antigen-antibody complex, this threshold means that each conformation is represented with 3275 coordinates.

9.2.4 Evaluation Setup

Whether for the free antibody or the antigen-antibody complex, we construct six different MSMs corresponding to the described settings. Convergence analysis and CK tests are utilized to evaluate the quality of each MSM. The best MSM (setting-1) obtained for each system is investigated in greater detail in the results section, and comparisons are made to understand the main differences in the conformation dynamics between the free and antigen-bound antibody.

9.3 Evaluation Results

Out of the six settings, according to the convergence analysis and the CK test, the best MSM obtained for the free antibody as well as for the antigen-antibody complex results from Setting 1. The rest of the analysis in this section focuses on the basins and MSMs resulting from this setting.

9.3.1 Visual Comparison of Embedded Landscapes

We first relate in Fig. 9.1(left panel) a two-dimensional embedding of the energy landscape of the free antibody and the antigen-antibody complex, respectively. The 5000 conformations sampled from the 12 MD trajectories for each system, are subjected to PCA. Each dot shows the projection of a conformation on the top two (highest-variance) PCs; two PCs capture close to 60% of the total variance for the free antibody and more than 60% of the variance for the antigen-antibody complex. The color-coding relates the energies of the projected conformations, with a blue-to-red color-scheme denoting low-to-high internal energies.

The energy landscape of the free antibody contains four main clusters with a diffused distribution of low-energy conformations (top-left panel). After antigen binding, the distribution of the clusters becomes more diffusive (bottom-left panel), and the low-energy conformations are enriched in only two clusters. These observations hold even when more

conformations are included in the analysis; adding $5,000$ more conformations for each system (selected every 16th frame over the MD trajectories) does not change the main features of the embedded landscapes; existing clusters do not merge, and no new clusters emerge as shown in Fig. 9.1(right panel).

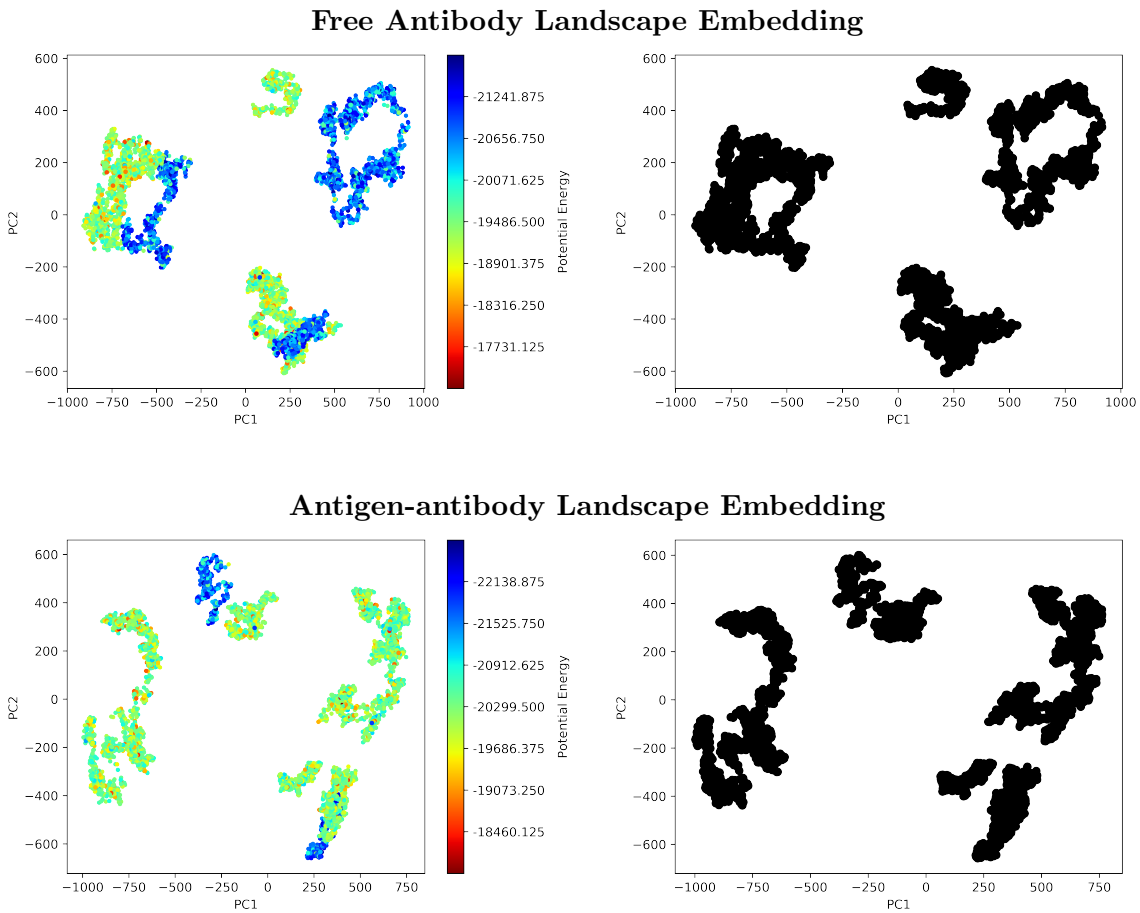


Figure 9.1: Left panel: Embedding of $5,000$ sampled conformations (selected every 32nd frame in MD trajectories over the top two PCs for (top panel) the free antibody and (bottom panel) antigen-antibody complex; projections are color-coded by the energies of corresponding conformations (low-to-high in a blue-to-red color scheme). Right panel: $5,000$ more conformations are sampled in each case, for a total of $10,000$ (selected every 16th frame in MD trajectories); PCA is re-conducted, and projections are shown.

9.3.2 Basin Analysis

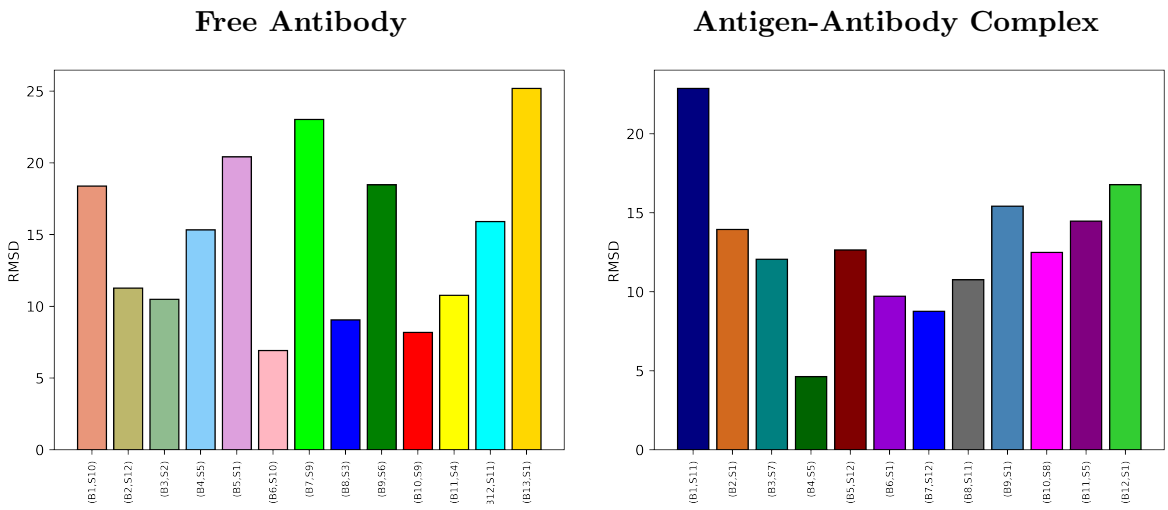


Figure 9.2: The minimum RMSD (Å) (over backbone atoms) between the focal minimum conformation representing a basin and the conformations starting each of the MD trajectories is shown. The x-axis shows the pairs corresponding to the minimum RMSD values. B* denotes (the focal minimum conformation of) a basin, numbered B1-13 for the free antibody and B1-12 for the antigen-antibody complex. S* denotes the conformation starting an MD trajectory, numbered S1-12.

Thirteen basins are identified over the free antibody landscape; twelve basins are identified over the antigen-antibody conformations. We evaluate whether the identification of basins is biased by the starting conformation of the various MD trajectories as follows. The focal minimum conformation that sits at the bottom of a basin is considered as an identifier and representative conformation of a basin. The RMSD between this conformation and each conformation starting an MD trajectory is calculated. This calculation is repeated for each basin (representative conformation) against all starting conformations (of all MD trajectories), and related in Fig. 9.2.

Fig. 9.2 shows that the identification of basins is not biased by the conformations starting

the MD trajectories. The method does not trivially assign the conformation that initiates an MD trajectory as the focal minimum of a basin. For the majority of the basins, the focal minimum conformation that uniquely represents a basin (its deepest point) resides on average more than 10Å away from the starting conformation of a trajectory. Some lower values are noted: between 5 and 10Å for 3 of the focal minima identified for the free antibody and for 2 of the focal minima identified for the antigen-antibody complex; in the latter case, one of the focal minima resides closer than 5Å to the starting conformation of an MD trajectory.

9.3.3 Summarization and Comparison of Dynamics

The best MSMs constructed for each system, free antibody versus antigen-antibody complex, are related in Fig. 9.3. For each system, the equilibrium/stationary distribution is shown first in the top panel via a pie chart limited to the 6 most populous macrostates (labeled as B1-6 for basins; the numbering of basins observes the basin size). In each pie chart, the population of the remaining macrostates is accumulated and labeled as B*. The bottom panel shows the transitions (again limited to the 6 most populous macrostates identified by the stationary distribution analysis).

Interesting observations can be drawn for the free antibody from the left panel of Fig. 9.3. One macrostate (largest basin), is significantly more thermodynamically-stable than the others, with an equilibrium population (31%) close to being 1.5 times than that of the next stable macrostate (23.95%). Three other macrostates have populations in the $8-16\%$ range, and the rest are 5.04% or lower. The self-transition probabilities are large. For four of the six basins, the self-transition probabilities are very high (above 0.94). Two exceptions are noted. Basin 1 has a lower self-transition probability of 0.8412. Basin 5 has an even lower self-transition probability of 0.623. While the cumulative out-of-basin transition probabilities for four basins (B2, B3, B4, B6) are just below 0.045; basin 5 transitions to basin 1 with a higher probability of 0.325, and basin 1 transitions to basin 5 with a probability of 0.1195.

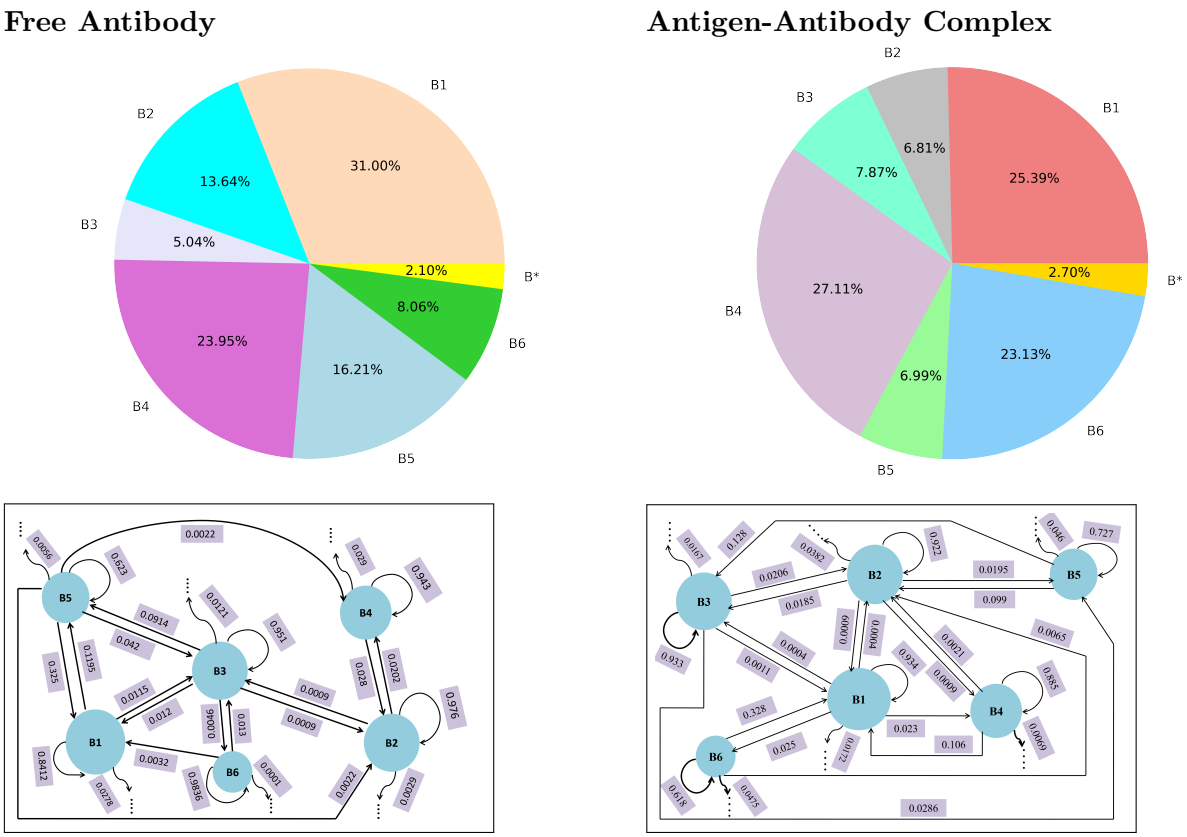


Figure 9.3: Top Panel: Pie chart of adjusted state populations, showing the stationary distribution for the 6 top-populated macrostates/basins, with the other states accumulated in B*. Bottom Panel: MSM schematic. Basins are drawn as disks, with radii proportional to size (number of conformations). Transitions between basins are drawn as arrows, and transition probabilities are shown. The visual summary is restricted to the six top-populated states. Trailing arrows indicate transitions to other states.

A similar analysis carried out over the MSM shown in the right panel of Fig. 9.3 for the antigen-antibody complex. The stationary distribution is not as skewed as for the free antibody. There is no single macrostate with a population significantly higher than others; three macrostates (B1, B4, B6) have comparable populations in the $23 - 27\%$ range. The self-transition probabilities for the antigen-antibody complex are lower than those observed for the free antibody. Basins 1-3 have self-transition probabilities between 0.92 and 0.93. Basins 4-6 have much lower self-transition probabilities in the range 0.6-0.88. Basin 4 transitions to basin 1 with a probability of 0.11. Basin 5 transitions to basin 3 with a probability of 0.13 and to basin 2 with a probability of 0.1. Basin 6 transitions to basin 1 with a probability of 0.3. This suggests that the energy landscape of the antigen-antibody complex allows for more transitions among the various basins than the energy landscape of the free antibody.

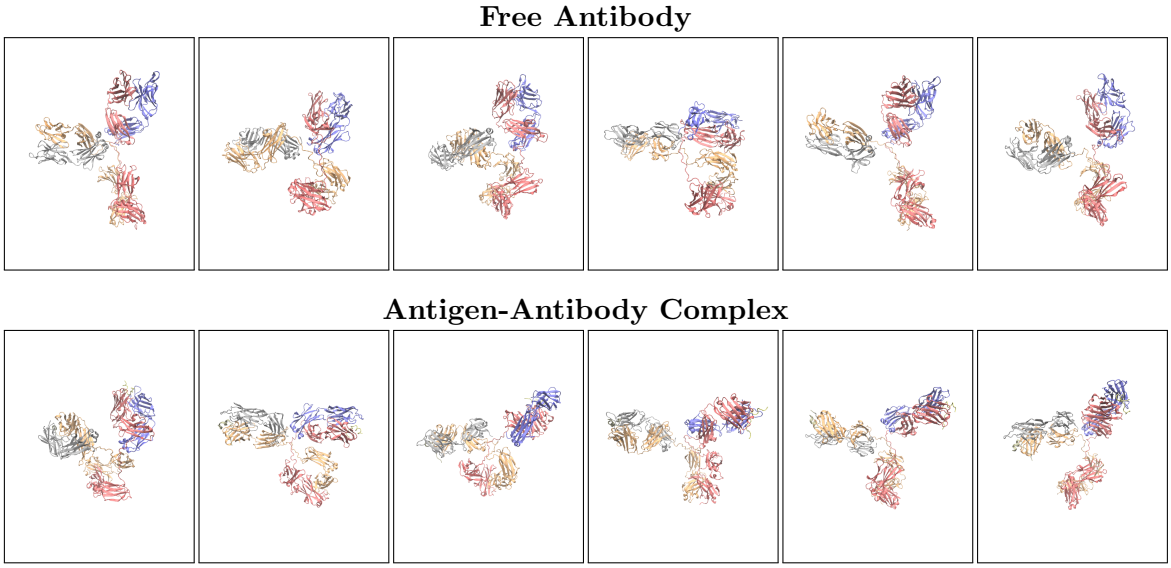


Figure 9.4: Focal minima conformations corresponding to basins B1-6 in the MSMs shown schematically above are drawn here with VMD [152] for (top panel) the free antibody and the (bottom panel) antigen-antibody complex, respectively. Chains are drawn in different colors.

9.3.4 Visualization of the Largest Basins.

Finally, we visualize the focal minima conformations corresponding to the 6 largest/most-populous basins for the free antibody and the antigen-antibody complex. Fig. 9.4 visualizes these conformations with the help of graphical representation in the VMD software [152] using different colors for the various `IgG` domains.

9.3.5 Discussion

Based on the first two PCs and the conformational energies, we find that the conformational energy landscape of the free antibody mainly contains four clusters with a diffused distribution of low-energy conformers. After antigen binding, the distribution of the four clusters becomes more diffusive. However, the low-energy conformers' distributions narrowed and are enriched only in two of the four clusters. Such behavior provides new insights into previous analyses. Previous studies found that the free antibody has one major cluster that splits into two clusters after antigen binding. Both the current work and previous studies agree on the two major clusters of antigen-antibody complexes, but the analysis presented here provides a more detailed view of the energy landscape.

The MSM-based analysis in this study shows that, with antigen binding, there are considerable conformation transitions among the different basins. These results suggest that the antigen-bounded form with high energy may provide many dynamic processes to further enhance co-factor binding of the antibody in the next step. We also observe that antigen binding causes reduction in the number of macrostates/basins across all the settings. Simulating the dynamics of large proteins and their complexes places large computational demands. Analyzing their conformation dynamics poses additional difficulties.

9.4 Summary

This chapter demonstrates how one can extend application of the framework developed in chapter 5 from small peptides to larger, even complexated biological systems and so makes

a case for the broader utility and generalization of state space discretization via Markov State models (MSM) of dynamics.

Chapter 10: Conclusion and Future Work

This dissertation has made several contributions in developing and applying unsupervised learning frameworks for organizing the structure space of peptides, uncomplexated, and complexated protein molecules to reveal one or more functionally-important (biologically-relevant) structural states. The major contributions can be summarized as:

- Chapter 4 demonstrated the utility of embedding computationally sampled tertiary structures of a protein molecule in a graph and leveraging the graph embedding coupled with community detection algorithms to elucidate the organization of the structure space by extracting groups that are highly likely to contain functionally-relevant structures.

- Chapter 5 introduced and demonstrated the utilization of an energy landscape-based framework to leverage the organization of the protein molecular structure space and construct discrete models of structural dynamics.

- Chapter 6 marked the opening of a relatively new avenue of research for molecular structure modeling under the umbrella of matrix factorization in addressing the challenges of identifying functionally-relevant structures from sparse datasets. Chapter 7 followed this line of work with a more powerful, feature-free, and non-parametric method with additional capabilities of not only extracting functionally-relevant groups but also selecting a single biologically-relevant structure.

- Chapter 8 introduced a more robust hybrid framework relying on tensor factorization which serves not only as a single-structure and multi-structure method but also as a complete solution package for the hallmark EMA problem.

- Chapter 9 investigated the applicability of the energy landscape-based approach on organizing structures of an antibody-antigen bound molecular system obtained via MD simulation to show the generalizability of the approach presented in Chapter 5.

Future Directions

We hope that the works presented in this dissertation will further inspire researchers along several envisioned research directions.

- It would be worthwhile to investigate the applicability of different methods described in this dissertation, especially the factorization-based techniques, in constructing discrete models of dynamics capable of quantifying the state-to-state structure transitions at equilibrium, revealing crucial mechanistic information about a molecule.

- Another interesting direction would be to figure out how different factorization-based frameworks can be employed to support additional applications, such as molecular docking, which is a vital component of the drug discovery and design process. It is worth mentioning here that such settings go beyond a single molecule system and hence this also comes with additional challenges, such as determining the role of different types of energies (potential energy, interaction energy) in the underlying process.

- On a more general note, finding the subset of structural coordinate dimensions that reveal the most informative grouping corresponding to the functionally-relevant ones warrants further exploration. One can consider dimension weighting and reduce the problem to learning the weights of the different dimensions to optimize an objective function that measures the quality of the biologically-relevant groups. Probably, a more sophisticated way to approach this is to apply subspace clustering, which is a relatively unexplored domain for these problem settings.

Finally, while the presented frameworks and evaluation techniques focuses on the tertiary structures of protein molecules, the methods and the selection strategies are general and can be extended beyond the settings of protein molecular system and investigated more generally for high-dimensional spaces.

Bibliography

Bibliography

[1] D. D. Boehr and P. E. Wright, "How Do Proteins Interact?" *Science*, 2008.

[2] T. Maximova, R. Moffatt, B. Ma, R. Nussinov, and A. Shehu, "Principles and Overview of Sampling Methods for Modeling Macromolecular Structure and Dynamics," *PLoS computational biology*, vol. 12, no. 4, p. e1004619, 2016.

[3] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, "The Shape and Structure of Proteins," in *Molecular Biology of the Cell. 4th edition.* Garland Science, 2002.

[4] D. D. Boehr, R. Nussinov, and P. E. Wright, "The Role of Dynamic Conformational Ensembles in Biomolecular Recognition," *Nature chemical biology*, vol. 5, no. 11, pp. 789–796, 2009.

[5] R. P. Feynman, R. B. Leighton, and M. Sands, "The Feynman Lectures on Physics," *American Journal of Physics*, vol. 33, no. 9, pp. 750–752, 1965.

[6] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000. [Online]. Available: https://www.rcsb.org/

[7] D. Devaurs, K. Molloy, M. Vaisset, A. Shehu, T. Siméon, and J. Cortés, "Characterizing energy landscapes of peptides using a combination of stochastic algorithms," *IEEE transactions on nanobioscience*, vol. 14, no. 5, pp. 545–552, 2015.

[8] K. Molloy, R. Clausen, and A. Shehu, "A stochastic roadmap method to model protein structural transitions," *Robotica*, vol. 34, no. 8, pp. 1705–1733, 2016.

[9] K.-i. Okazaki, N. Koga, S. Takada, J. N. Onuchic, and P. G. Wolynes, "Multiple-basin energy landscapes for large-amplitude conformational motions of proteins: Structure-based molecular dynamics simulations," *Proceedings of the National Academy of Sciences*, vol. 103, no. 32, pp. 11 844–11 849, 2006.

[10] 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.

[11] R. Nussinov and P. G. Wolynes, "A second molecular biology revolution? the energy landscapes of biomolecular function," *Physical Chemistry Chemical Physics*, vol. 16, no. 14, pp. 6321–6322, 2014.

[12] N. Akhter, "Summarization, Visualization, and Mining of Molecular Landscapes," Ph.D. dissertation, George Mason University, 2020.

[13] M. Karplus and J. A. McCammon, "Molecular dynamics simulations of biomolecules," *Nature structural biology*, vol. 9, no. 9, pp. 646–652, 2002.

[14] S. A. Hollingsworth and R. O. Dror, "Molecular dynamics simulation for all," *Neuron*, vol. 99, no. 6, pp. 1129–1143, 2018.

[15] D. A. Case, V. Babin, J. Berryman, R. Betz, Q. Cai, D. Cerutti, T. Cheatham Iii, T. Darden, R. Duke, H. Gohlke *et al.*, "Amber 14," 2014.

[16] E. Lindahl, B. Hess, and D. Van Der Spoel, "Gromacs 3.0: a package for molecular simulation and trajectory analysis," *Molecular modeling annual*, vol. 7, no. 8, pp. 306–317, 2001.

[17] J. C. Phillips, D. J. Hardy, J. D. Maia, J. E. Stone, J. V. Ribeiro, R. C. Bernardi, R. Buch, G. Fiorin, J. Hénin, W. Jiang *et al.*, "Scalable molecular dynamics on cpu and gpu architectures with namd," *The Journal of chemical physics*, vol. 153, no. 4, p. 044130, 2020.

[18] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. a. Swaminathan, and M. Karplus, "Charmm: a program for macromolecular energy, minimization, and dynamics calculations," *Journal of computational chemistry*, vol. 4, no. 2, pp. 187–217, 1983.

[19] R. D. Malmstrom, C. T. Lee, A. T. Van Wart, and R. E. Amaro, "Application of Molecular Dynamics-based Markov State Models to Functional Proteins," *Journal of chemical theory and computation*, vol. 10, no. 7, pp. 2648–2657, 2014.

[20] J. D. Chodera and F. Noé, "Markov State Models of Biomolecular Conformational Dynamics," *Current Opinion in Structural Biology*, vol. 25, pp. 135–144, 2014.

[21] A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak, K. W. Kaufman, P. D. Renfrew, C. A. Smith, W. Sheffler *et al.*, "ROSETTA3: An Object-oriented Software Suite for the Simulation and Design of Macromolecules," *Methods in Enzymology*, vol. 487, pp. 545–574, 2011.

[22] 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.

[23] 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.

[24] A. B. Zaman, "Evolutionary Techniques for De Novo Protein Conformation Ensemble Generation," Ph.D. dissertation, George Mason University, 2021.

[25] D. Kihara, H. Chen, and Y. D. Yang, "Quality assessment of protein structure models," *Current Protein and Peptide Science*, vol. 10, no. 3, pp. 216–228, 2009.

[26] J. Chen and S. W. Siu, "Machine learning approaches for quality assessment of protein structures," *Biomolecules*, vol. 10, no. 4, p. 626, 2020.

[27] N. Akhter and A. Shehu, "From Extraction of Local Structures of Protein Energy Landscapes to Improved Decoy Selection in Template-free Protein Structure Prediction," *Molecules*, vol. 23, no. 1, p. 216, 2018.

[28] N. Akhter, G. Chennupati, K. L. Kabir, H. Djidjev, and A. Shehu, "Unsupervised and supervised learning over the energy landscape for protein decoy selection," *Biomolecules*, vol. 9, no. 10, p. 607, 2019.

[29] N. Akhter, K. L. Kabir, G. Chennupati, R. Vangara, B. Alexandrov, H. N. Djidjev, and A. Shehu, "Improved Protein Decoy Selection via Non-negative Matrix Factorization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

[30] K. L. Kabir, G. Chennupati, R. Vangara, H. Djidjev, B. S. Alexandrov, and A. Shehu, "Decoy selection in protein structure determination via symmetric non-negative matrix factorization," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 23–28.

[31] K. Uziela and B. Wallner, "Proq2: estimation of model accuracy implemented in rosetta," *Bioinformatics*, vol. 32, no. 9, pp. 1411–1413, 2016.

[32] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *J Amer Chem Soc*, vol. 118, no. 9, pp. 2309–2309, 1996.

[33] T. Lazaridis and M. Karplus, "Discrimination of the native from misfolded protein models with an energy function including implicit solvation," *J Mol Biol*, vol. 288, no. 3, pp. 477–487, 1999.

[34] S. Miyazawa and R. L. Jernigan, "An empirical energy potential with a reference state for protein fold and sequence recognition," *Proteins: Struct, Funct, and Bioinf*, vol. 36, no. 3, pp. 357–369, 1999.

[35] B. J. McConkey, V. Sobolev, and M. Edelman, "Discrimination of native protein structures using atom–atom contact scoring," *Proc Natl Acad Sci USA*, vol. 100, no. 6, pp. 3215–3220, 2003.

[36] K. T. Simons, I. Ruczinski, C. Kooperberg, B. A. Fox, C. Bystroff, and D. Baker, "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins," *Proteins: Struct, Funct, and Bioinf*, vol. 34, no. 1, pp. 82–95, 1999.

[37] B. Park and M. Levitt, "Energy functions that discriminate X-ray and near-native folds from well-constructed decoys," *J Mol Biol*, vol. 258, no. 2, pp. 367–392, 1996.

[38] A. K. Felts, E. Gallicchio, A. Wallqvist, and R. M. Levy, "Distinguishing native conformations of proteins from decoys with an effective free energy estimator based on the opls all-atom force field and the surface generalized born solvent model," *Proteins: Struct, Funct, and Bioinf*, vol. 48, no. 2, pp. 404–422, 2002.

[39] K. L. Kabir, L. Hassan, Z. Rajabi, N. Akhter, and A. Shehu, "Graph-based community detection for decoy selection in template-free protein structure prediction," *Molecules*, vol. 24, no. 5, p. 854, 2019.

[40] A. Kryshtafovych, A. Barbato, K. Fidelis, B. Monastyrskyy, T. Schwede, and A. Tramontano, "Assessment of the assessment: evaluation of the model quality estimates in casp10," *Proteins: Struct, Funct, and Bioinf*, vol. 82, pp. 112–126, 2014.

[41] A. Kryshtafovych, B. Monastyrskyy, K. Fidelis, T. Schwede, and A. Tramontano, "Assessment of model accuracy estimations in casp12," *Proteins: Struct, Funct, and Bioinf*, vol. 86, pp. 345–360, 2018.

[42] H. Li and Y. Zhou, "SCUD: fast structure clustering of decoys using reference state to remove overall rotation," *J Comp Chem*, vol. 26, no. 11, pp. 1189–1192, 2005.

[43] S. C. Li and Y. K. Ng, "Calibur: a tool for clustering large numbers of protein decoys," *BMC Bioinf*, vol. 11, no. 1, p. 25, 2010.

[44] F. Berenger, Y. Zhou, R. Shrestha, and K. Y. Zhang, "Entropy-accelerated exact clustering of protein decoys," *Bioinformatics*, vol. 27, no. 7, pp. 939–945, 2011.

[45] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New Directions in Statistics Physics*, L. T. Wille, Ed. Berlin, Heidelberg: Springer, 2004, pp. 273–309.

[46] Z. He, M. Alazmi, J. Zhang, and D. Xu, "Protein structural model selection by combining consensus and single scoring methods," *PLoS ONE*, vol. 8, no. 9, p. e74006, 2013.

[47] M. Pawlowski, L. Kozlowski, and A. Kloczkowski, "MQAPsingle: A quasi single-model approach for estimation of the quality of individual protein structure models," *Proteins: Struct, Funct, and Bioinf*, vol. 84, no. 8, pp. 1021–1028, 2016.

[48] X. Jing, K. Wang, R. Lu, and Q. Dong, "Sorting protein decoys by machine-learning-to-rank," *Sci Reports*, vol. 6, p. 31571, 2016.

[49] S. Chatterjee, S. Ghosh, and S. Vishveshwara, "Network properties of decoys and CASP predicted models: a comparison with native protein structures," *Molecular BioSystems*, vol. 9, no. 7, pp. 1774–1788, 2013.

[50] B. Manavalan and J. Lee, "SVMQA: support–vector-machine-based protein single-model quality assessment," *Bioinformatics*, vol. 33, no. 16, pp. 2496–2503, 2017.

[51] B. Manavalan, J. Lee, and J. Lee, "Random forest-based protein model quality assessment (RFMQA) using structural features and potential energy terms," *PloS one*, vol. 9, no. 9, p. e106542, 2014.

[52] S. P. Nguyen, Y. Shang, and D. Xu, "DL-PRO: A novel deep learning method for protein model quality assessment," in *Int Conf Neural Networks (IJCNN)*. IEEE, 2014, pp. 2071–2078.

[53] R. Cao, D. Bhattacharya, J. Hou, and J. Cheng, "DeepQA: improving the estimation of single protein model quality with deep belief networks," *BMC Bioinf*, vol. 17, no. 1, p. 495, 2016.

[54] S. Mirzaei, T. Sidi, C. Keasar, and S. Crivelli, "Purely structural protein scoring functions using support vector machine and ensemble learning," *IEEE/ACM Trans Comp Biol & Bioinf*, 2016.

[55] Z. He, Y. Shang, D. Xu, Y. Xu, and J. Zhang, "Protein structural model selection based on protein-dependent scoring function," *Statistics & Interface*, vol. 5, no. 1, pp. 109–115, 2012.

[56] H. Zhou and J. Skolnick, "GOAP: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction," *Biophys J*, vol. 101, no. 8, pp. 2043–2052, 2011.

[57] J. Qiu, W. Sheffler, D. Baker, and W. S. Noble, "Ranking predicted protein structures with support vector regression," *Proteins: Struct, Funct, and Bioinf*, vol. 71, no. 3, pp. 1175–1182, 2008.

[58] A. Ray, E. Lindahl, and B. Wallner, "Improved model quality assessment using proq2," *BMC Bioinf*, vol. 13, no. 1, p. 224, 2012.

[59] J. Hou, R. Cao, and J. Cheng, "Deep convolutional neural networks for predicting the quality of single protein structural models," *bioRxiv*, p. 590620, 2019.

[60] G. Pagès, B. Charmettant, and S. Grudinin, "Protein model quality assessment using 3d oriented convolutional neural networks," *bioRxiv*, p. 432146, 2018.

[61] R. Sato and T. Ishida, "Protein model accuracy estimation based on local structure quality assessment using 3d convolutional neural network," *PloS one*, vol. 14, no. 9, p. e0221347, 2019.

[62] J. Hou, T. Wu, R. Cao, and J. Cheng, "Protein tertiary structure modeling driven by deep learning and contact distance prediction in casp13," *Proteins: Structure, Function, and Bioinformatics*, 2019.

[63] M. Wiltgen, "Algorithms for Structure Comparison and Analysis: Homology Modeling of Proteins," *Encyclopedia of Bioinformatics and Computational Biology*, pp. 38–61, 2018.

[64] P. J. Ballester and W. G. Richards, "Ultrafast shape recognition to search compound databases for similar molecular shapes," *Journal of computational chemistry*, vol. 28, no. 10, pp. 1711–1723, 2007.

[65] A. B. Zaman, P. Kamranfar, C. Domeniconi, and A. Shehu, "Reducing ensembles of protein tertiary structures generated de novo via clustering," *Molecules*, vol. 25, no. 9, p. 2228, 2020.

[66] Ó. Álvarez, J. L. Fernández-Martínez, C. Fernández-Brillet, A. Cernea, Z. Fernández-Muñiz, and A. Kloczkowski, "Principal Component Analysis in Protein Tertiary Structure Prediction," *Journal of bioinformatics and computational biology*, vol. 16, no. 02, p. 1850005, 2018.

[67] G. A. Tribello and P. Gasparotto, "Using Dimensionality Reduction to Analyze Protein Trajectories," *Frontiers in molecular biosciences*, vol. 6, p. 46, 2019.

[68] K. Olechnovič, B. Monastyrskyy, A. Kryshtafovych, and Č. Venclovas, "Comparative Analysis of Methods for Evaluation of Protein Models against Native Structures," *Bioinformatics*, vol. 35, no. 6, pp. 937–944, 2019.

[69] A. D. McLachlan, "A mathematical procedure for superimposing atomic coordinates of proteins," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 28, no. 6, pp. 656–657, 1972.

[70] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer, "MaxSub: An Automated Measure for the Assessment of Protein Structure Prediction Quality," *Bioinformatics*, vol. 16, no. 9, pp. 776–785, 2000.

[71] 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.

[72] K. Raha and K. M. Merz Jr, "Calculating binding free energy in protein–ligand interaction," *Annual reports in computational chemistry*, vol. 1, pp. 113–130, 2005.

[73] M. A. Murcko, "Computational methods to predict binding free energy in ligand-receptor complexes," *Journal of medicinal chemistry*, vol. 38, no. 26, pp. 4953–4967, 1995.

[74] E. C. Meng, B. K. Shoichet, and I. D. Kuntz, "Automated docking with grid-based energy evaluation," *Journal of computational chemistry*, vol. 13, no. 4, pp. 505–524, 1992.

[75] W. Wang, W. A. Lim, A. Jakalian, J. Wang, J. Wang, R. Luo, C. I. Bayly, and P. A. Kollman, "An analysis of the interactions between the sem- 5 sh3 domain and its ligands using molecular dynamics, free energy calculations, and sequence analysis," *Journal of the American Chemical Society*, vol. 123, no. 17, pp. 3986–3994, 2001.

[76] S. Neelamraju, D. J. Wales, and S. Gosavi, "Protein energy landscape exploration with structure-based models," *Current opinion in structural biology*, vol. 64, pp. 145–151, 2020.

[77] F. Cazals and T. Dreyfus, "The structural bioinformatics library: modeling in biomolecular science and beyond," *Bioinformatics*, vol. 33, no. 7, pp. 997–1004, 2017.

[78] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[79] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[80] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *International symposium on computer and information sciences*. Springer, 2005, pp. 284–293.

[81] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical review E*, vol. 76, no. 3, p. 036106, 2007.

[82] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics*, vol. 178, no. 1, pp. 13–23, 2009.

[83] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[84] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[85] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.

[86] R. A. Fisher, "On the interpretation of χ 2 from contingency tables, and the calculation of p," *Journal of the Royal Statistical Society*, vol. 85, no. 1, pp. 87–94, 1922.

[87] G. A. Barnard, "A new test for 2$\times$ 2 tables," *Nature*, vol. 156, no. 3954, p. 177, 1945.

[88] N. Akhter, W. Qiao, and A. Shehu, "An Energy Landscape Treatment of Decoy Selection in Template-free Protein Structure Prediction," *Computation*, vol. 6, no. 2, p. 39, 2018.

[89] K. L. Kabir, N. Akhter, and A. Shehu, "Connecting molecular energy landscape analysis with markov model-based analysis of equilibrium structural dynamics," in *Proceedings of 11th International Conference on Bioinformatics and Computational Biology (BICOB)*, vol. 60, 2019, pp. 181–189.

[90] ——, "From molecular energy landscapes to equilibrium dynamics via landscape analysis and markov state models," *Journal of bioinformatics and computational biology*, vol. 17, no. 6, p. 1940014, 2019.

[91] M. P. Harrigan, M. M. Sultan, C. X. Hernández, B. E. Husic, P. Eastman, C. R. Schwantes, K. A. Beauchamp, R. T. McGibbon, and V. S. Pande, "MSMBuilder: Statistical Models for Biomolecular Dynamics," *Biophysical journal*, vol. 112, no. 1, pp. 10–15, 2017.

[92] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, "PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models," *Journal of chemical theory and computation*, vol. 11, no. 11, pp. 5525–5542, 2015.

[93] Y. Naritomi and S. Fuchigami, "Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: the case of domain motions," *The Journal of chemical physics*, vol. 134, no. 6, p. 02B617, 2011.

[94] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, "Identification of slow molecular order parameters for markov model construction," *The Journal of chemical physics*, vol. 139, no. 1, pp. 015 102_1–13, 2013.

[95] A. Koneru, S. Satyanarayana, and S. Rizwan, "Endogenous opioids: their physiological role and receptors," *Global J Pharmacol*, vol. 3, no. 3, pp. 149–153, 2009.

[96] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[97] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011.

[98] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[99] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 353–362.

[100] J. Zhang and Y. Zhang, "A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction," *PloS one*, vol. 5, no. 10, p. e15386, 2010.

[101] H. Zhou and Y. Zhou, "Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction," *Protein science*, vol. 11, no. 11, pp. 2714–2726, 2002.

[102] Y. Yang and Y. Zhou, "Specific interactions for ab initio folding of protein terminal regions with secondary structures," *Proteins: Structure, Function, and Bioinformatics*, vol. 72, no. 2, pp. 793–803, 2008.

[103] M. Lu, A. D. Dousis, and J. Ma, "Opus-psp: an orientation-dependent statistical all-atom potential derived from side-chain packing," *Journal of molecular biology*, vol. 376, no. 1, pp. 288–301, 2008.

[104] S. Wang, W. Li, S. Liu, and J. Xu, "Raptorx-property: a web server for protein structure property prediction," *Nucleic acids research*, vol. 44, no. W1, pp. W430–W435, 2016.

[105] S. Miller, J. Janin, A. M. Lesk, and C. Chothia, "Interior and surface of monomeric proteins," *Journal of molecular biology*, vol. 196, no. 3, pp. 641–656, 1987.

[106] K. W. Plaxco, K. T. Simons, and D. Baker, "Contact order, transition state placement and the refolding rates of single domain proteins," *Journal of molecular biology*, vol. 277, no. 4, pp. 985–994, 1998.

[107] A. B. Zaman, P. V. Parthasarathy, and A. Shehu, "Using sequence-predicted contacts to guide template-free protein structure prediction," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2019, pp. 154–160.

[108] B. Boashash, *Time-frequency signal analysis and processing: a comprehensive reference*. Academic press, 2015.

[109] K. Wang, B. Fain, M. Levitt, and R. Samudrala, "Improved protein structure selection using decoy-dependent discriminatory functions," *BMC structural biology*, vol. 4, no. 1, pp. 1–18, 2004.

[110] *Protein Structure Prediction Center*, Last Accessed: October 21, 2021. [Online]. Available: https://predictioncenter.org/

[111] J. Cheng, M.-H. Choe, A. Elofsson, K.-S. Han, J. Hou, A. H. Maghrabi, L. J. McGuffin, D. Menéndez-Hurtado, K. Olechnovič, T. Schwede *et al.*, "Estimation of model accuracy in casp13," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1361–1377, 2019.

[112] L. A. Abriata, G. E. Tamò, B. Monastyrskyy, A. Kryshtafovych, and M. Dal Peraro, "Assessment of hard target modeling in casp12 reveals an emerging role of alignment-based contact prediction methods," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, pp. 97–112, 2018.

[113] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2005, pp. 1601–1608.

[114] K. Pelechrinis, *Spectral clustering with eigengap heuristic: A MATLAB implementation*, 2013 (Last Accessed: August 30, 2020). [Online]. Available: http://kokkodis.blogspot.com/2013/02/spectral-clustering-with-eigengap.html

[115] Y.-H. Kung, P.-S. Lin, and C.-H. Kao, "An optimal k-nearest neighbor for density estimation," *Statistics & Probability Letters*, vol. 82, no. 10, pp. 1786–1791, 2012.

[116] K. Huang, N. D. Sidiropoulos, and A. Swami, "Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 1, pp. 211–224, 2013.

[117] Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki, "Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2117–2131, 2011.

[118] D. Kuang, S. Yun, and H. Park, "Symnmf: nonnegative low-rank approximation of a similarity matrix for graph clustering," *Journal of Global Optimization*, vol. 62, no. 3, pp. 545–574, 2015.

[119] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011.

[120] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *Pattern recognition*, vol. 41, no. 4, pp. 1350–1362, 2008.

[121] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proceedings of the 2012 SIAM international conference on data mining*. SIAM, 2012, pp. 106–117.

[122] N. Akhter, R. Vangara, G. Chennupati, B. S. Alexandrov, H. Djidjev, and A. Shehu, "Non-negative matrix factorization for selection of near-native protein tertiary structures," in *International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 70–73.

[123] M. Karasikov, G. Pagès, and S. Grudinin, "Smooth orientation-dependent scoring function for coarse-grained protein quality assessment," *Bioinformatics*, vol. 35, no. 16, pp. 2801–2808, 2019.

[124] J. Zhang and D. Xu, "Fast algorithm for population-based protein structural model analysis," *Proteomics*, vol. 13, no. 2, pp. 221–229, 2013.

[125] S. Garcia and F. Herrera, "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons," *Journal of machine learning research*, vol. 9, no. Dec, pp. 2677–2694, 2008.

[126] K. L. Kabir, M. Bhattarai, B. Alexandrov, and A. Shehu, "Single model quality estimation of protein structures via non-negative tensor factorization," in *International Conference on Computational Advances in Bio and Medical Sciences*, 2021.

[127] K. Olechnovič, B. Monastyrskyy, A. Kryshtafovych *et al.*, "Comparative analysis of methods for evaluation of protein models against native structures," *Bioinformatics*, vol. 35, no. 6, pp. 937–944, 2019.

[128] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *ICML*, 2011.

[129] D. P. Truong, E. Skau, V. I. Valtchinov, and B. S. Alexandrov, "Determination of latent dimensionality in international trade flow," *Machine Learning: Science and Technology*, vol. 1, no. 4, p. 045017, 2020.

[130] F. Nüske, P. Gelß, S. Klus, and C. Clementi, "Tensor-based computation of metastable and coherent sets," *Physica D: Nonlinear Phenomena*, vol. 427, p. 133018, 2021.

[131] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[132] D. Krompass, M. Nickel, X. Jiang, and V. Tresp, "Non-negative tensor factorization with rescal," in *Tensor Methods for Machine Learning, ECML workshop*, 2013, pp. 1–10.

[133] M. Bhattarai, G. Chennupati, E. Skau, R. Vangara, H. Djidjev, and B. S. Alexandrov, "Distributed non-negative tensor train decomposition," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2020, pp. 1–10.

[134] R. Vangara, M. Bhattarai, E. Skau, G. Chennupati, H. Djidjev, T. Tierney, J. P. Smith, V. G. Stanev, and B. S. Alexandrov, "Finding the number of latent topics with semantic non-negative matrix factorization," *IEEE Access*, 2021.

[135] M. Bhattarai, B. Nebgen, E. Skau, M. Eren, G. Chennupati, R. Vangara, H. Djidjev, J. Patchett, J. Ahrens, and B. Alexandrov, "pydnmfk: Python distributed non negative matrix factorization," https://github.com/lanl/pyDNMFk, 2021.

[136] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[137] K. Uziela and B. Wallner, "Proq2: estimation of model accuracy implemented in rosetta," *Bioinformatics*, vol. 32, no. 9, pp. 1411–1413, 2016.

[138] K. Uziela, N. Shu, B. Wallner, and A. Elofsson, "Proq3: Improved model quality assessments using rosetta energy terms," *Scientific reports*, vol. 6, no. 1, pp. 1–10, 2016.

[139] K. Uziela, D. Menendez Hurtado, N. Shu, B. Wallner, and A. Elofsson, "Proq3d: improved model quality assessments using deep learning," *Bioinformatics*, vol. 33, no. 10, pp. 1578–1580, 2017.

[140] D. Menéndez Hurtado, K. Uziela, and A. Elofsson, "A novel training procedure to train deep networks in the assessment of the quality of protein models," 2019.

[141] K. L. Kabir, R. Nussinov, B. Ma, and A. Shehu, "Antigen binding reshapes antibody energy landscape and conformation dynamics," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2021, pp. 2519–2526.

[142] J. Zhao, R. Nussinov, and B. Ma, "Antigen binding allosterically promotes Fc receptor recognition," *MAbs*, vol. 11, no. 1, pp. 58–74, 2019.

[143] Y. Chen, G. Wei, J. Zhao, R. Nussinov, and B. Ma, "Computational investigation of Gantenerumab and Crenezumab recognition of Abeta fibrils in Alzheimer's disease brain tissue," *ACS Chem Neurosci*, vol. 11, no. 20, pp. 3233–3244, 2020.

[144] R. Nussinov and P. G. Wolynes, "A second molecular biology revolution? the energy landscapes of biomolecular function," *Phys Chem Chem Phys*, vol. 16, no. 14, pp. 6321–6322, 2014.

[145] A. D. McLachlan, "A mathematical procedure for superimposing atomic coordinates of proteins," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 28, no. 6, pp. 656–657, 1972.

[146] K. L. Kabir, N. Akhter, and A. Shehu, "Unsupervised learning of conformational states present in molecular dynamics simulation data for summarization of equilibrium conformational dynamics," *Biophysical Journal*, vol. 116, no. 3, pp. 291a–292a, 2019.

[147] F. Noé and F. Nuske, "A variational approach to modeling slow processes in stochastic dynamical systems," *Multiscale Modeling & Simulation*, vol. 11, no. 2, pp. 635–655, 2013.

[148] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, "Markov models of molecular kinetics: Generation and validation," *The Journal of chemical physics*, vol. 134, no. 17, p. 174105, 2011.

[149] L. Orellana, O. Yoluk, O. Carrillo, M. Orozco, and E. Lindahl, "Prediction and validation of protein intermediate states from structurally rich ensembles and coarse-grained simulations," *Nat Commun*, vol. 7, p. 12575, 2016.

[150] R. Clausen, B. Ma, R. Nussinov, and A. Shehu, "Mapping the conformation space of wildtype and mutant H-Ras with a memetic, cellular, and multiscale evolutionary algorithm," *PLoS Comput Biol*, vol. 11, no. 9, p. e1004470, 2015.

[151] W. Qiao, N. Akhter, X. Fang, T. Maximova, E. Plaku, and A. Shehu, "From mutations to mechanisms and dysfunction via computation and mining of protein energy landscapes," *BMC Genomics*, vol. 19, p. 671, 2018.

[152] W. Humphrey, A. Dalke, and K. Schulten, "VMD - Visual Molecular Dynamics," vol. 14, no. 1, pp. 33–38, 1996, http://www.ks.uiuc.edu/Research/vmd/.

Curriculum Vitae

Kazi Lutful Kabir is currently working towards the completion of his Ph.D. degree in Computer Science (CS) at George Mason University. He received his B.S. degree in Computer Science and Engineering (CSE) from Military Institute of Science and Technology (MIST), Dhaka, Bangladesh in 2014. He also obtained an M.S. degree in CS from George Mason University, Fairfax, VA, USA in 2019. In fact, He is completing his Ph.D. in the summer of 2022. Before starting his Ph.D., he served as a lecturer (teaching faculty) in BRAC University, Dhaka, Bangladesh for 1.5 years. Prior to that, he was a lecturer in MIST, Dhaka, Bangladesh for 2 years. His research interests include machine learning and data mining with applications in structural bioinformatics. Recently his works are focused on the applications of unsupervised learning techniques for organizing high-dimensional spatial structure data corresponding to tertiary structures of biomolecules.

Education

- Master of Science, George Mason University, 2019.
- Bachelor of Science, Military Institute of Science and Technology, 2014.

Awards

- Applied Machine Learning (AML) Research Fellowship, Los Alamos National Laboratory, 2021.
- Computational Sciences Graduate Internship, Physics and Chemistry of Materials Group, Theoretical Division (T-1), Los Alamos National Laboratory, 2021.
- Applied Machine Learning (AML) Research Fellowship, Los Alamos National Laboratory, 2020.
- Best Paper Award, International Conference on Bioinformatics and Computational Biology (BICOB), 2019.
- Summer Research Initiation Award, Department of Computer Science, George Mason University, 2018.
- IEEE EMBS Bangladesh Chapter Best Paper Award, International Conference on Medical Engineering, Health Informatics and Technology (MediTec), 2016.

Selected Publications

Journal Publications

- **Kazi Lutful Kabir**, Ruth Nussinov, Buyong Ma, and Amarda Shehu. "Fewer Dimensions, More Structures for Improved Discrete Models of Dynamics of Free Versus Antigen-Bound Antibody." Biomolecules, 2022 (under review).
- Nasrin Akhter, **Kazi Lutful Kabir**, Gopinath Chennupati, Raviteja Vangara, Boian Alexandrov, Hristo N. Djidjev, and Amarda Shehu. "Improved protein decoy selection via non-negative matrix factorization." IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2021.
- **Kazi Lutful Kabir**, Nasrin Akhter, and Amarda Shehu. "From molecular energy landscapes to equilibrium dynamics via landscape analysis and markov state models." Journal of Bioinformatics and Computational Biology, 2019.
- Nasrin Akhter, Gopinath Chennupati, **Kazi Lutful Kabir**, Hristo Djidjev, and Amarda Shehu. "Unsupervised and supervised learning over the energy landscape for protein decoy selection." Biomolecules, 2019.
- **Kazi Lutful Kabir**, Nasrin Akhter, and Amarda Shehu. "Unsupervised Learning of structural States Present in Molecular Dynamics Simulation Data for Summarization of Equilibrium structural Dynamics" Biophysical Journal, 2019.
- **Kazi Lutful Kabir**, Liban Hassan, Zahra Rajabi, Nasrin Akhter, and Amarda Shehu. "Graph-based community detection for decoy selection in template-free protein structure prediction." Molecules, 2019.
- Md Kishwar Shafin, **Kazi Lutful Kabir**, et al. "Impact of Heuristics in Clustering Large Biological Networks." Computational Biology and Chemistry, 2015.

Conference and Workshop Publications

- **Kazi Lutful Kabir**, Manish Bhattarai, Boian S. Alexandrov, Amarda Shehu. "Single Model Quality Estimation of Protein Structures via Non-negative Tensor Factorization." International Conference on Computational Advances in Bio and medical Sciences (ICCABS 2021).
- **Kazi Lutful Kabir**, Ruth Nussinov, Buyong Ma, Amarda Shehu. "Antigen Binding Reshapes Antibody Energy Landscape and Conformation Dynamics." IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2021) Workshop XIII: Computational Structural Biology Workshop (CSBW 2021).
- **Kazi Lutful Kabir**, Gopinath Chennupati, Raviteja Vangara, Hristo Djidjev, Boian S. Alexandrov, and Amarda Shehu. "Decoy selection in protein structure determination via symmetric non-negative matrix factorization." IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2020).

- **Kazi Lutful Kabir**, Nasrin Akhter, and Amarda Shehu. "Connecting Molecular Energy Landscape Analysis with Markov Model-based Analysis of Equilibrium Structural Dynamics." International Conference on Bioinformatics and Computational Biology (BICOB 2019). ***Best Paper Award***
- Wali Mohammad Abdullah, **Kazi Lutful Kabir**, et al. "An Improved Chromosome Sorting Algorithm by Permutation Group-based Inverted Block-interchanges." International Conference on Medical Engineering, Health Informatics and Technology (MediTec 2016). ***IEEE EMBS Bangladesh Chapter Best Paper Award***
- Md Kishwar Shafin, **Kazi Lutful Kabir**, et al. "New Heuristics for Clustering Large Biological Networks." International Symposium on Bioinformatics Research and Applications (ISBRA 2015).