Greedy-proximal A* and Hybrid Spectral/Subspace Clustering for Molecular Dynamics Simulations

By

Ivan Syzonenko

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

in

Computational Sciences

Middle Tennessee State University

December 2019

Dissertation Committee:

Dr. Joshua L. Phillips, Chair

Dr. Chrisila C. Pettey

Dr. Anatoliy Volkov

Dr. Justin Miller

DEDICATION

This dissertation is dedicated to my wife Svitlana whose faith in my success, support, and encouragement inspired me to pursue and complete this research.

ACKNOWLEDGEMENTS

I would like to thank Dr. Chrisila Pettey for initial acceptance to the graduate Computer Science program.

I would like to thank Dr. John Wallin for the opportunity to study as a part of the Computational Science Ph.D. program and for providing an assistantship. I would like to thank Dr. Joshua L. Phillips for providing knowledge about Molecular Dynamics and protein folding.

I would like to thank Dr. Joshua L. Phillips, Dr. John Wallin, and Dr. Anatoliy Volkov for providing access to computational resources at MTSU. I would like to thank Dr. Justin Miller for advice about the experiment setup. I would like to thank Dr. Anatoliy Volkov for numerous suggestions on how to improve the manuscript.

ABSTRACT

Protein folding plays a crucial role in human biochemistry. Proteins are the building blocks for most of our tissues, help to transfer signals through the bloodstream and other fluids, and help to cure diseases. On the opposite side, pathogens and viruses also consist of proteins, which make our understanding of protein function a top priority to save and prolong human life. Even small changes in folding patterns may lead to serious diseases like Alzheimer's or Parkinson's where proteins are folded either too quickly or too slowly.

The protein folding problem has been studied in the field of molecular biophysics for many years (Maximova et al., 2016; Scheraga et al., 2007), however many questions are still unanswered. Mainly they are: "what is the final conformation (3-dimensional shape or structure) given the primary sequence of amino acids? ", "how does the conformation change over time?", and "how does a protein's secondary and tertiary conformation affect its functions?". Molecular dynamics (MD) is one of the tools used to understand how proteins fold into native conformations (Chen et al., 2008). It uses computational techniques to calculate the interactions of molecules. While it captures sequences of conformations that lead over time to the folded state, limitations in simulation timescales remain problematic (Klepeis et al., 2009). One of the limitations is the notion of "energy wells" (Liu et al., 2012) - conformations with low potential energy which reduce the probability to form other conformations and finally fold (reach the global minimum of the potential energy) within a computationally feasible timescale. The complete set of energies for all possible conformations is called the energy landscape (Liu et al., 2012; Wales, 2003). Although many approaches have been suggested to speed-up the simulation process using rapid changes in temperature or pressure, we propose a rational approach, Greedy-proximal A* (GPA*), derived from path finding algorithms to explore the supposed shortest-path folding pathway. Such an algorithm should not

only reduce the computational time needed to obtain the folded conformation without adding artificial energy bias, but also make it possible to generate trajectories which contain minimal motions needed for the folding transition. We introduce several new protein structure comparison metrics based on the contact map distance to help mitigate the challenges faced by "standard" metrics. We test our approach on proteins which represent the two main types of secondary structure: *a*) the Trp-Cage Miniprotein Construct TC5b (1L2Y) (Neidigh et al., 2002) which is a short, fast-folding protein that represents alpha-helical secondary structure formed because of a locked tryptophan in the middle, *b*) the immunoglobulin binding domain of streptococcal protein G (1GB1) (Gronenborn et al., 1991), containing an alpha-helix and several beta-sheets and *c*) the chicken villin subdomain HP-35, N68H protein (1YRF) (Chiu et al., 2005) - one of the fastest folding proteins which forms three alpha-helices. We compare our algorithm to Replica-Exchange Molecular Dynamics (REMD) and Steered Molecular Dynamics (SMD) methods which represent the main algorithms used for accelerating folding proteins with MD.

Another common application of MD simulations is for future experimental validation and energy landscape exploration for studying metastable conformations and the transitions between them (Phillips, 2012; Bowman and Pande, 2010). While the problem of capturing metastable states may often be successfully resolved within the timescale of the simulation, finding those states is often performed with automated techniques such as clustering (Bhowmik and Ramanathan, 2018; Sittel and Stock, 2016). Although there are many clustering algorithms available, not all of them can be successfully applied to high-dimensional data such as MD simulations (Steinbach et al., 2004). In particular, recent work from the clustering literature (Sakuraba et al., 2010) shows that many high-dimensional data sets explore a mixture of independent subspaces and previous clustering studies of MD data have ignored such effects. In this study we explore the

application of subspace clustering techniques to MD simulation data and compare their performance with traditional Spectral clustering (SPC) algorithms (Ng et al., 2002) and demonstrate when and why such approaches may be superior to traditional techniques.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

1GB1 immunoglobulin binding domain of streptococcal protein G. v, 43, 71–73, 107, 115–118

1L2Y Trp-Cage Miniprotein Construct TC5b. v, 34, 43, 45, 46, 50, 52, 63, 70–74, 85, 107, 115–118, 125, 126

1YRF chicken villin subdomain HP-35, N68H protein. v, 43, 50, 52, 70–73, 79

5EQJ NSP1 protein, tRNA m1A58 methyltransferase. 108, 115–118, 124–126

AARMSD all atom RMSD. 25, 37, 49, 52, 74, 79, 86

AMBER AMBERff99SB-ILDN force field. 43, 45, 46, 63, 71, 108

AND contact map distance agreement. 35, 70–73

ANDH hydrogen bonds contact map distance agreement. 35, 70, 71, 73

ANGL dihedral angle distance. 34, 35, 63, 70–72, 74

BASH Bourne again shell. 39, 44

BBRMSD backbone RMSD. 25, 37, 49, 52, 53, 74, 86

CHARMM CHARMM36-nov2018 force field. 43, 71

DN Dense. 109

DOF degrees of freedom. 12, 13, 29, 91, 108

GPA* Greedy-proximal A*. iv, 41, 43, 45, 46, 48–50, 53, 70, 76, 79–81, 84, 85, 88, 89, 130, 132

GROMACS GROningen MAchine for Chemical Simulations. 39–41, 43–45, 108

GROMOS GROMOS54a7 force field. 43, 52, 53, 63, 70–74

GSF Gaussian similarity function. 93, 96, 109

IDP intrinsically disordered proteins. 107, 108, 116–119, 124, 129

KNN k-nearest neighbors. xxviii, 93, 109, 110, 120, 123, 124

MC Monte Carlo method. 7

MD Molecular dynamics. iv–vi, 1, 3, 4, 6, 7, 9, 12, 13, 15–17, 20, 29–31, 33, 37, 39–43, 45, 46, 50, 76, 80, 88, 90, 91, 96, 100, 101, 107, 129, 130

MDSCTK Molecular Dynamics Spectral Clustering Toolkit. 109

NFP natively folded proteins. 107, 115–119, 124–126, 129

NMI normalized mutual information. 100, 101, 103, 109–112, 115–118, 123–126, 129

NMR nuclear magnetic resonance. 29, 37, 43, 46, 49, 50, 52, 71, 74, 76, 79, 81, 84, 88, 130, 132

OPLS OPLSaa force field. 43, 52, 63, 70, 71

PCA principal components analysis. 95

PDB Protein Data Bank. 37, 43, 53

REMD Replica-Exchange Molecular Dynamics. v, 6, 16–19, 45, 46, 49, 76, 79, 81, 88, 130

RMSD Root-mean-squared deviation. 25, 27, 31, 34, 35, 37, 44–46, 48, 49, 52, 53, 63, 70–72, 74, 76, 79, 81, 84–86, 88, 130

SDS dot product. 103, 116–119, 123–126, 129

SES element-wise product. 103, 116–119, 123–125, 129

SMD Steered Molecular Dynamics. v, 17, 19, 35, 46, 49, 84, 85

SP Sparse. 109, 117

SPC Spectral clustering. vi, 91, 103, 115–119, 123–125, 129

SS Super-sparse. 109

SSC Subspace Clustering. 99, 103, 115–118, 129

SVD singular value decomposition. 94, 103

XOR contact map distance disagreement. 35, 70, 71, 73

YJM1418 nucleoporin NUP116p protein. 108, 115–118, 124–126

CHAPTER I.

INTRODUCTION

Molecular modeling and simulation are modern techniques to study proteins. Researchers have many questions about each protein; it's movement, folding, interactions with other proteins, etc. However, answering them takes too much time and resources experimentally. Molecular dynamics (MD) (Scheraga et al., 2007) helps to obtain the answers faster by using computational techniques to perform computer simulations of the studied proteins. While increased computational power leads to decreased time needed to perform each simulation, scalability is still a problem (Balasubramanian et al., 2016). Additionally, postprocessing of the simulation may take the same or even more time than the simulation itself. Furthermore, faster algorithms allow more simulations to be performed on the same hardware. Here we propose an enhanced sampling technique which takes inspiration from the artificial intelligence field and helps to perform more efficient MD simulations which use *a*) less computational resources and *b*) result in trajectories which are easier to understand. While there are algorithms that try to reduce the time needed to obtain solutions, usually they introduce an unnatural bias in the simulation. Biasing the computational experiment may lead to biased results and wrong conclusions.

As mentioned above, postprocessing plays an important role in the analysis of the simulation. The ability to separate trajectories into groups helps to find common qualities and often avoid typical problems. Additionally, finding parts of different trajectories tightly related to each other may unveil patterns which are easy to miss by visual analysis of the trajectories. However, separation of the trajectories is not an easy process, since they often lie in a multidimensional nonlinear space due to the possible motions of the protein. While there exists many clustering algorithms, they either *a*) can cluster nonlinear data or *b*) can cluster data that lies in the multiple subspaces. Therefore, here we also propose a clustering algorithm that takes inspiration from the Machine Learning field and helps to improve the quality of the trajectory analysis.

CHAPTER II.

FOLDING

INTRODUCTION

The protein folding problem has been studied in the field of molecular biophysics for many years (Maximova et al., 2016; Scheraga et al., 2007), however many questions are still unanswered. Some examples are: "what determines the final conformation (3-dimensional shape or structure) given the primary sequence of amino acids? (Dill and MacCallum, 2012) ", "how does the conformation change over time?", and "how does a protein's secondary and tertiary conformation affect its functions?".

Information about the protein folding pathway lies at the root of basic protein science and many modern technologies like disease prevention and treatment (Selkoe, 2003), manipulation of plants and animals (Yon, 2001), bio-fuels discovery and improvement (Sticklen, 2008), generation and study of new proteins and their interactions (Gidalevitz et al., 2006). Molecular dynamics (MD) simulation complements experimental studies in these domains by providing *in silico* hypothesis testing and mechanistic explanations (Karplus and Kuriyan, 2005). It is also one of the most prominent tools used to understand how proteins fold into native conformations (Chen et al., 2008). MD uses computational techniques to compute the interactions of molecules that can be subsequently validated through lab experiments (van Gunsteren et al., 2018; Bottaro

et al., 2018). In particular, MD captures sequences of conformations that lead over time to the folded state, but limitations in simulation timescales remain problematic (Klepeis et al., 2009). One of the limitations is the notion of "energy wells" (Liu et al., 2012) - conformations with low potential energy which reduce the probability to form other conformations and finally fold (reach the global minimum of the potential energy) within an expected timescale. The complete set of potential energies of all possible conformations is called the energy landscape (Liu et al., 2012; Wales, 2003).

Although many approaches have been suggested to speed-up the simulation process, for example, by using rapid changes in temperature or pressure or attaching a virtual spring which forces the unfolded protein to it's folded conformation, all of them add artificial energy bias which affects the shape of the energy landscape thus distorting the natural folding sequence of conformations. Having a method which speeds up the process without such bias would be beneficial.

Additionally, having the shortest possible sequence of events would greatly help scientists to understand how particular mutations change the folding sequence. We also think that the shortest pathway for the folding protein can potentially be the most probable way of folding. A potential application of this approach would be for example, finding mutants that can be used in biofuel production to speed up the cellulose degradation process. Furthermore, information about folding may help to study the misfolding behavior which is at the root of diseases like Alzheimer's, Parkinson's, Huntington's, and many more. Finally, general improvement in the performance of folding trajectory generation along with efficient usage of modern computer hardware would reduce the time and resources spent for research.

BACKGROUND

Molecular Dynamics

Molecular dynamics (MD) is a computer simulation that computes the physical motions of atoms and molecules based on a numerical solution of Newton's equation of motion where forces between particles are usually computed with molecular mechanics force fields (Rappé et al., 1992; Maple et al., 1988). It is a complex process that depends on both experimental setup (salinity, concentration, temperature, pressure, initial conformation, etc.) and on simulation rules (force fields, water models, thermostat or barostat models, initial velocities, atomic charges, etc.). But even having all of them set perfectly under ideal conditions, the folding process is not guaranteed to complete in any finite time (i.e. within the timescale accessible to MD simulation) since the protein may fall into a local minimum energy well and keep it's nonnative conformation (Stefani, 2008; Daidone et al., 2003; Bernardi et al., 2015). While the mathematical equations are not very complex, every iteration step requires the computation of forces on each particle which may result in long computational time since typical simulation consists of billions of steps or/and protein size may reach thousands of amino acids. Since timesteps are at most femtoseconds long (due to the hydrogen bond vibration frequency (Kühn and Wöste, 2007)), overall simulation length is typically bound to the millisecond timescale on even the largest computing clusters (Klepeis et al., 2009). As mentioned above one of the attributes of the MD approach is the notion of force fields which model the forces between atoms. While force fields are parameterized to aim for the best representation of specific molecular interactions, interactions which were not taken into account may be represented differently, which introduces an artificial bias in the folding conformation sequence (Freddolino et al., 2009). Due to these difficulties, there are many modified MD protocols/algorithms that are designed to reduce time spent on computing the

folding trajectories. Popular approaches are Replica-Exchange Molecular Dynamics (REMD) (Sugita and Okamoto, 1999), Targeted MD (Izrailev et al., 1999), and pressure jump (P-jump) MD (Liu et al., 2014; Wirth et al., 2015).

We also have to mention the Monte Carlo method (MC) simulation technique (Metropolis and Ulam, 1949) which uses MC sampling approach to guess native conformations (Carnevali et al., 2003), sequences of conformations (Kolinski and Skolnick, 1994; Hoffmann and Knapp, 1996) that can be viewed as a trajectory, or even protein-protein interactions (Kawai et al., 1989). The MC method generates a large number of states much faster than the MD method (Ulmschneider et al., 2006) since energy can be computed directly from the force field's equations (Ulmschneider et al., 2006). However its main drawback lies in the simplifications that the MC approach usually includes: usage of implicit solvent (Ulmschneider et al., 2006), limiting motion of the protein's backbone only (Ulmschneider et al., 2006), criterion of the acceptance/rejection of the move (Metropolis and Ulam, 1949). There are also successful attempts (Peter and Shea, 2017) to combine MD and kinetic MC methods by using distance-dependent fluctuations derived from dissipative particle dynamics (Groot and Warren, 1997; Espanol and Warren, 1995) as a part of the random walk process (Peter and Shea, 2017).

Protein Secondary structure

One of the main aspects of the protein folding problem is a prediction of the three-dimensional conformation of the protein based on its amino acid sequence and discovery of a set of conformations that lead to the native conformation from the initial unfolded conformation (Dill and MacCallum, 2012).

All proteins are essentially poly-peptides and consist of one or more chains where the main building blocks are amino acids. Each amino acid consists of an amino group and a carboxyl group held together by a carbon (alpha-carbon). There are roughly 20

amino acids use by most organisms to construct proteins. The only difference between all amino acids is the side chain (R-group) which is connected to the alpha-carbon. The simplest example of an amino acid can be found in Figure 1. The sequence of connected amino acids that make up a protein is called the primary structure. The

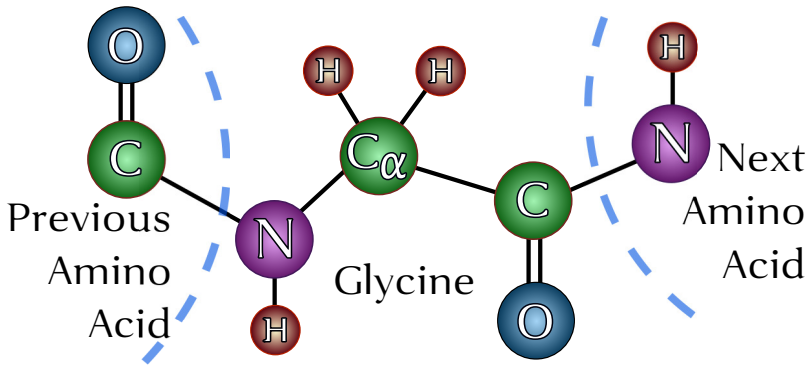


Figure 1: Glycine, the simplest amino acid, drawn as a part of a polypeptide.

backbone of the peptide corresponds to the chain of $N - C_\alpha - C$ atoms formed along by the primary structure. Different properties (hydrophilicity, charges, etc.) of the R-group dictate interactions between the amino acids which often result in common patterns in the shape of the backbone, called a secondary structure. The most typical secondary structures are alpha-helices, beta-pleated sheets, and coils (absence of regular secondary structure) (Kabsch and Sander, 1983). However, there other structures like 3_{10} helix, π helix, beta-bends, polyproline helix, alpha-sheet, etc (Kabsch and Sander, 1983). All of the interactions are based on hydrogen bonds, which can be formed and/or dismissed before the final folded conformation is achieved. Figure 2 demonstrates an example of alpha-helical structure (colored region). It is common to use a cartoon representation of the alpha-helical structures (as shown in Figure 2) which are stabilized by hydrogen bonds formed between every fourth amino acid along the helix. Beta-pleated sheets are often represented as sheet-like structure (as shown in Figure 3) which is stabilized by hydrogen bonds formed between carbonyl and amino groups of backbone. Figure 3 demonstrates a beta-pleated sheet structure (parallel and anti-parallel).

Besides the backbone interactions, polypeptide chains also experience hydrophilic/hydrophobic interactions between R groups, which help to stabilize the structure. Secondary structures often come together (for example, alpha-helix is positioned near the particular beta-sheet) due to these interactions to form tertiary structure. The protein shown in Figures 2 and Figures 3 is a good example of tertiary structure where an alpha-helix and two beta-sheets form the protein's particular conformation - the R groups of the amino acids, which usually point outward from the alpha helix, interact with beta-pleated sheets.

Quaternary structure, also known as protein-protein docking, is essentially a tertiary structure but formed between two or more different polypeptide chains.

Protein structure prediction can be divided into the following sub-problems: secondary structure prediction, tertiary structure prediction, and quaternary structure prediction (Huang and Zou, 2010). Structure prediction given only a primary sequence, referred as 'ab initio' protein structure prediction, is typically solved in two ways: 1) "template-based-modeling" (Šali and Blundell, 1993; Karplus et al., 1998; Yang et al., 2011), when similar 'solved' protein used as a template, or 2) 'ab initio' modeling (Klepeis et al., 2005), de novo modeling (Bradley et al., 2005a,b), physics-based modeling (Ołdziej et al., 2005), or free modeling (Jauch et al., 2007) in which the main idea is either a coarse generation of conformations and testing them, or application of MD-like approaches. While MD is not the main tool for structure prediction, it can be used to generate a sequence of conformations which lead to the predicted structure, thus providing mechanistic details on the folding process as well.

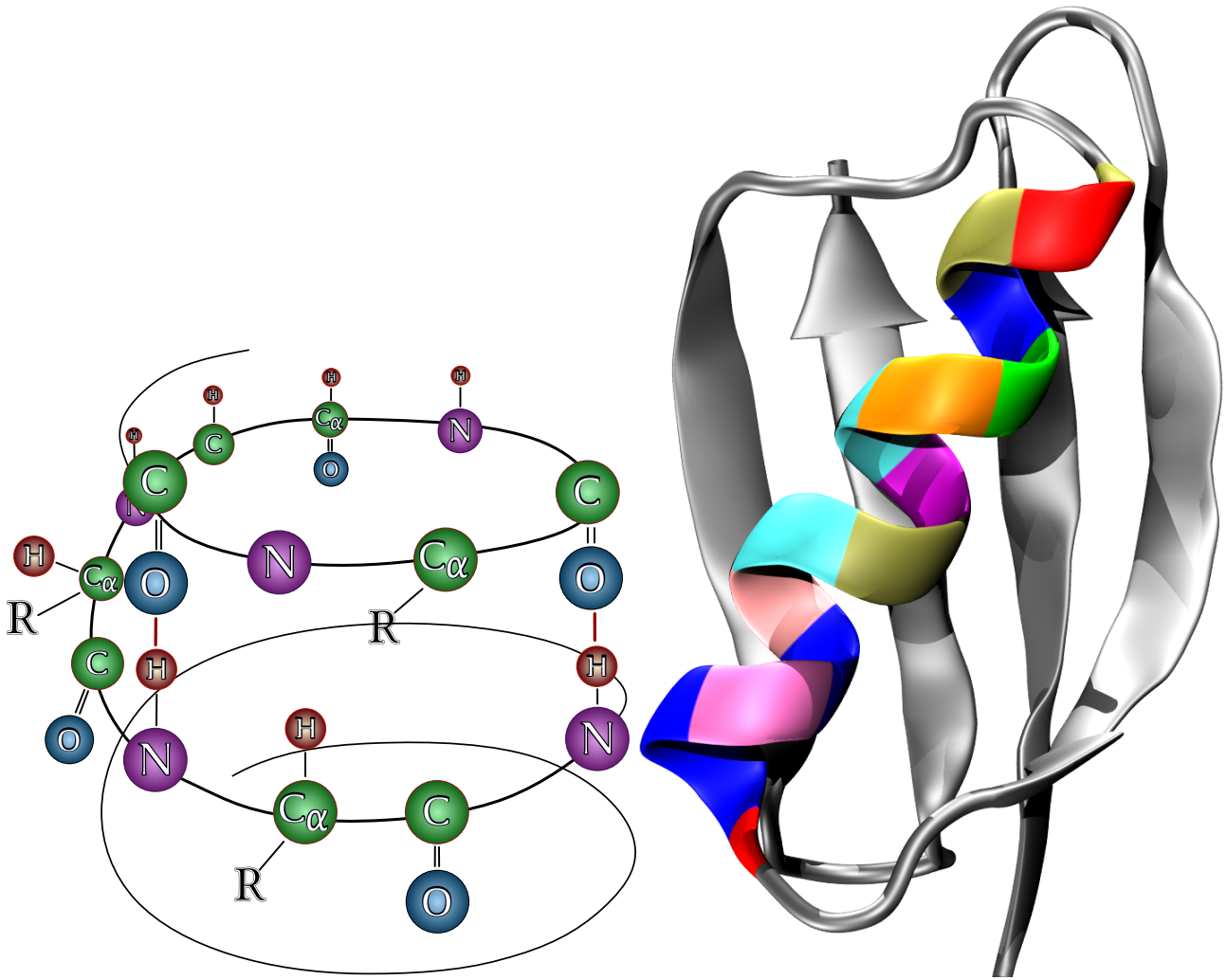


Figure 2: Example of the alpha-helical protein secondary structure. Backbone chain is connected with hydrogen bonds to itself. R - side chain (R group), specific to each amino acid, but does not matter in the secondary structure representation. A - stereochemical representation, B - visualization representation.

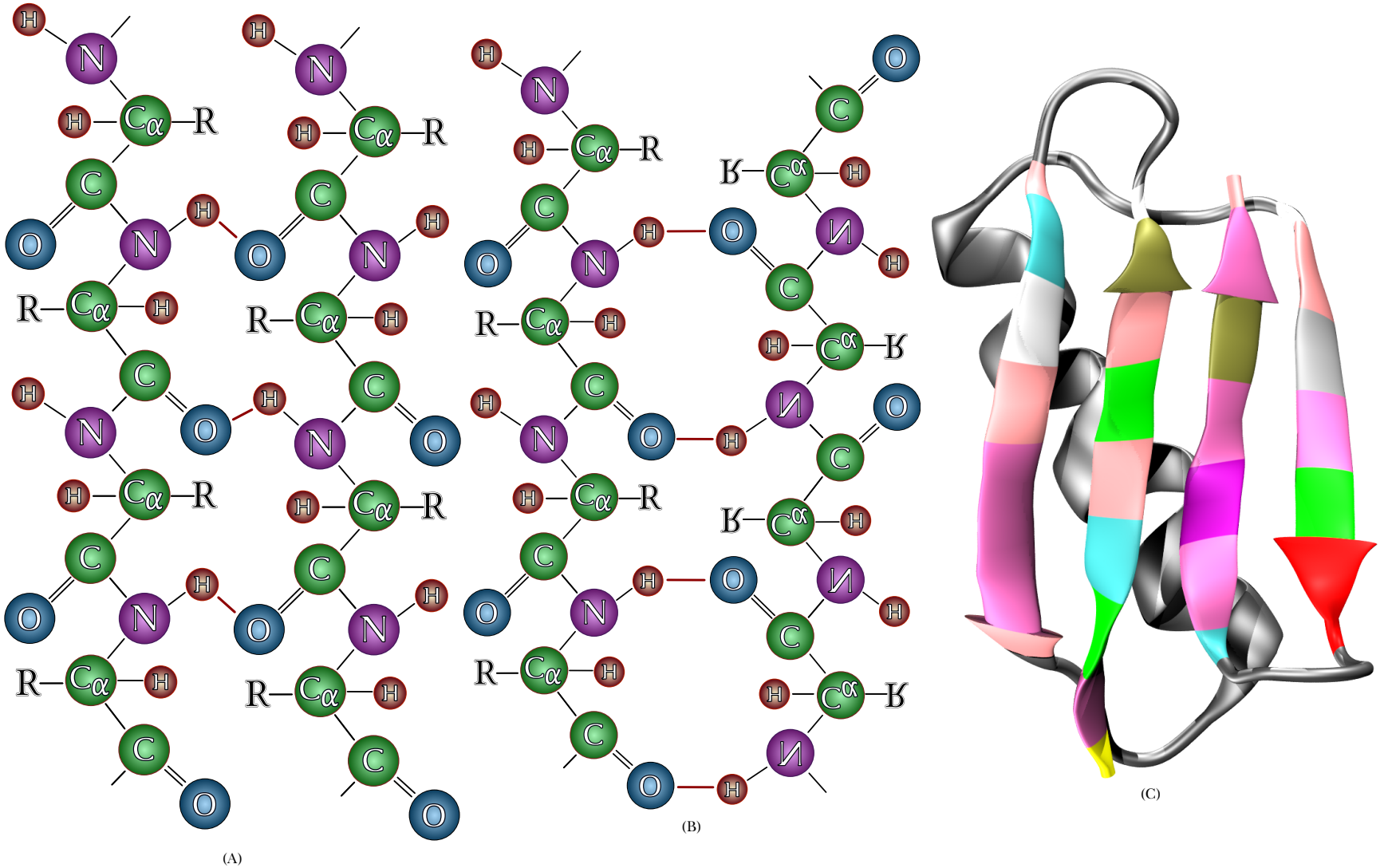


Figure 3: Example of the reverse beta-sheet protein secondary structure. Two backbone chains connected with hydrogen bonds (red lines). R - side chain (R group), specific to each amino acid, but does not matter in the secondary structure representation. A, B - stereochemical representation of straight and reversed beta-sheet, C - visualization representation.

Transition states

Assuming we introduce reasonable force fields, water models and other parameter values, the next problem that a typical MD application encounters is energy barriers (Wales, 2003). That is, the motion of the protein is defined by the interaction of all molecules in a simulated volume, and only certain degrees of freedom (DOF) are available due to attraction/repulsion forces defined by the state of the system. Such a limitation in DOF dictates that there can be no straight path from the unfolded to the folded state. Instead the protein has to change gradually exploring available DOF. Bottleneck conformations (dictated by energy barriers and thus limited DOF) needed to pass from one low-energy state to another state in the folding process are called transition states.

The folding process is often associated with changes in potential energy and the Gibbs free energy. The folded state is often associated with the minimum of the energy landscape which is based on changes of the potential energy. The Gibbs free energy ΔG described by the following equation:

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

where ΔH - is the change in enthalpy, ΔS - is the change in entropy, and T is the temperature (Steinberg and Scheraga, 1963; Karplus et al., 1987). The folded state typically has the lowest potential energy and low number of DOF.

Figure 4 demonstrates an example of protein folding. Note that to get from the initial state (red circle) to the folded state (green circle) the protein has to increase it's free energy to pass the energy barriers. Additionally, high enough kinetic energy may help protein to avoid becoming trapped in some of the energy wells, it also can increase its chances to jump back to a more unfolded conformation. An example of such an event is described in Figure 5 and Figure 7 on page 16. Figure 5 shows an example where the probability (p_2) of passing the energy barrier 2 is lower than probability p_1 of passing

the barrier 1. However, in Figure 5 we also see that once the protein jumps over the higher energy barrier, the probability of jumping back would be very small in contrast to Figure 7 which shows an altered energy landscape due to higher temperature and thus higher kinetic energy.

In practice, MD is significantly simpler when compared to quantum mechanical calculations for atoms and molecules. While force fields are partially derived from quantum mechanics, the general approximation of the MD approach may create artificial barriers and/or decrease influence of the natural barriers.

Increase of kinetic energy and access to more DOF approaches lie in the roots of many methods that bias (artificially warp the energy landscape) the energetics of the system which statistically increases the chances of quickly passing through such regions on the energy surface (Wales, 2003).

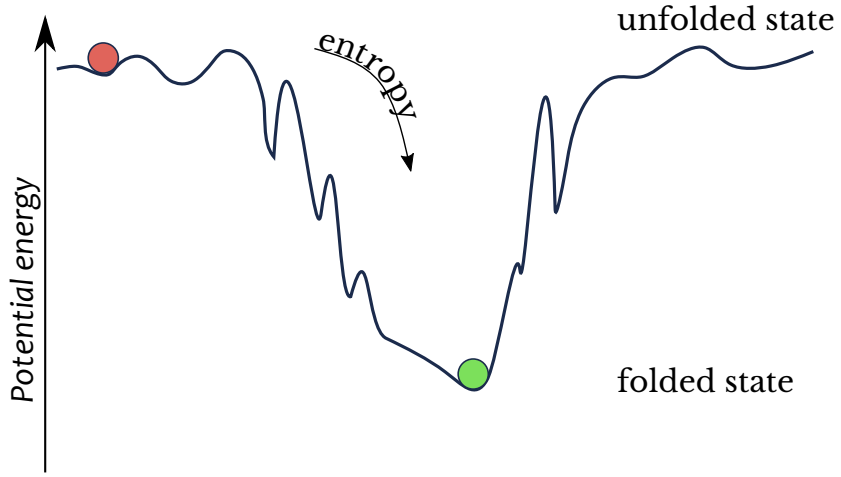


Figure 4: Representation of the folding process from the energy landscape viewpoint. Red circle represents unfolded conformation with higher potential energy, green circle represents folded conformation with lower potential energy and entropy.

Enhanced Sampling Approaches

In this section we will widely use the term 'sampling'. In MD, sampling means obtaining a collection of protein structures. MD samples new conformation as the

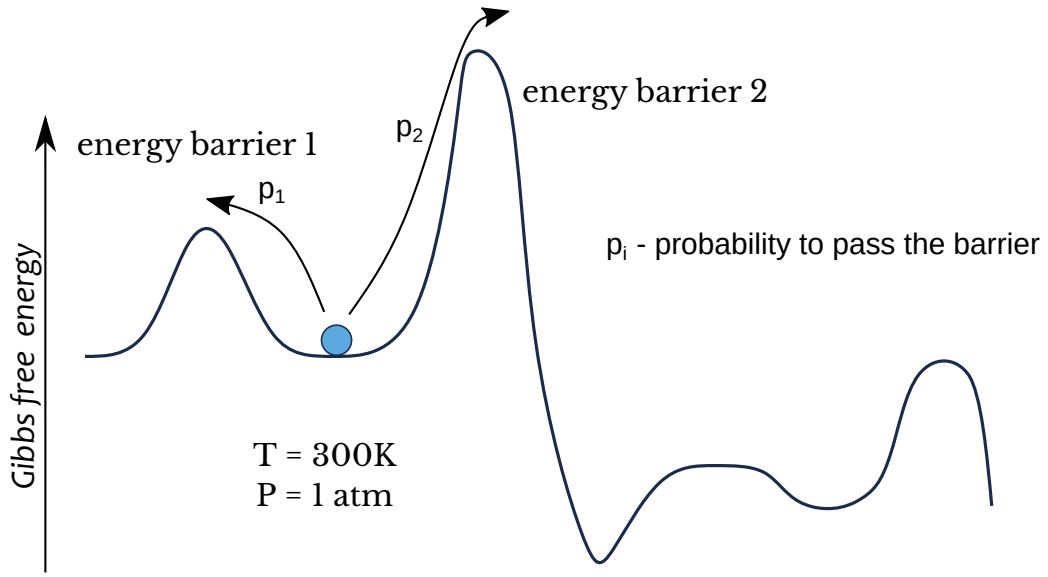


Figure 5: Representation of the energy barrier. Blue circle represents protein conformation at particular state, p1 and p2 - represent probabilities of passing barriers 1 and 2 respectfully, under the constant room temperature and pressure.

simulation proceeds. Enhanced sampling includes efforts to speed up the sampling rate of these new conformations.

Multi-replicate Molecular Dynamics

This is the simplest sampling approach based on the notion that given enough independent trajectories, with different initial velocities, but constant in all other parameters, the union of these trajectories will cover most of the energy landscape and allow a certain number of trajectories to avoid energy barriers. A validation of such an approach was proposed by Duan and coworkers (Chowdhury et al., 2004). The main motivation for this method is that multiple independent trajectories improve the chances of sampling the protein folding event. Since trajectories do not depend on each other, they can be run in parallel, thus, with sufficient computing power, reducing simulation time to only the length of a single folding trajectory. However, the folding time of even simple

proteins may still take a very long time, even on a modern computer hardware, since the fastest folding trajectories are rarely found.

Simulated annealing

Other approaches introduce bias in the energy landscape to enhance sampling. One of the simplest examples of such an approach is simulated annealing, extensively studied by Car et al. (Car and Parrinello, 1985). It follows a temperature profile during the simulation guided by the notion that higher temperatures increase the likelihood to overcome energy barriers. This notion was validated (Callender and Dyer, 2002) by using a laser as a source of rapid temperature change. There are two main approaches to design the temperature profile (sequence of temperature changes) : the first one is to follow possible natural temperature changes, and the second one, more typical, is to have several rapid heating and slow cooling events, additionally guided by the application of stochastic methods. The first type of temperature schedule has the advantage of not adding artificial energy bias, but may not help with energy barriers caused by artificial sources. The second type of a temperature schedule, when designed perfectly, would result in a quick folding process, but it would also introduce artificial bias which cannot happen inside a natural environment. Additionally, imperfections in the temperature schedule may result in unfolding of the protein when the temperature is too high, or almost no movement when the protein falls into a low-energy well and the temperature is too low to overcome it.

P-jump

Wirth et al. (Wirth et al., 2015) suggested an interesting approach, called P-jump, which consists in rapid increase of the pressure, which denatures the protein, followed by returning pressure to the initial values. While it showed agreement between physical and MD approaches, natural processes rarely have such drastic pressure changes, thus

the introduced bias cannot be natural. Figure 6 demonstrates change of the pressure inside the simulation box which allows protein to fold faster.

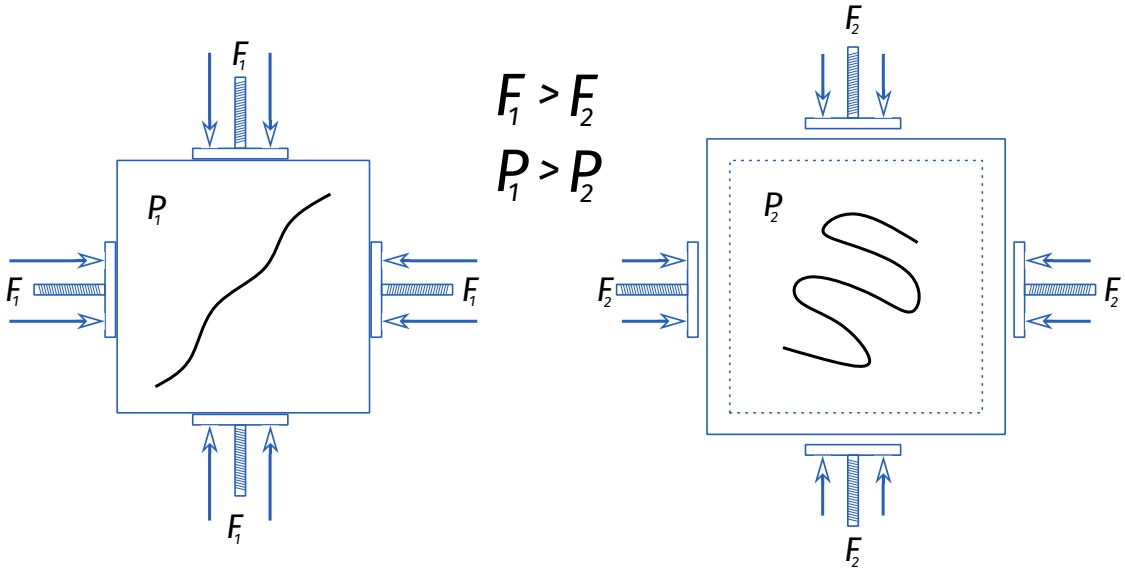


Figure 6: Representation of the P-jump process. Left square represents a simulation volume with regular pressure P_1. Right square represents a simulation volume with reduced pressure P_2 which allows protein to take a favorable conformation.

Replica-Exchange Molecular Dynamics

REMD is currently the most commonly used tool to study protein folding landscapes because of its ability to overcome energy barriers, and it is also highly parallelizable (Eleftheriou et al., 2006). It combines simulated annealing and multi-replicate MD and consists of running multiple MD simulations at different temperatures for a short time (defined prior to the simulation). Once the simulations are complete, temperatures are exchanged between simulations and another stage of simulation is performed. Exchange itself happens such that trajectories with higher potential energy of the protein are moved to the simulation setup with higher temperature (kinetic energy) in hopes to more easily overcome the energy barrier while trajectories with lower potential energy of the protein are moved to the simulation setup with lower temperature in order reduce the probability of jumping back to a previously found low-energy state. Such a behavior is very

similar to the Metropolis criterion (Gustafson, 1998), however, there is no probability to perform the exchange in the opposite order in REMD, when $E_{low} > E_{high}$ as in the original implementation of the Metropolis criterion (Gustafson, 1998).

An example of the REMD energy landscape can be found on Figure 7. Note changes in the landscape's shape - because of the higher temperature (or lower pressure), barriers do not seem to be as high as in Figure 5 thus rising the probability of successfully passing over them. However, during the same simulation setup, the probability to jump back to the less folded conformation is also higher.

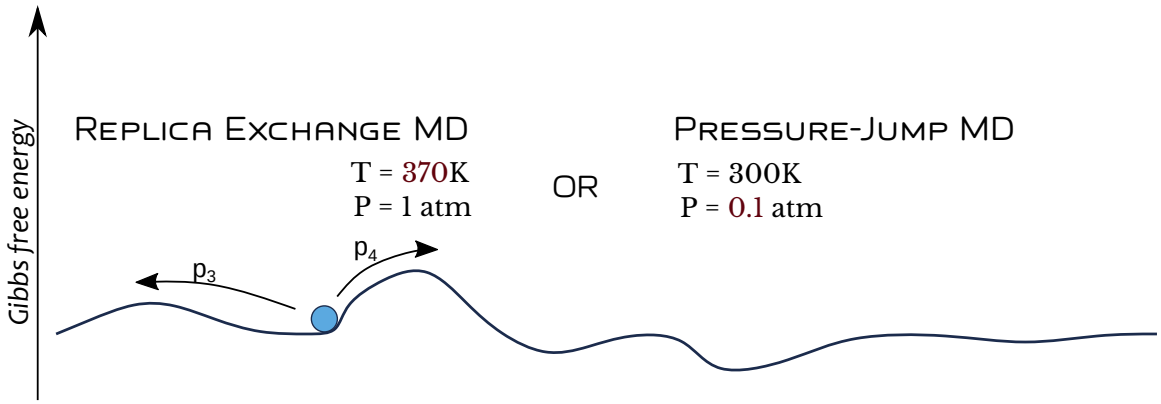


Figure 7: Representation of the energy landscape with lower (compared to the Figure 5 on page 13) energy barriers caused either by higher temperature or lower pressure.

Steered Molecular Dynamics

Steered Molecular Dynamics (SMD) (Izrailev et al., 1999), uses an artificial force that affects the protein's trajectory by pulling it along one or more directions to accelerate typical MD applications (Izrailev et al., 1999), overcoming energy barriers. To achieve such an effect, the protein's movement is constrained by a harmonic potential which pulls the protein into the folded state (or other target conformation). The force's magnitude may be either constant or variable which will result in constant or variable bias in the protein's movement speed. Example of SMD is shown on Figure 8.

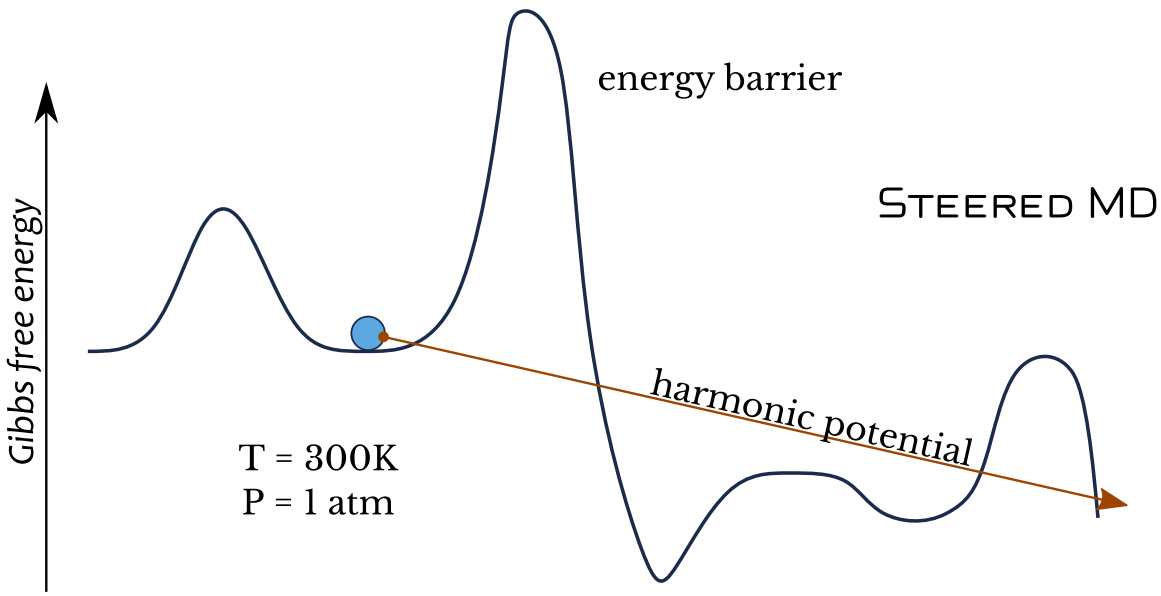


Figure 8: Representation of SMD. Blue circle represents protein conformation. Brown arrow represents harmonic potential that drags protein into the desired conformation.

High-Temperature Unfolding Simulations of Proteins

This method (Daggett and Levitt, 1993), is based on the principle of reversibility of the folding process and implies that unfolding is just the reverse process of folding. Unlike REMD and P-jump, it does not change temperature or pressure during the simulation, but keeps temperature constant and high enough for the protein to unfold. Several studies of various proteins (Daggett and Levitt, 1993; Mayor et al., 2000; Settanni and Fersht, 2008) show that the unfolding trajectory may be very close to the reversed folding trajectory. However, it was proved (Dinner and Karplus, 1999), that the energy landscape is temperature dependent, thus, the unfolding behavior at higher temperatures may well be very different. Finkelstein (Finkelstein, 1997) appeals to the principle of detailed balance, which states that the folding process must proceed via the same trajectories as the unfolding process when both of them are held at the same conditions, but processes held under different temperatures are not obligated to follow the same trajectory so one

cannot claim that the resulting trajectories (folding and unfolding) are necessarily the same.

Metadynamics

Dama et. al. propose a method that relies on modifying the energy landscape to offset free energy barriers (Dama et al., 2014). It adds bias to all previously sampled points which increases the probability of successful escape from the energy well and reduces the chances of becoming trapped in it again. That is, the more time the protein spends in roughly the same area, the more 'hills' are added to this area, making the transition event inevitable and significantly reducing the time needed to reach the folded conformation.

FRODAN

FRODAN is a geometric targeting method which uses idea that protein folding can be determined by geometric relationships between atoms (Farrell et al., 2010). While this method cannot generate optimal folding trajectories, since it does not account energy barriers caused by forces between atoms, it results in stereochemically plausible paths which can give an insight into the motions needed to perform changes in the conformations. This method takes into account covalent bond geometry, possible overlap of atoms, torsion angles, hydrogen bonds, and hydrophobic contacts. This method also demonstrates very high performance (up to 1000 times faster than regular or even SMD) of pathways generation which allows it to be applied to even very large proteins and protein complexes.

Problems With Enhanced Sampling Approaches

Note that REMD, SMD, high temperature unfolding, simulated annealing, and metadynamics are all biasing the energy landscape to achieve more efficient sampling.

Multi-replicate MD, while unbiased, is not too efficient since it does not guide the protein into the folded conformation but only relies on the time and nonzero probability that protein will fold in a finite time.

In contrast to methods mentioned above, an ideal approach to enhanced sampling of protein folding should be able to both reduce the simulation time without introducing artificial bias in the simulation. To our knowledge, no such method has been proposed in the literature so far.

Path-finding algorithms

Most of the methods described above use the notion of sampling to speed-up the simulation process and extract the main transitions that lead the protein to change it's conformation. We found that this problem is closely related to the shortest path problem (Yu and Yang, 1998) which consists of finding the shortest possible path between two nodes (points with nonzero distance, treated as independent entities), within an interconnected graph structure, preferably with as little computation as possible. With such an approach we hope to reduce simulation time and extract the shortest path which is complete enough to reconstruct the order of events that happened during the folding process without introducing energy bias. Shortest path algorithms are typically facilitated by the graph data structures (Madkour et al., 2017), however our problem has special properties, such as the inability to select the direction of the motion, the inability to check whether two nodes are neighbors, and other protein-specific issues described in detail below. We will therefore use trees (undirected; acyclic graphs) for the definition of the algorithms since trees better represent our problem domain because of the protein specific properties discussed in later sections. The most commonly used algorithms are the Breadth First Search (Kurant et al., 2010), the Uniform Cost Search (Felner, 2011), and the A* shortest path algorithm (Hart et al., 1968; Soltani et al., 2002). All of them

use an ordered queue of prospective nodes which are sorted by an estimated distance to the goal. The notion of the *wall* is typically defined as a barrier that prevents further movement in particular direction, while the notion of distance is defined differently within each algorithm.

A* search algorithm

The A* shortest path algorithm (Hart et al., 1968) uses information about the length of the total path traveled prior reaching the current node $g(x)$ where x defines current node, and the notion of a heuristic function that tries to predict the length of the path to the goal, $h(x)$. Absence of connection between the two nodes is called a *wall*. This algorithm also relies on a sorted queue. The sorted queue is a data structure which stores entries in sorted order defined by a sorting function. The sequence of steps involved in this algorithms is defined as follows:

1. Put starting node into the *open queue*: the sorting function is defined as $f(s) = g(s) + h(s)$
2. Take the first node from the *open queue*
3. Explore all nodes adjacent to the current node and put them into the *open queue*, ordered by the sorting function
4. Mark the current node as visited by putting it into the *visited queue*
5. If the goal is reached, reconstruct and report the full path, if *open queue* is empty, report that the goal is unreachable, otherwise go back to step 2

Such an approach tries to visit only the most promising nodes and assumes that the goal will be reached before exploring most of the nodes from the *open queue*. The attempt to do so depends on the computation of the heuristic function $h(x)$, and the best results can be obtained when it represents the exact distance to the goal. Underestimation or

overestimation of this distance may affect either the search time or optimality of the result, respectively. Special cases of using only $g(x)$ or only $h(x)$ lead to two different algorithms: Greedy Best-First Search (Doran and Michie, 1966) and Uniform Cost Search algorithm (Russell and Norvig, 2016; Nau, 1983). Figure 9 represents an example of the A* algorithm execution. Every circle represents a node and squares represent 'walls'. Line that connects start node with the goal node represents the path between these two nodes. Note how shorter, compared to Figure 10, the shortest path is.

Uniform Cost Search

This algorithm uses the same steps as A* with the sorting function defined as $f(x) = g(x)$. That is, it uses only information about traveled path to the node x. Such an approach guarantees the shortest path solution, but explores more nodes in the *open queue* as compared to the A*.

Greedy Best-First Search algorithm

This algorithm uses the same steps as A* with the sorting function defined as $f(x) = h(x)$. That is, it uses only heuristic information about the distance from node x to the goal. Such an approach depends heavily on the quality of the heuristic function. In the case of a perfect distance computation, this algorithm results in the shortest path solution with minimum of nodes visited from the *open queue* same as A* with perfect $h(x)$. However, the perfect heuristic distance function is not feasible in almost all practical cases, which may lead to a longer (suboptimal) paths. Figure 10 represents an example of the greedy search algorithm. Note suboptimal path to the goal caused by the backtracking.

Protein distance

All shortest path algorithms described above operate under notion of the distance between two nodes. Below we will review the most common approaches to measure the

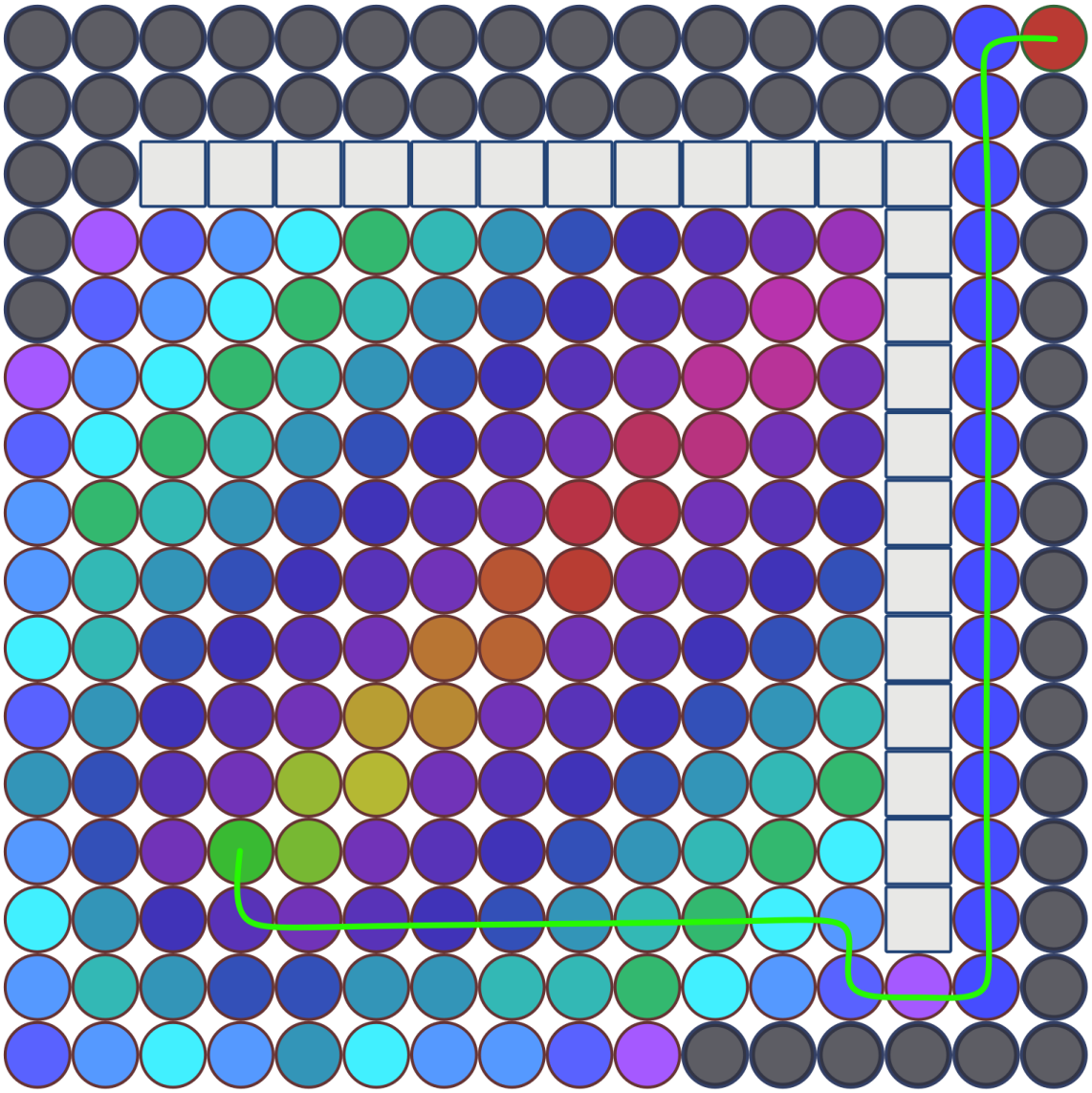


Figure 9: The A* algorithm progress map. Circles represent potential moves, squares represent a 'wall'. Green line shows final route to the goal. Circles' color gradient (green-red-blue) represents order of unvisited nodes exploration. Green circle represents initial point, red circle represents goal.

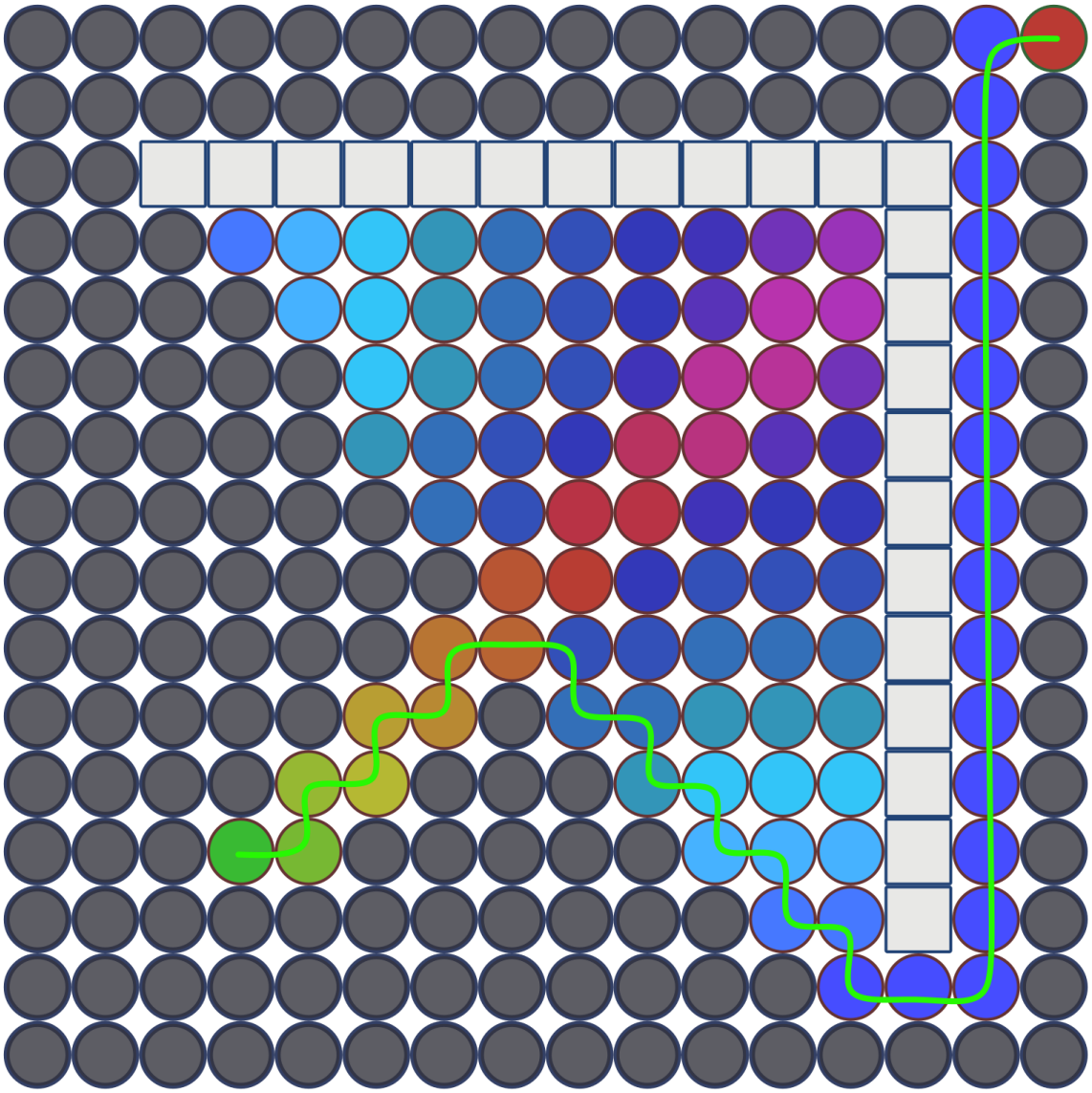


Figure 10: Greedy algorithm progress map. Circles represent potential moves, squares represent a 'wall'. Green line shows final route to the goal. Circles' color gradient represents order of unvisited nodes exploration. Green circle represents initial point, red circle represents goal.

distance between two conformations since these may be used to formulate $g(x)$ and $h(x)$ in the shortest path algorithms.

RMSD

Root-mean-squared deviation (RMSD) is a commonly used measure for comparing protein conformations (Kufareva and Abagyan, 2011; Carugo, 2007). It is defined by the equation:

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|r_i - \check{r}_i|^2}, \tag{2}$$

where $|r_i - \check{r}_i|^2 = \left(r_{i,x} - \check{r}_{i,x}\right)^2 + \left(r_{i,y} - \check{r}_{i,y}\right)^2 + \left(r_{i,z} - \check{r}_{i,z}\right)^2$, is the square distance between superimposed molecules, where r and $\check{r}$ are the coordinates of the common atoms from two respective conformations, n - total number of atoms. Before computing this metric, the two protein conformations are aligned to minimize the distance between them. There are two common ways to measure RMSD: backbone RMSD (BBRMSD) and all atom RMSD (AARMSD). The first way is more sensitive to the secondary structure and ignores sidechain groups, while the second way takes into account additional details in the tertiary structure. An example of this metric can be viewed on Figure 11. As one may see, the two conformations are aligned first and only then the shortest distance is being computed. However, RMSD is sensitive to outliers and bends in flexible regions of proteins, which can result in a significant increase in RMSD even when two structures are nearly identical. An example of such behavior is shown in Figure 12 which shows how one angle change (C') in the conformation can drastically change the average distance between the compared atoms.

Dihedral angles

A dihedral (torsion) angle is the angle formed by two intersecting plains, each defined by a group of three atoms. In molecular biology such angles are used to define the

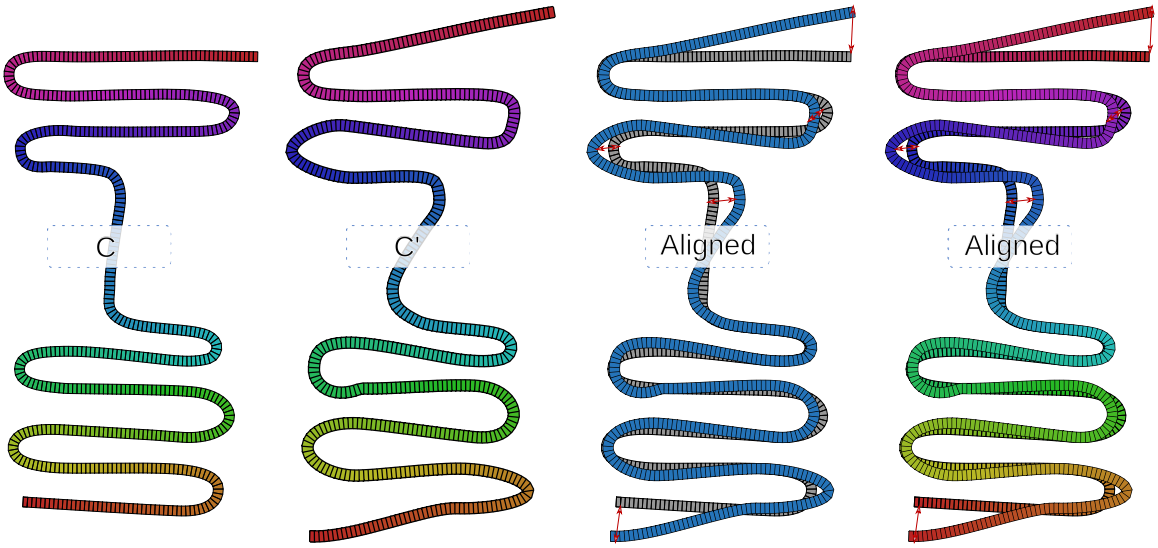


Figure 11: Example of RMSD metric. C - original conformation. C' - slightly different conformation. Aligned - two proteins aligned to each other. Red arrows represent distance between corresponding elements.

angle between two sets of three atoms. Originally introduced in 1963 (Ramachandran, 1963), the protein dihedral angles are called ϕ, ψ, and ω formed by following the chain of backbone atoms. That is, each amino acid forms three such dihedral angles along the backbone of the protein. ϕ represents the angle between the backbone nitrogen and alpha-carbon, ψ - between the alpha-carbon and carboxyl carbon, ω - between the carboxyl carbon and backbone nitrogen. ω is commonly omitted since it fluctuates tightly around ± 180° *(trans)* for all amino acids except *cis* conformations which have 90°. An illustration of dihedral angle positions is shown in Figure 13.

Additionally, measuring pure angular distance can lead to undesired behavior when the angle values are 1°and 359°. While the real angle is just 2°, pure subtraction would result in |358°|. To address this specific behavior angles are translated using a sin/cos projection (Rajan et al., 2010). A distance metric for two conformation of the same

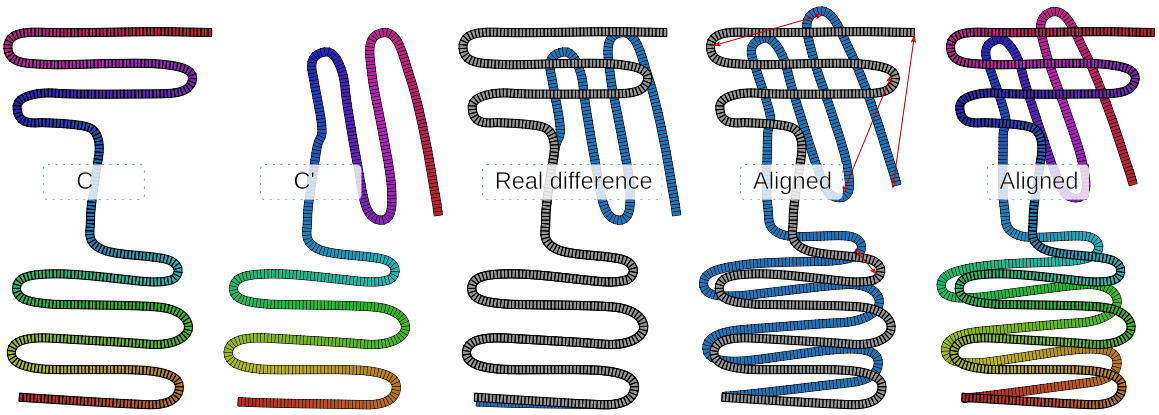


Figure 12: Example of how one bent can increase RMSD metric.
C - original conformation. C' - slightly different conformation.
Real difference - shows how protein was changed. Aligned - two proteins aligned to each other.
Red arrows represent distance between corresponding elements.

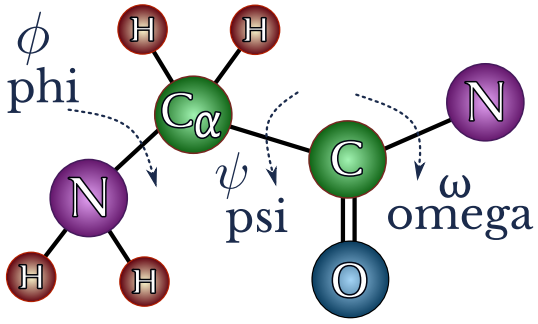


Figure 13: Dihedral angles' positions on an amino acid.

protein can therefore be defined as:

$$\sum_{i}^{A} (|sin(a_i) - sin(\check{a}_i)| + |cos(a_i) - cos(\check{a}_i)|), \tag{3}$$

where A - number of amino acids in the protein, a_i - ith dihedral angle of the first conformation and $\check{a}_i$ - ith dihedral angle of the second conformation. Such a metric is more stable to bends of protein's flexible regions as compared to RMSD.

Contact map

A protein's contact map is constructed by first computing a symmetric, square matrix of pairwise, inter-residue contacts (Kufareva and Abagyan, 2011): $\sigma(a_i, a_j) = |r_i - r_j|$, where r_i represents the coordinates between two elements which can be alpha-carbons, hydrogens, or all atoms. A contact matrix can then be described as a boolean matrix where 'True' value is assigned when distance between two residues is less than ξ. A typical ξ value is taken as 2.7 Å- approximate size of the water molecule (Huang et al., 2013). That is, if a water molecule cannot fit between two elements - we call it a contact. Such a map can be used to represent secondary or tertiary structure, but may lack precision with ξ values which are either too high (there is always a contact) or too low (not possible to achieve such distance).

METHODOLOGY

Folding methodology

As discussed in the 'Path-finding algorithms' subsection (page 19) there is a need for an algorithm capable of speeding up the folding process without introduction of an unnatural bias. We suggest that a combination of the shortest path problem and a sequence of short molecular dynamics simulations may result in a compact reproduction of the folding process.

Protein specific properties

Before we continue with merging the shortest path finding algorithms with Molecular dynamics (MD), it is necessary to review specific behaviors of proteins:

- An initial state in our case is an unfolded (denaturated) conformation of the protein. However, any conformation can be viewed as the initial conformation.
- A goal is the conformation that we want to reach. While it can be any conformation of the same protein, we view the goal as a folded conformation obtained from nuclear magnetic resonance (NMR) spectroscopy.
- Proteins in general have at most $3N - 6$, where N is the number of atoms, degrees of freedom (DOF) available for the movement, but we cannot explicitly choose which DOF to use as a direction of the movement due to coupled atomic interactions in the force equations.
- In practice, proteins have a limited number of DOF (less than $3N - 6$) available for movement. These are complex nonlinear motions which we model using MD methods.

- *Walls*, defined in shortest path algorithms as an absence of the connection between the two nodes, are energy barriers which have nonzero probability of passing over them.
- A goal may not be achievable in any finite time, because of the imperfections in the force field and/or MD in general which cause force field/MD-specific energy barriers which do not exist in the natural environment.
- There are many metrics to measure the distance between the conformations, but they do not always agree, and sometimes contradict each other.
- While a specific metric can claim small differences between two conformations, the real difference between them may be much larger, because of energy barriers (when close to the goal) and vice versa: large distance can be covered very quickly (initial folding steps).
- Two conformations with a small distance, may have opposite directions of movement and velocities, which may prevent transitions between the conformations (Figure 14).
- While generally proteins move towards the folded conformation, particular parts of the trajectory may not be strictly necessary. With some probability, a protein can move in the opposite direction as well.

Sorting function

In the Background section on page 20 we showed that Dijkstra and Greedy approach (Reddy, 2013) are special cases of the A* algorithm. From Figures 10 and 9 we can see that a more greedy approach results in a reduced number of nodes visited, but a less optimal path. We propose the usage of an additional tuning parameter $0 < \alpha \leq 1$ during the sorting process since MD steps are not guaranteed to always make progress

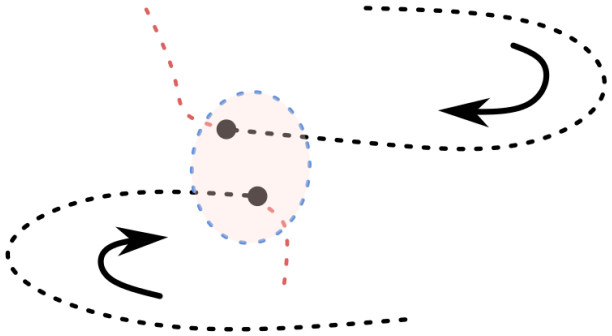


Figure 14: Example where points with a small distance may not be called neighbors. Red color represents future steps in the trajectory.

to the goal (folded or target conformation): $f(x) = \alpha g(x) + h(x)$. The initial value of α is set to 1, and after a defined number of failures to make a step closer to the goal, it's value is decreased, thus making the algorithm increasingly greedy and reducing exploration of nodes with lower probability of becoming closer to the goal during the very next step.

All shortest path algorithms are based on the notions of starting node, goal node, distance metric that exists between any two nodes, and the amount of distance that can be covered per one time unit. Reflections of these properties can be found in the molecular dynamics simulation of proteins as follows: the starting position is the unfolded conformation, and the goal position is the folded (target) conformation. The distance between the current and target conformations can be taken as the Root-mean-squared deviation (RMSD), the sum of the $\phi - \psi$ amino acid angles, or the difference in formed contacts. We also use a fixed simulation time to sample protein motion using MD which we will call a *step*.

A general, efficient approximate shortest path algorithm applicable to MD protein folding may therefore be defined as Algorithm 1 on page 32. C_s, C_g - start and goal nodes, C_c - current node, C_{new} - newly explored nodes (forward neighbors), α - greediness tuning parameter.

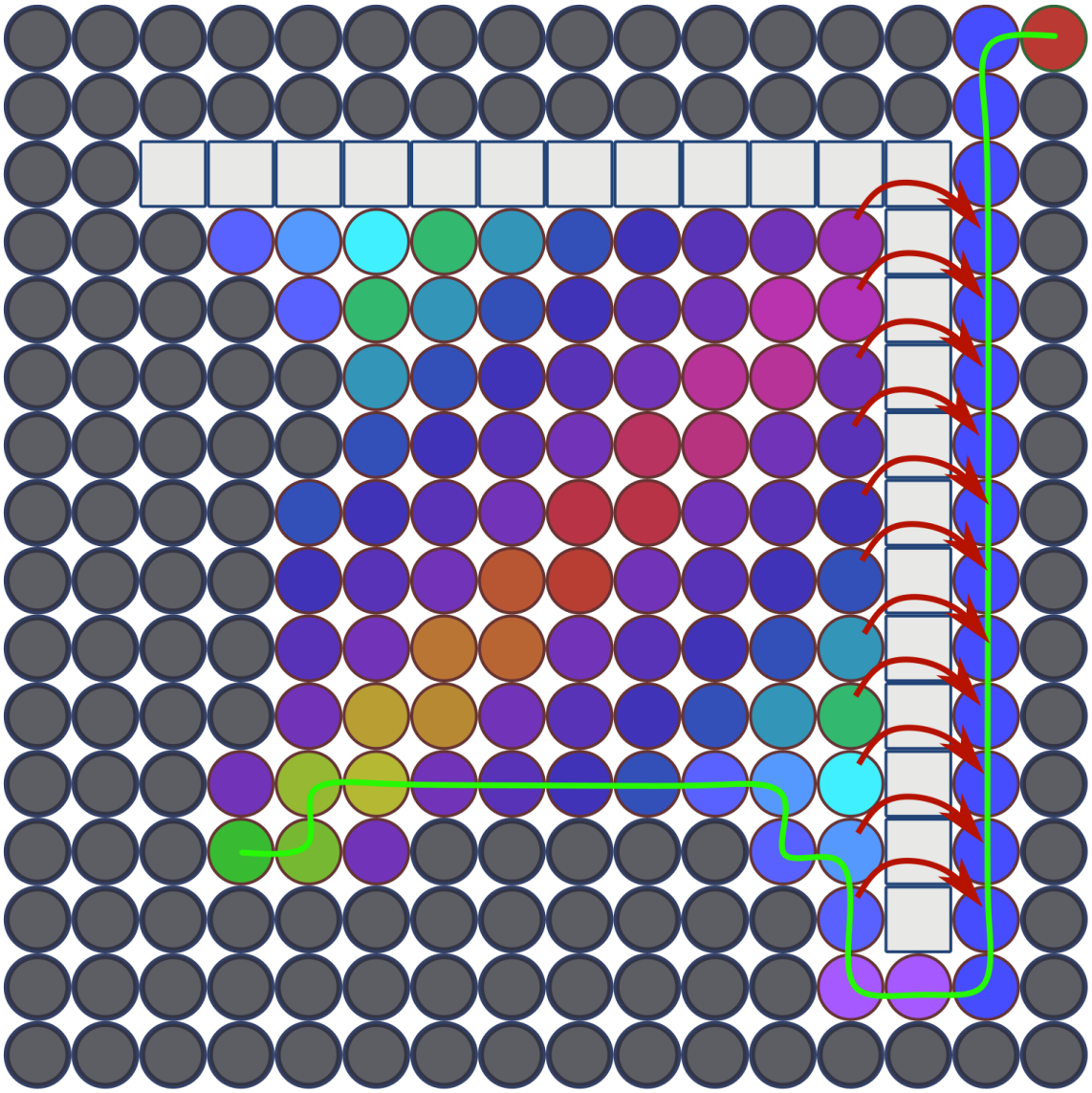


Figure 15: Shortest path algorithm with variable greedy factor. The circles represent potential moves, squares represent a 'wall'. The green line shows final route to the goal. The circles' color gradient represents the order of exploring unvisited nodes. The green circle represents the initial point, and the red circle represents the goal.

Algorithm 1 General greedy proximate A* shortest path finding algorithm

```
 1: procedure GPA*(C_s, C_g)
 2:     OpenQueue ← SortedQueue.create(sort_func = "f(c) = αg(c) + (1 − α)h(c)")
 3:     C_c ← C_s
 4:     while not OpenQueue.empty() and C_c ≠ C_g do
 5:         C_new ← Explore_neighbours(C_c)
 6:         for all C_new do
 7:             C_new[i].to_goal ← Dist_compute(C_new[i], C_g )
 8:             C_new[i].from_prev ← Dist_compute(C_new[i], C_c )
 9:             OpenQueue.put(C_new[i])
10:         end for
11:         ClosedQueue.put(C_c)
12:         C_c ← OpenQueue.pop()
13:         α ← Adjust_greed_factor()                          ▷ if needed
14:     end while
15:     if C_c == C_g then
16:         Path_reconstruct(ClosedQueue)
17:     else
18:         Report_goal_unreachable()
19:     end if
20: end procedure
```

Note that Algorithm 1 is a general algorithm and it is applicable to any path finding problem when the shortest path is desired, but not required. However, MD-specific properties defined in "Folding methodology" subsection require us to review Algorithm 1 and include some changes:

- The open queue is going to grow very quickly since every iteration will introduce $m-1$ new nodes, where m is the number of parallel MD simulations (steps) with different initial velocities from the current node.
- We do not define the exact metric to use as a distance measurement between two conformations since each metric has strong and weak points. Instead we introduce a method for utilizing alternative metrics when progress has stalled.

Let's review these problems in more detail.

Queue size

As stated in the previous section, we cannot define the direction of our movement or check whether a node was visited before, so the size of the open queue will be increased by $m-1$ new elements, where m is number of steps from the current node. Since we are interested in finding a path to the goal in the first place and only wish to have the shortest path, by the time we meet our primary goal, the open queue size would contain $(m-1)*k$ nodes, where k is the total number of steps needed to build a path to the goal (within the termination condition distance from the goal, ϵ). Furthermore, some of the steps will be in the opposite direction, making them practically useless. This property would cause the algorithm to pick an inefficient step from the open queue and perform further simulations (which are computationally expensive). To solve the problems stated above, we propose an additional check condition before new nodes are inserted into the open queue: insert only when a new node is closer to the goal than any other nodes found so far, or when it's distance change towards the goal is greater than a predefined threshold T and distance from the previous node is greater than θ. That is, filtration of nodes that do not make enough progress either towards the goal or from the previous node (conformation) should reduce the chances of sampling conformations which are essentially the same. While the θ value may be varied, we recommend $T/2$ as a good balance between queue growing speed and retention of promising nodes.

Proper distance metric

While RMSD is the most commonly used distance metric (Kufareva and Abagyan, 2011; Carugo, 2007), our studies have shown that it may be inaccurate. For example, Figure 16 shows a folding trajectory of the Trp-Cage Miniprotein Construct TC5b (1L2Y) described by the RMSD (blue line) and dihedral angle distance (ANGL) (red

line) metrics. We can see that at the very beginning RMSD rapidly decreases, while the ANGL metric values increase. Furthermore, Figures 17 and 18 indicate that neither RMSD nor ANGL have a strong match with the protein's potential energy, thus we cannot claim that one metric is more correct than another. In order to overcome this problem we propose metric switching during the search process. That is, when the algorithm registers no progress after a particular number of attempts with the current metric, the algorithm switches metrics from RMSD to ANGL or the contact distance. Such an approach should not only help to resolve a particular metric's artifacts (RMSD favors whole structure collapse, ANGL does not account for the sidechain), but also opens up new ways of passing energy barriers. Furthermore, we introduce several new metrics based on the contact map distance.

As was mentioned before, the contact map shows whether the smallest distance between two atoms is smaller than some particular threshold. We suggest using logical functions such as XOR and AND to find the difference between the current and goal contact map results to obtain a disagreement score (XOR) or an agreement score (AND). We call these metrics contact map distance disagreement (XOR) and contact map distance agreement (AND). Additionally, we introduce the hydrogen bonds contact map distance agreement (ANDH) metric which considers only contacts between hydrogen atoms and other atoms, and should show agreement between current-target hydrogen bonds which are essential for secondary structure formation. The metric usage order may play a significant role in the algorithm properties, for example, RMSD favors collapsing motions, similarly to the Steered Molecular Dynamics (SMD) approach, while ANGL tries to rotate backbone atoms in the protein. One of the options is a round-robin selection, another way is to measure the signal to noise ratio (SNR), where signal is the difference between distances to the goal from current and previous nodes, and noise is the ambient noise measured initially. The choice of the best order needs further

research. We also introduce the notion of a *guiding metric* - such a metric from the set of metrics that, once lower than the lowest distance to the goal, updates all other metrics' best pointers to the current node.

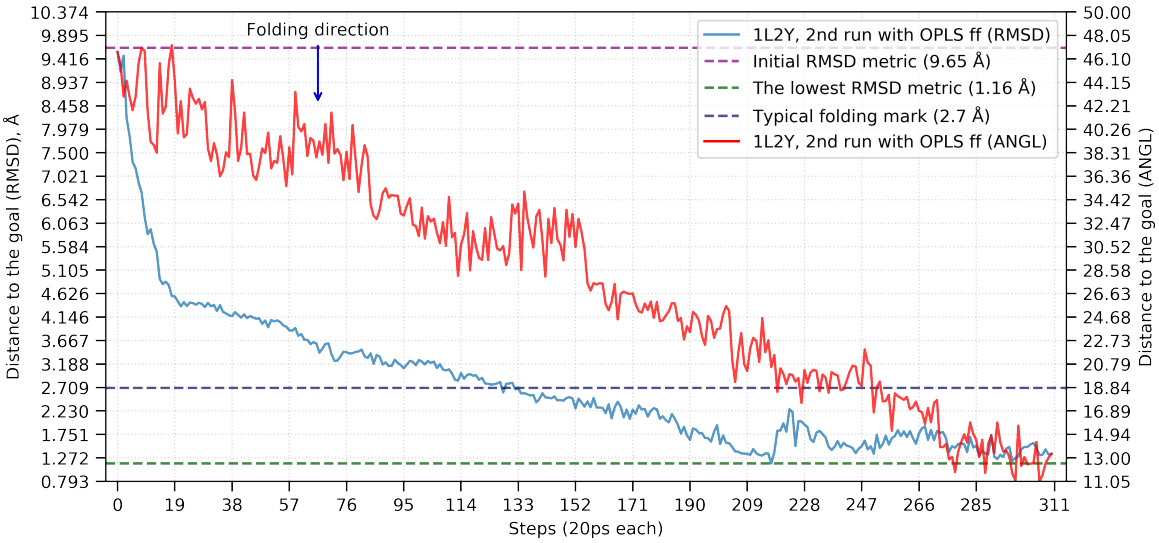


Figure 16: Discrepancy between the RMSD (blue) and ANGL (red) metrics. Generated from folding trajectory of 1L2Y with the OPLS force field.

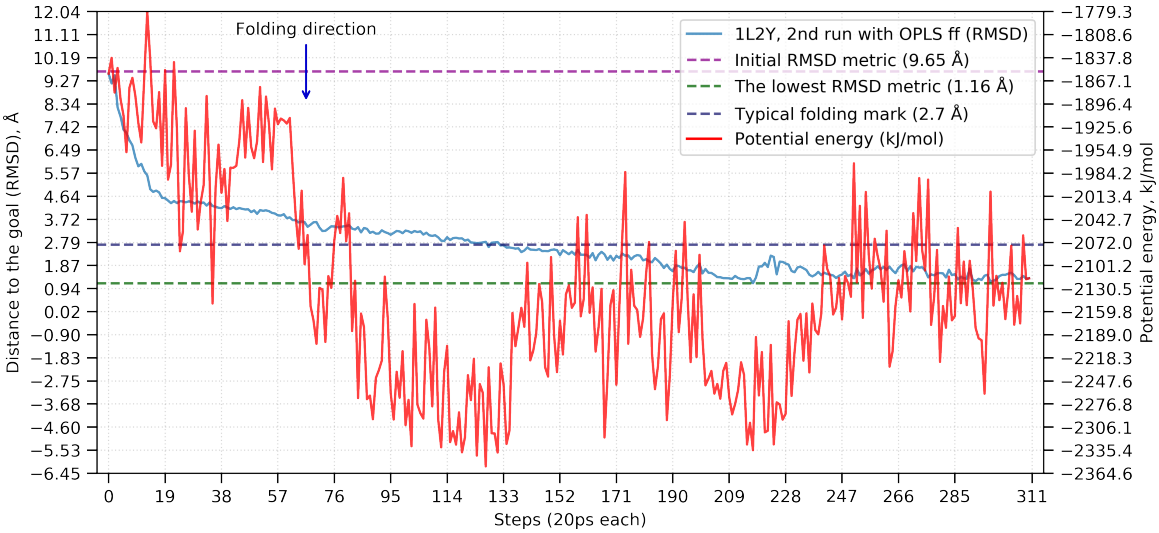


Figure 17: Discrepancy between the RMSD (blue) metrics and protein's potential energy (red). Generated from folding trajectory of 1L2Y with the OPLS force field.

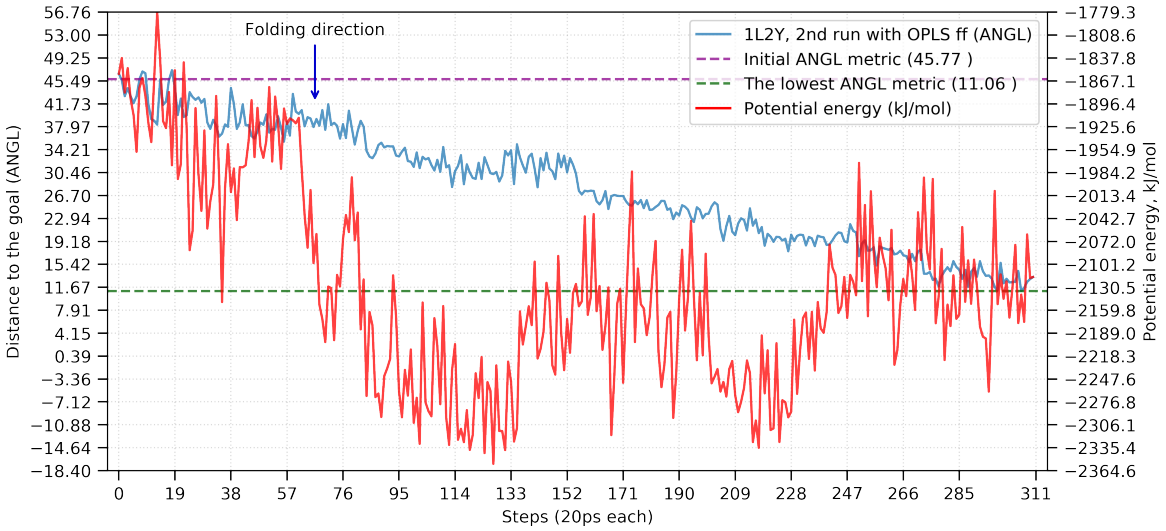


Figure 18: Discrepancy between the ANGL (blue) metrics and protein's potential energy (red). Generated from folding trajectory of 1L2Y with the OPLS force field.

Stop condition

It is common in the literature that proteins are considered folded when backbone RMSD (BBRMSD) between alpha carbons is less than or equal to 3 Å which is an approximation of the 2.7 Å size of the water molecule (Huang et al., 2013). Since our algorithm did not have problems with reaching 3 Å not only between alpha carbons, but between all atoms, as a stop condition we decided to use all atom RMSD (AARMSD) to the goal equal to or greater than NMR resolution of the goal's Protein Data Bank (PDB) structure. However, due to imperfections in the force fields and the statistical properties of MD, we cannot guarantee that the goal will be achieved in any finite time. So, we stopped calculations when RMSD distance between all atoms was at least 2.7 Å and there was no observed improvement in the reducing distance to the goal across all metrics. Note, that there is always a non zero probability that an extra iteration would result in a smaller RMSD distance.

All extra improvements of the Algorithm 1 are reflected in Algorithm 2.

Algorithm 2 MD specific greedy proximate A* shortest path finding algorithm

```
1: procedure GPA*(C_s, C_g)
2:     OpenQueue ← SortedQueue.create(sort_func = "f(c) = αg(c) + (1 − α)h(c)")
3:     metric ← Change_metric(metrics)
4:     C_c ← C_s
5:     while not OpenQueue.empty() and C_c ≠ C_g do
6:         C_new ← Explore_neighbours(C_c)
7:         for all C_new do
8:             C_new[i].to_goal ← Dist_compute(metric, C_new[i], C_g )
9:             C_new[i].from_prev ← Dist_compute(metric, C_new[i], C_c )
10:            if C_new[i].to_goal > best_so_far or (C_c.to_goal − C_new[i].to_goal ≥ T
    and C_new[i].from_prev − C_c.from_prev ≥ T/2 then
11:                OpenQueue.put(C_new[i])
12:            end if
13:        end for
14:        ClosedQueue.put(C_c)
15:        C_c ← OpenQueue.pop()
16:        if fails_num > threshold then
17:            α ← Adjust_greed_factor()          ▷ decrease, make more greedy
18:            metric ← Change_metric(metrics)                    ▷ if needed
19:        end if
20:        if progress then
21:            α ← Adjust_greed_factor()          ▷ increase, make less greedy
22:        end if
23:    end while
24:    if C_c == C_g then
25:        Path_reconstruct(ClosedQueue)
26:    else
27:        Report_goal_unreachable()
28:    end if
29: end procedure
```

Software implementation

General

Python3 (3.6) (Van Rossum and Drake, 2011) was selected as a programming language since it allows rapid development and acceptable performance. Additionally, it allows for extensions with the C/C++ language to improve the performance of computationally demanding parts. Our experience indicates that with small and medium size proteins there was no need in tuning the performance any further since most of the execution time is spent waiting for the MD simulations to complete.

While our algorithm can work with any MD simulation package, we implemented wrappers to support GROningen MAchine for Chemical Simulations (GROMACS) (Pall et al., 2014). We use Bourne again shell (BASH) (Ramey and Fox, 2003) to specify GROMACS parameters of the execution, while Python script generates configuration files for each simulation.

Detection of inefficient steps

Initially we take the goal conformation and run several short plain MD simulations. The number of simulations is determined by the number of parallel steps that algorithm is going to use during the regular execution. Once simulations are complete, we measure all metric distances between the two conformations and select the smallest values among all parallel runs. Finally, these values are multiplied by an empirically chosen value (0.8) to represent the smallest possible deviation that goal conformation can produce. We call these values ambient noise and they are used as described in the Methods section, to filter steps that did not make any movement from the previous conformation or reduced distance to the goal conformation less than 50% of the noise value. These values are computed only once and saved in the text file. Later, in case of crash or desire to continue

the search process, noise values are read from the the file and the regular algorithm execution continues.

Data storage

One of the MD-specific parameters is initial velocities of water molecules, which depends on, so called, seed value. During each step a set of MD simulations is performed, each using a different seed value (the user can control the start seed number of the sequence, the algorithm then decides when to perform the switch to the next seed number), generating several output files:

- .gro text file with the last state of the simulation (in GROMACS), which can be used as the first frame of the new simulation.
- .xtc binary file with compressed coordinates of the protein only, which is used to compute all distances to the goal and from the previous step.
- .angl binary file which contains dihedral angles information, which is used to compute all distances to the goal and from the previous step.
- .cont.npz archived Numpy (Oliphant, 06) list which contains contact information, which is used to compute all distances to the goal and from the previous step.

Filenames are generated with BLAKE2s-256 (Aumasson et al., 2013) - fast and collision resistant hash function which generates a 256 bit hash string (64 characters) from the sequence of seeds visited prior to the current step. A database (SQLite (Team, 10) 3.28.0) is used only for storage and future analysis of the results and for recovery after crash. The database entity relationship (ER) diagram can be found in Figure 19.

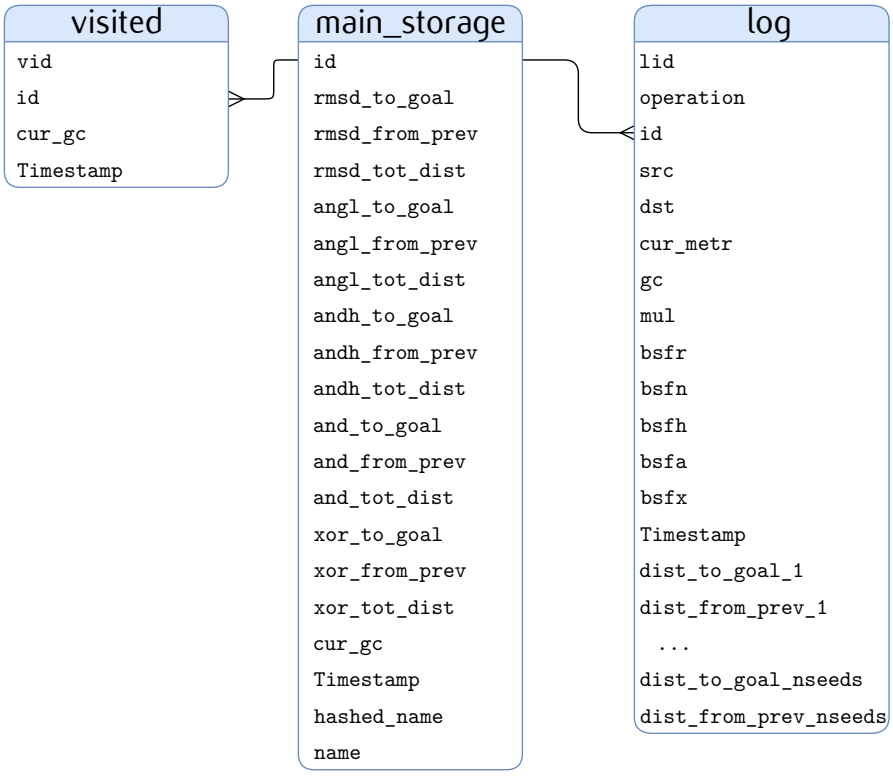


Figure 19: ER diagram for entities in the database. Three dots in log table mean that number of columns is determined during the first execution and is equal to the number of seeds.

Parallelism

There exist several levels of parallelism in our software implementation: parallelism in the main script, parallelism in the execution of different seeds, and parallelism of the MD software.

All MD simulation packages are essentially parallel, but optimal settings in most cases have to be configured manually. GROMACS is no exception and accepts the number and type of threads through program arguments. Greedy-proximal A* (GPA*) has wrappers for GROMACS which allow it to tune the simulation's performance. The software itself also uses separate threads to perform I/O operations and database communication. In

theory, any other MD simulation package could be used, but this is beyond the scope of our work.

Fault recovery

Faults may happen at any point since hardware is not perfect. Although our implementation stores enough information about every step to be able to recompute distances between points without repeating MD simulation, the process may still take a significant time bounded by the I/O performance of the hardware. To remove such an inconvenience and use the computing resources optimally, we implemented a way to continue computing in a much less time without the risk of using corrupted data. We store all local dictionaries in pickled (data serialization protocol) form on the disk. We keep two copies, current and previous. Once we activate recovery from the previous crash, our implementation assumes that a previous database exists and no algorithm parameters were altered. It then finds common sequences between in-memory data structures and tables of the database, and copies data from the old database into the current database in such a way that the very next operation of the algorithm is not included in the new database even though it may exist in the previous one. That is, after the recovery state is several steps behind the last run. All steps after the backup common point will be processed by computing metrics only or reading their values if sufficient files exists. In the case of an absent node metric file, a new file will be generated. In the case of an absent .gro or .xtc file - a full simulation step will be executed.

Such an approach allowed to reduce startup time after the crashes from days to minutes.

Testing protocol

GPA* testing protocol

For testing GPA* algorithm performance we selected two proteins that reflect the two main classes of secondary structure - alpha-helical structure (1L2Y) and beta-sheet structure (1GB1). Additionally, we extended our set of proteins with another protein that represents alpha-helical structure - 1YRF (Chiu et al., 2005), which is known as the fastest folding protein and is a typical candidate for testing MD approaches (Snow et al., 2005).

All simulations were performed using GROMACS 2019.3 (Pall et al., 2014), however, our algorithm can be used with any other MD simulation packages like NAMD (Phillips et al., 2005) or AMBER (Case et al., 2018; Lindorff-Larsen et al., 2010). Several force fields were used: AMBERff99SB-ILDN force field (AMBER) (Lindorff-Larsen et al., 2010), CHARMM36-nov2018 force field (CHARMM) (Huang and MacKerell Jr, 2013), GROMOS54a7 force field (GROMOS) (Oostenbrink et al., 2004), and OPLSaa force field (OPLS) (Jorgensen et al., 1996). With AMBER, CHARMM, and OPLS we used the TIP3P water model (MacKerell Jr et al., 1998), while with GROMOS we used SPC/E water model (Berendsen et al., 1984). Water molecules were substituted with Na^+ and Cl^- to reach a salt concentration of 1.5 mol/liter. Afterwards, if the total charge was not zero, additional atoms were added to neutralize the system. We used the velocity rescale thermostat (NVT) to control the temperature and the LINCS Hess et al. (1997) algorithm to constrain hydrogen bonds. The default MD simulation temperature was set to 300 K. The default step duration was set to 20 ps.

As the goal conformation we selected the NMR PDB structures of the 1L2Y (Neidigh et al., 2002), chicken villin subdomain HP-35, N68H protein (1YRF) (Chiu et al., 2005), and immunoglobulin binding domain of streptococcal protein G (1GB1) (Gronenborn et al., 1991). Note, that if the PDB structure contained several conformations, we used only the first one for comparison. It is possible that other structures may result in even

Table 1: MD simulation box properties

Protein	Box side length, Å	Number of molecules			
		Protein	Water	Na^+	Cl^-
GROMOS					
1L2Y	44.7	198	2678	80	81
1YRF	42.6	383	2263	70	72
1GB1	49.1	562	3391	111	107
AMBER, CHARMM, OPLS					
1L2Y	45.2	304	2730	84	85
1YRF	43.6	582	2350	75	77
1GB1	49.1	855	3364	111	107

better RMSD results. The unfolded conformation was obtained by heating each protein to 800 K during 2 fs × 500000 = 1000000 fs = 1 ns to completely remove any secondary structure. Additionally, we used visualization tools to verify the absence of any signs of secondary structures in the conformation. In order to guarantee the same initial structure, the unfolded conformation was translated to .pdb (force field agnostic) format. The unfolded conformation was adapted to every force field and used as the initial conformation in the folding process. In our experiments we used a cubic box shape. The simulation box specifications can be found in Table 1.

In order to guarantee the same experimental conditions, we stored the whole preparation process sequence in a BASH script, which was executed for every protein/force field combination. Finally, we extracted a protein-only structure for future visualization and analysis.

For distance computation we used metrics in the following order: RMSD, ANGL, AND, ANDH, XOR. RMSD was our guiding metric.

During each iteration 4 trajectories with different initial conditions are produced. After testing GROMACS performance, we found that the most optimal execution can be achieved by running one instance per node. During the research process we successfully

ran GPA* on hardware with various computing capabilities (Biosim, MrGreen, Voltron, Babbage), which typically had 24-64 cores per node, 64-256Gb of RAM and 4-20 nodes. To measure the overhead of our implementation we compared the time required for one step (4 parallel simulations) during the rebuild process (no MD simulations are executed) and during the regular simulation process and found that the rebuild step took only 5-7% of the overall time. However, this time heavily depends on the I/O performance, and since we randomly read small files, it can be significantly reduced when using solid state drives. Additionally, these numbers show that the usage of high performance programming languages like C/C++ or Fortran would result only in several percent improvement of the total performance. However, integration of this method into the MD simulation software may have a significant positive performance impact due to procedures needed to initiate the simulation process.

We did not know ahead of time how many steps would be required to reach the folded conformation for each protein, so we launched the algorithm with each force field and tracked the performance. Once an acceptable RMSD was reached for the majority of the force fields, we stopped the execution process. The exact simulation times for each run can be found in the Results section. Since 1L2Y was folding relatively quickly (one week of real time), we ran the algorithm twice for this protein. We list only the RMSD metric since it is a common metric used to compare results in the literature.

REMD testing protocol

We used GROMACS 2019.3 to perform Replica-Exchange Molecular Dynamics (REMD) to match GPA* simulation conditions. In order to reduce the time needed for simulation we used only the AMBER force field. The number of steps for REMD was selected to approximately match the total number of steps used by GPA*, divided evenly between the number of replicas. A temperature ladder was selected from 300 K to 400 K. The number of replicas was selected with a temperature predictor for parallel

tempering simulations (Patriksson and van der Spoel, 2008) separately for each protein. All tables can be found in Appendix A, Chapter I Extra Tables. Since we ran GPA* twice for 1L2Y we decided to have two runs of REMD. VMD 1.9.4a35 (Humphrey et al., 1996) was used to compute RMSD to the NMR conformation across the whole trajectory for each replica. Both 1L2Y runs were performed on the Voltron cluster with GPU acceleration (OpenCL). After the simulation we rebuilt the full trajectories for further analysis. We list only RMSD metric since it is a common metric to compare the results.

FRODAN testing protocol

While this approach does not involve MD, we compare its trajectories with the GPA* final trajectories. Since this approach does not use force fields, we generated three folding trajectories, one for each protein. VMD 1.9.4a35 (Humphrey et al., 1996) was used to compute RMSD to the NMR conformation across the whole trajectory.

SMD testing protocol

SMD was selected as a goal oriented method. Similarly to REMD we used only the AMBER force field. Temperature was the same as in GPA* - 300 K. We tried different magnitudes of harmonic potential force and empirically selected the following values: 1 kJ/mol, and 10 - 90 kJ/mol with step 10 kJ/mol. Higher values usually lead to rapid folding in several picoseconds, while lower values were not sufficient to form the secondary structure. Combinations of different magnitudes would be able to solve these problems, but even then additional study is required to determine sufficient force magnitudes. VMD 1.9.4a35 (Humphrey et al., 1996) was used to compute RMSD to the NMR conformation across the whole trajectory.

Visualization protocol

Visualizations were done in VMD 1.9.4a35 (Humphrey et al., 1996) compiled with Tachyon (Stone, 1998) raytracer accelerated by OptiX framework (Parker et al., 2010) for movie generation and POV-Ray (of Vision , TM) raytracer for static images. For secondary structure computation we used the STRIDE (Frishman and Argos, 1995) binary VMD plugin along with a SSCache (, dalke@ks.uiuc.edu) TCL script which was able to run the secondary structure computation for every frame, thus allowing us to track secondary structure formation events.

Analysis protocol

GPA* analysis

Shortest trajectory analysis We extracted trajectories that form steps with the smallest distance value for each metric in every run. Since we ran our algorithm with several proteins and force fields, trajectories were extracted for all combinations of proteins/force fields.

- For better analysis of the relations between metrics and force fields, we extracted the total time spent during the search. Time was expressed in steps, since different CPUs did not have the same performance.
- To further understand each metric's specific properties we computed the correlation and determination coefficients between each metric's best folding trajectory for each protein and force field used during the simulation.
- We extracted the protein's potential energy from the simulation and computed the correlation and determination coefficients between each metric's best folding trajectory for each protein and force field used during the simulation.

- We plotted values of all combinations of metrics to visualize similarities and differences between the metrics.
- We added protein's potential energy values as an independent metric and compared it to each metric's values.
- We plotted relations between distance to the goal and past distance to determine locations of energy barriers that were encountered during the search process.
- During the visualizations of the trajectories, we found that the final conformations obtained from different metrics are not always the same. Thus, we provided examples of such behaviors, similar to the previous analysis that was done to all the force fields used in our GPA* runs.
- We plotted the first and last frames of the folding process for each protein to display the folded conformation with all secondary structures that we were able to achieve during the search process.

Analysis of the best achieved conformations To understand the consistency in the results for GPA* runs across different force fields, we extracted the smallest RMSD values achieved during the folding process for each protein and force field. Additionally, we extracted the time at which the best RMSD value was reached. For multiple runs with the same combination of protein and force field, we computed the average results across all the runs.

Metric analysis In order to perform analysis of the metrics used, we extracted the number of promotions (events when the current distance is smaller than any encountered before) while each metric was active.

Since the RMSD metric was prioritized during all runs, we normalized all our results and compared them. This analysis helped to see what metric was guiding GPA* better

and to analyze the combinations of metric/force field to determine which combination works the best.

REMD performance compared to the GPA*

As part of the analysis of REMD, we extracted the smallest RMSD to the goal found in each replica, and visualized the trajectory from the very beginning until the smallest RMSD was reached. For a complete and fair comparison of REMD and GPA* we used the following metrics: best achieved RMSD distance to the NMR conformation during the REMD and GPA* runs, average distance to the NMR conformation during the REMD and GPA* runs, time when the smallest RMSD was reached during the REMD simulation, and trajectory lengths of REMD and GPA* and when the smallest RMSD was met.

SMD and FRODAN performance in comparison to the GPA*

For the SMD analysis, similarly to the REMD comparison, we used the final AARMSD to the NMR conformation. Additionally, we computed BBRMSD distance between the initial conformation and the NMR structure, and the final conformation and the NMR conformation. We visualized the folding trajectory and visually compared it to GPA* to find out whether they exhibit a similar sequence of secondary structure formation during the folding process.

RESULTS

The performed Molecular dynamics (MD) simulations were able to achieve secondary structure formation for all proteins we tested. For Trp-Cage Miniprotein Construct TC5b (1L2Y) and chicken villin subdomain HP-35, N68H protein (1YRF) all force field runs were very similar to the nuclear magnetic resonance (NMR) structure. Figures 20, 21, and 22 show the comparison of the initial and best folded conformations. The initial conformation (subfigure A, blue) does not have any secondary structure while the final conformation (subfigure B, blue) clearly shows a match of the secondary structure with the NMR conformation (red). From Table 2 we can see that Greedy-proximal A* (GPA*) was able to achieve very close distance to the NMR structure. Tables 3, 5, and 7 show that trajectories with the conformation closest to the NMR conformation were also very short. For example, the longest trajectory for 1L2Y was less than 8 ns, while the longest trajectory for 1YRF was less than 16 ns. All of our lengths of folding trajectories were much shorter than any published to date. Before we move further, we need to mention that we used only one NMR structure as a goal, so it is possible that some of the obtained conformations which did not have a perfect match with the goal were very close to some other NMR conformations.

Table 2: GPA* shortest trajectory lengths along with total sampling lengths. All values in the table are in nanoseconds.

TRP

	RMSD	ANGL	AND	ANDH	XOR	Total time
AMBER 1	7.58	7.9	7.64	7.64	7.62	1887.04
AMBER 2	7.96	5.88	7.36	7.36	7.36	1992.8
CHARMM 1	5.08	5.6	4.56	4.56	4.46	1675.72
CHARMM 2	4.34	3.94	3.94	4.14	4.02	1497.08
GROMOS 1	3.98	4.54	4.42	4.28	4.48	2022.6
GROMOS 2	4.58	4.5	4.62	4.66	4.52	2031.5
OPLS 1	5.06	3.64	3.14	2.3	2.38	1818.62
OPLS 2	7.02	6.34	6.24	6.06	6.22	1743.78

VIL

	RMSD	ANGL	AND	ANDH	XOR	Total time
AMBER	10.8	7.66	3.12	3.46	3.28	11589
CHARMM	7.56	8.34	2.8	3	2.8	9884.4
GROMOS 1	7.22	5.2	4.58	4.88	4.44	12141.52
GROMOS 2	11.58	11.52	11.66	11.66	11.48	592.24
OPLS	15.82	15.52	15.96	2.82	2.94	11052.94

GB1

	RMSD	ANGL	AND	ANDH	XOR	Total time
AMBER	21.36	5.56	2.4	2.4	2.32	11958.54
CHARMM	14.1	7.74	3.22	2.96	3.2	14229.54
GROMOS	15.22	12.7	12.76	12.76	15.4	12636.5
OPLS	9.56	13.28	13.16	13.18	13.18	10752.9

GPA* analysis of 1L2Y

All 1L2Y runs were able to achieve a conformation with the proper secondary and tertiary structure regardless of the force field selected. The average all atom RMSD (AARMSD) was around 1.3 Å across all runs performed, however, mean backbone Root-mean-squared deviation (RMSD) was much lower - 0.85 Å. The only exception is the first run with the OPLSaa force field (OPLS) force field which was able to achieve the RMSD of only 2.569 Å as its smallest all atom RMSD to the NMR conformation. Figures 23 and 24 show the lowest achieved RMSD during the search process and illustrate differences in behavior across different force fields. For the RMSD the trajectory lengths were in the range 3.98 ns - 7.58 ns. Lindorff-Larsen et al. (Lindorff-Larsen et al., 2011) used the OPLSua (Jorgensen et al., 1996) force field with the same temperature (300 K) which are very similar to our parameters, except for using the GB/SA (Qiu et al., 1997) implicit solvent. Their results were in the range 1.5-8.7 μs (Snow et al., 2005), while our longest trajectory with OPLS force field was 7.02 ns, 214-1239 times shorter. Furthermore, the specified that their trajectories achieved at least 2.5-3.0 Å backbone RMSD (BBRMSD), while our final conformations reached 0.8 Å AARMSD or 0.38 Å BBRMSD.

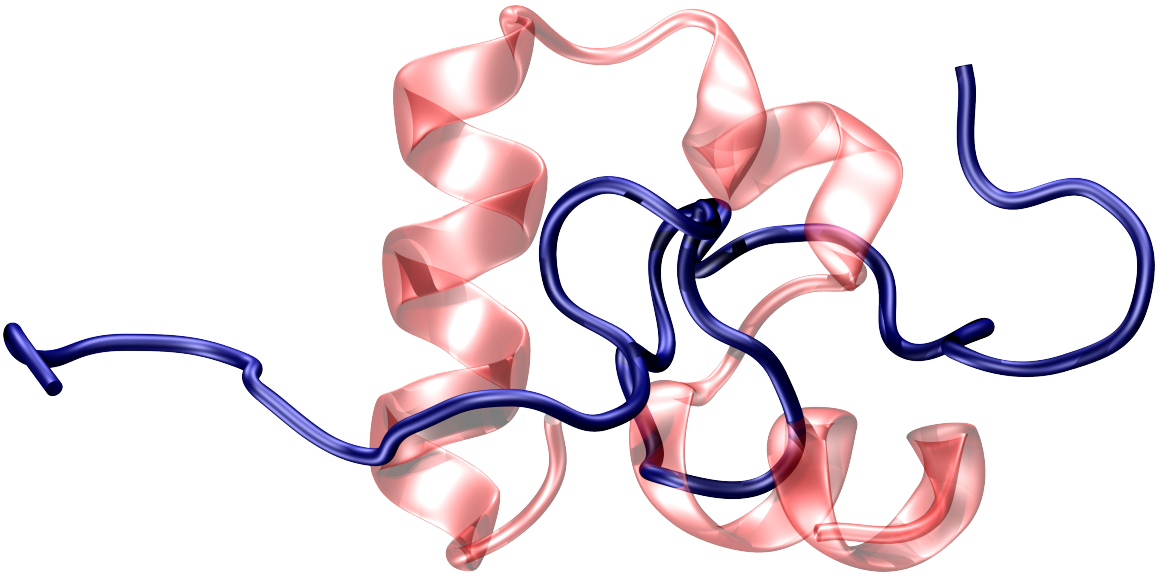
GPA* analysis of 1YRF

All 1YRF runs were able to achieve conformations with proper secondary and tertiary structure regardless of the force field selected. The average AARMSD was around 2.1 Å across all runs performed. However, the mean backbone RMSD was much lower - 1.5 Å. The only exception is the first run with the GROMOS54a7 force field (GROMOS) force field which was able to achieve RMSD of only 3.832 Å as it's smallest AARMSD to the NMR conformation. Figure 25 shows the lowest achieved RMSD during the search process and also illustrates differences in behavior across different force fields. However,

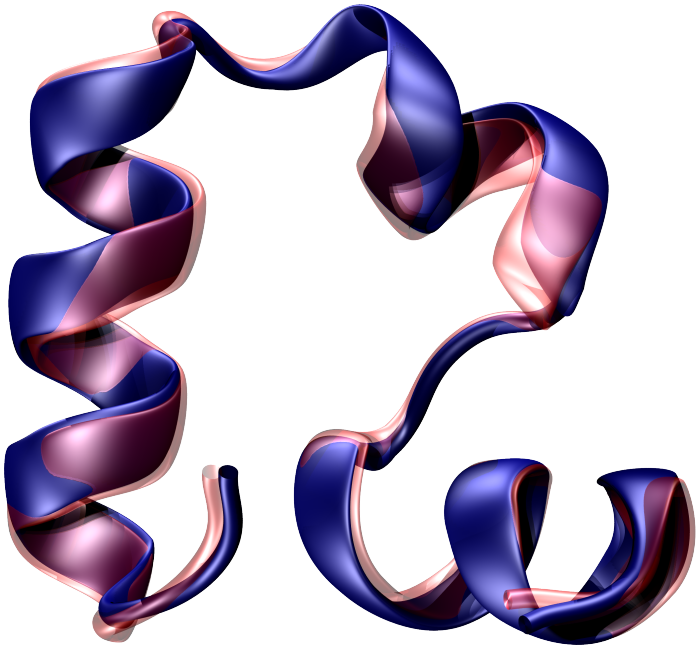
the second run of GPA* with the GROMOS force field was absolutely opposite in effect as compared to the first run, and demonstrated the lowest RMSD across all GPA* runs for the this protein. Furthermore, the second run with the GROMOS force field had the shortest total simulation time 592.24 ns, compared to 11 μs - the average duration of other runs with this protein reported in the literature. For the RMSD metric, trajectory lengths were in the range 7.22 ns - 15.82 ns. Ensign et al. (Ensign et al., 2007) reported 752.6 ns as the fastest average length, folded with the AMBER2003 (Wang et al., 2000) force field at 300 K, but they used the 2F4K Protein Data Bank (PDB) structure which had two extra mutations: K65(NLE) and K70(NLE) to encourage faster folding than our simulated variant. We did not compare RMSD distance directly since they measured RMSD for each alpha helix (3 in total). Our trajectories were 47.6-104 times shorter.

GPA* analysis of 1GB1

For the RMSD, the trajectory length was in the range 9.56 ns - 21.36 ns. Lindorff-Larsen et al. (Lindorff-Larsen et al., 2011) reported folding of the 1MI0 mutant in 65 μs at 350 K with a resulting BBRMSD distance of 1.2 Å. Since higher temperature allows easier passage over the energy barriers, our algorithm would benefit from the higher temperature, but we intentionally did not use such an approach to bring the experiment closer to natural conditions. We expect that conformation with RMSD distance of 1.2 Å would require more folding time, thus being longer than our's (1.75 Å), but not significantly longer. Our trajectory was 3043-6799 times shorter than the best reported.

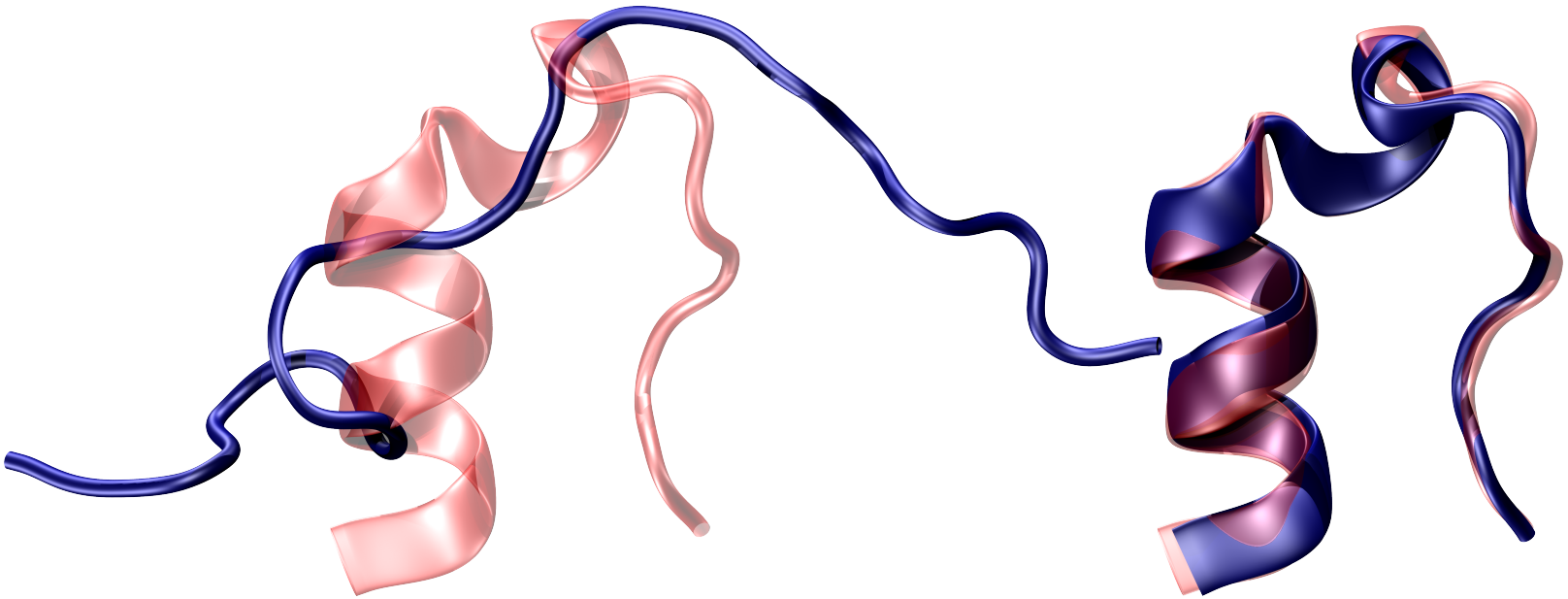


(A) RMSD: 11.083 Å



(B) RMSD: 0.977 Å

Figure 20: Initial (A) and final (B) conformation comparison of 1YRF after folding with GPA*

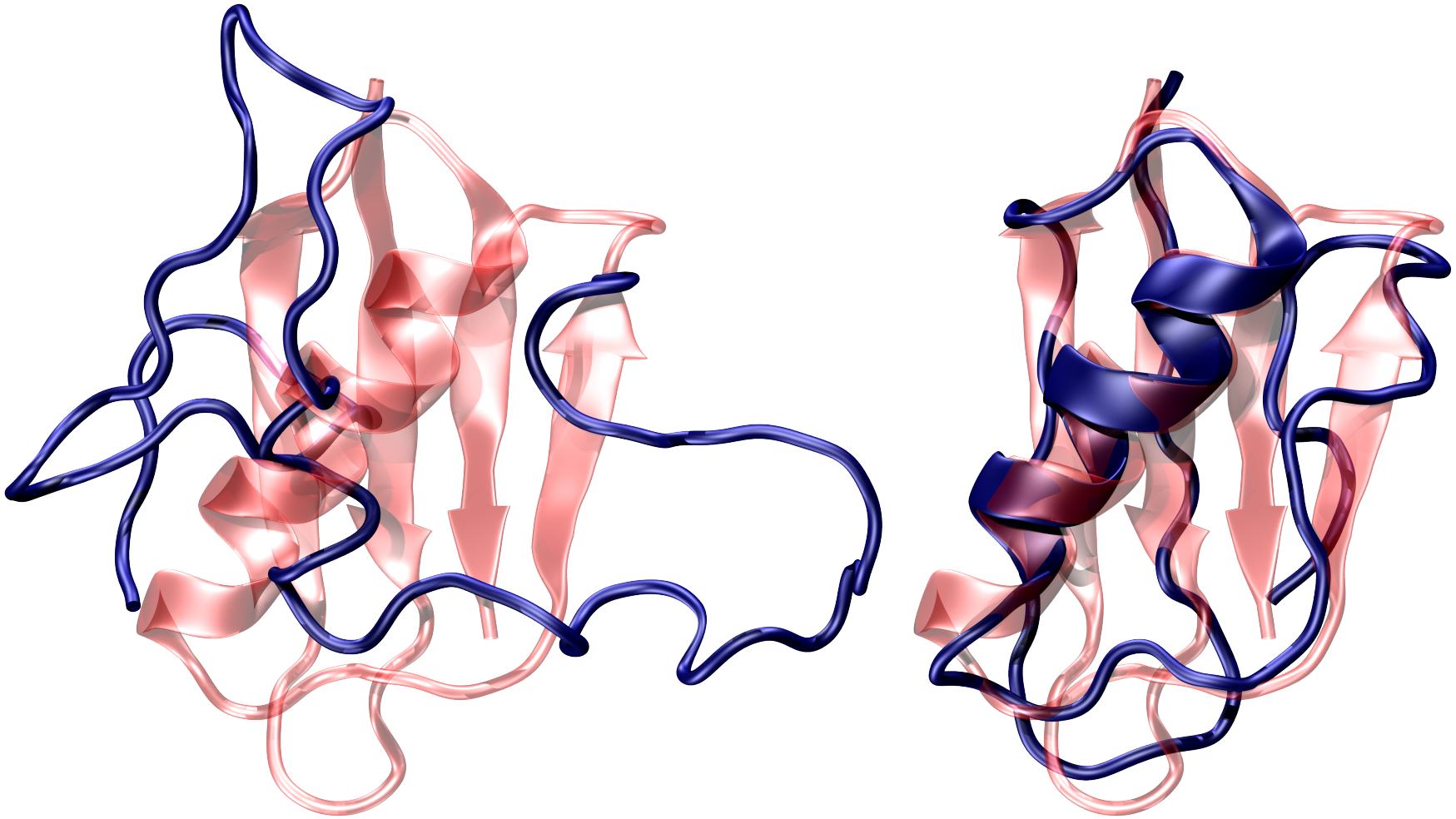


(A) RMSD: 11.083 Å (B) RMSD: 0.977 Å

Figure 21: Initial (A) and final (B) conformation comparison of 1L2Y after folding with GPA*

Table 3: Trajectories which contain smallest AARMSD distance to the NMR conformation of 1L2Y folding with the GPA*. Total time represents the total elapsed time over all simulations.

	RMSD, Å		Mean RMSD, Å	Best RMSD, Å	Total time, ns		Best result reached at, ns	
	(run 1)	(run 2)			(run 1)	(run 2)	(run 1)	(run 2)
AMBER	1.22	1.27	1.24	1.22	1887	1993	1328	1911
CHARMM	1.33	0.89	1.11	0.89	1676	1497	1535	1260
GROMOS	1.08	1.27	1.17	1.08	2023	2032	1862	560
OPLS	2.57	0.80	1.68	0.80	1819	1744	793	1707



(A) RMSD: 13.014Å

(B) RMSD: 2.726 Å

Figure 22: Initial (A) and final (B) conformation comparison of 1GB1 after folding with GPA*

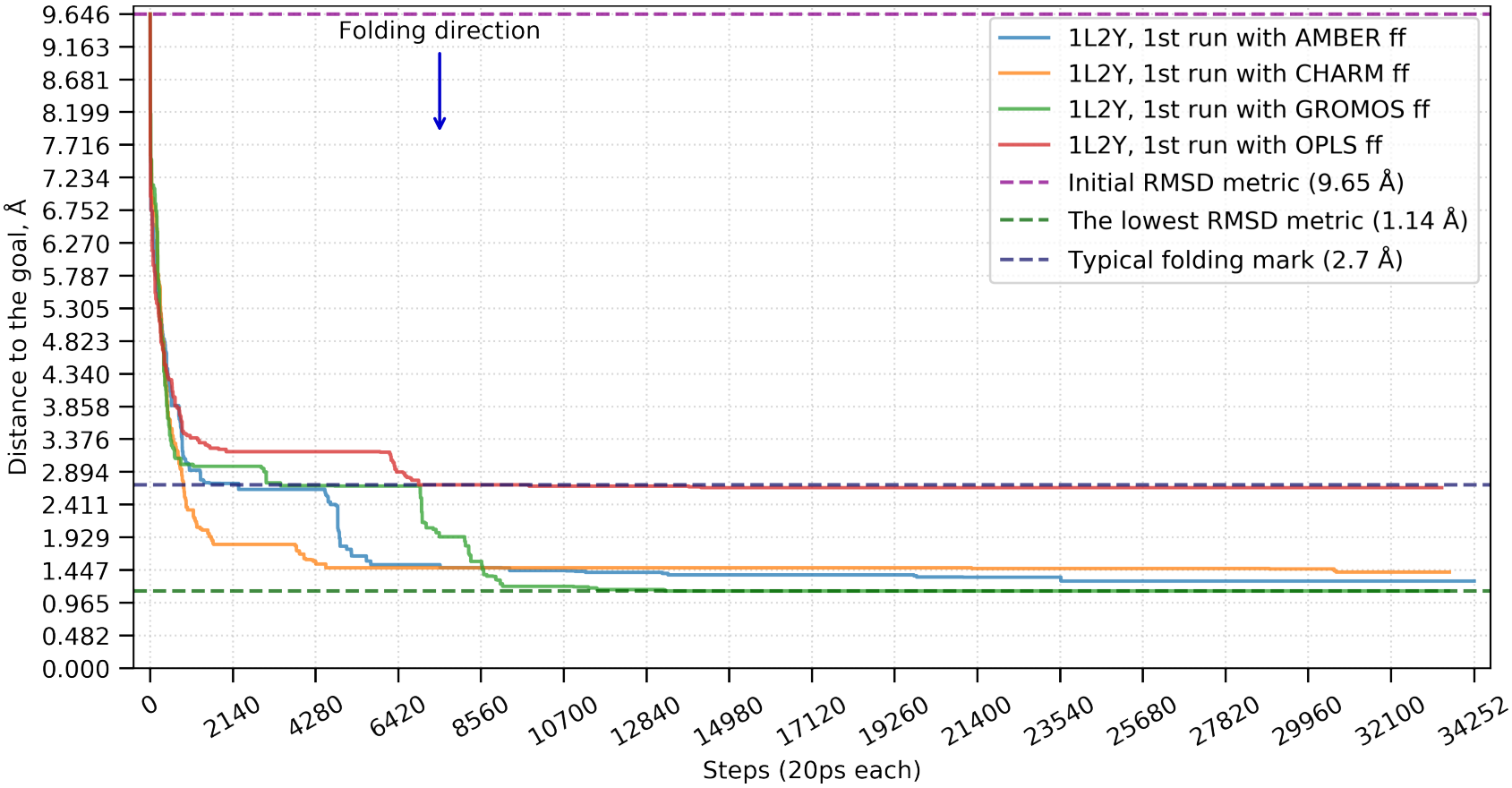


Figure 23: Best reached RMSD metric for the 1L2Y first run with the AMBER (blue), CHARMM (yellow), GROMOS (green), and OPLS (red) force field.

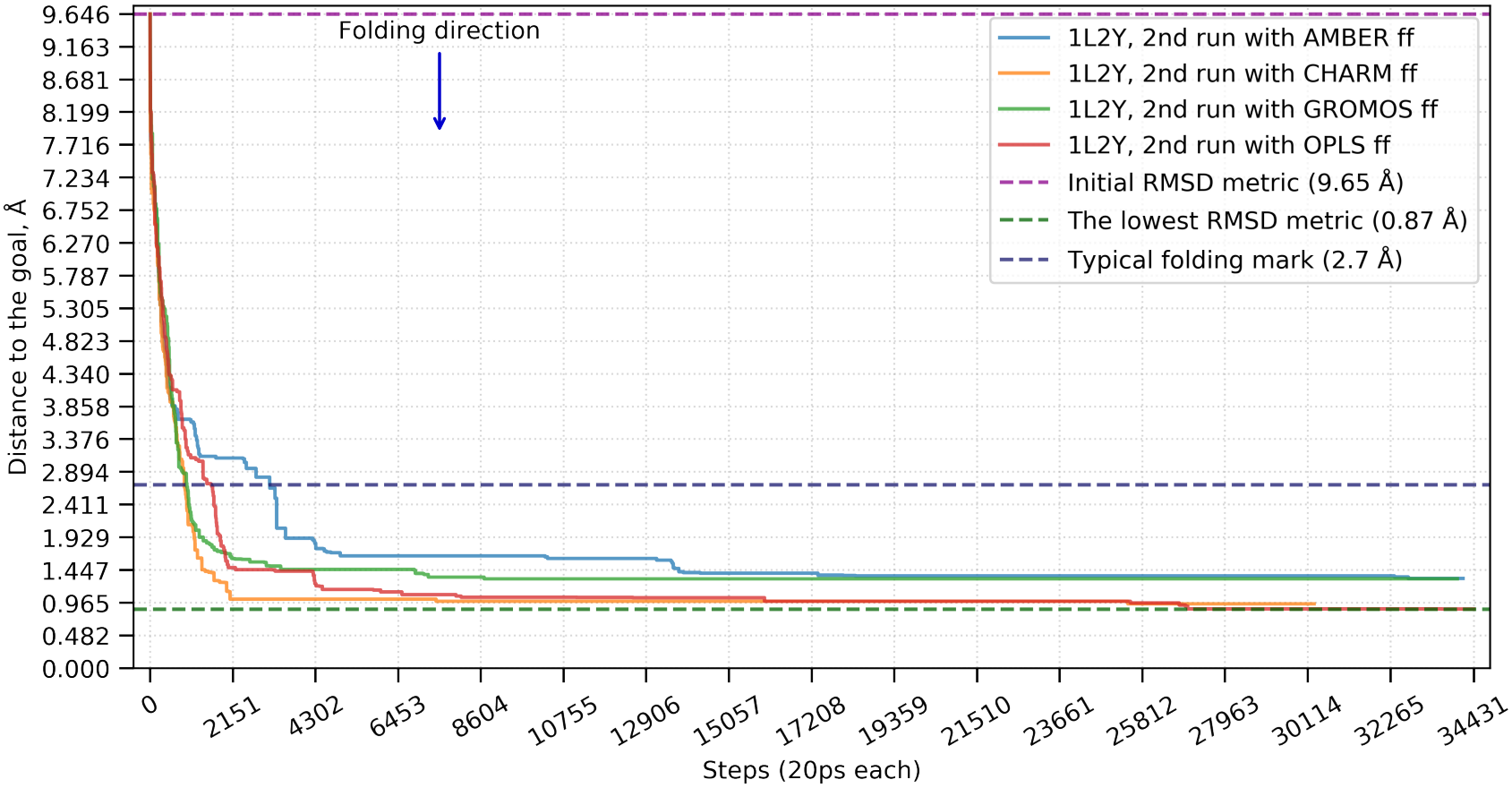


Figure 24: Best reached RMSD metric for the 1L2Y second run with the AMBER (blue), CHARMM (yellow), GROMOS (green), and OPLS (red) force field.

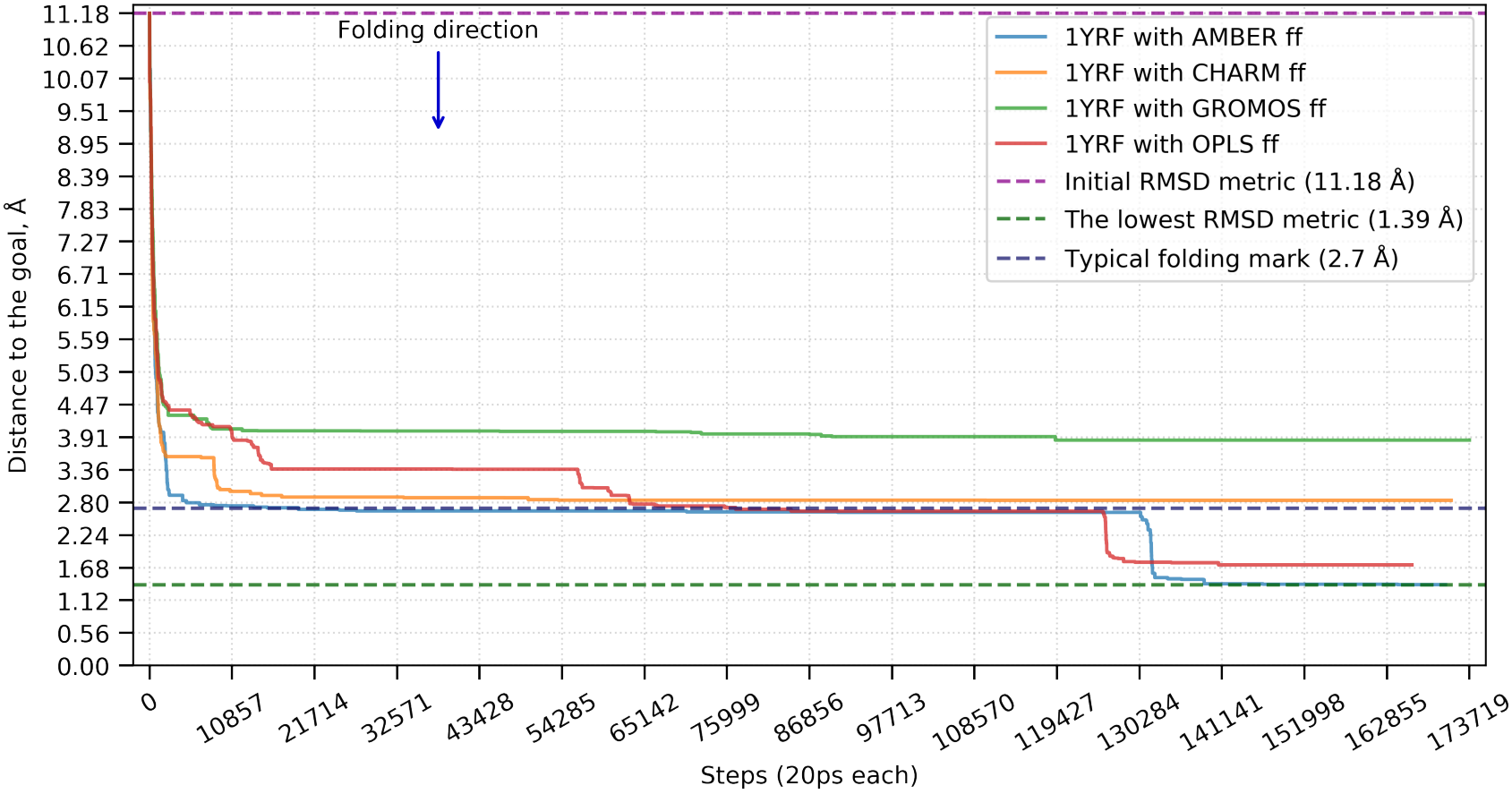


Figure 25: Best reached RMSD metric for the 1YRF first run with the AMBER (blue), CHARMM (yellow), GROMOS (green), and OPLS (red) force field.

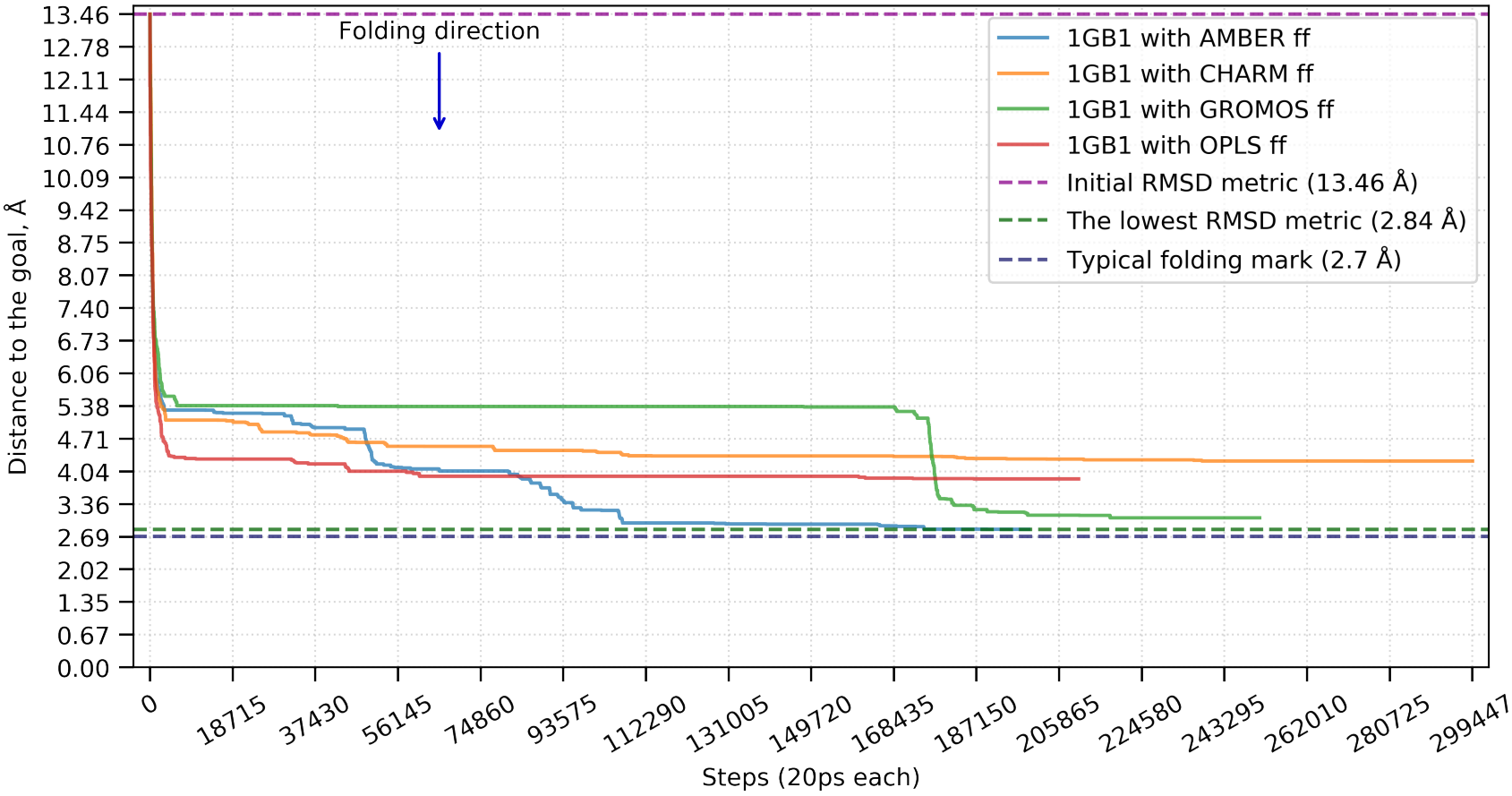


Figure 26: Best reached RMSD metric for the 1GB1 first run with the AMBER (blue), CHARMM (yellow), GROMOS (green), and OPLS (red) force field.

Table 4: Trajectories which contain smallest BBRMSD distance to the NMR conformation of 1L2Y folding with the GPA*. Total time represents the total elapsed time over all simulations.

	BBRMSD, Å		Mean BBRMSD, Å	Best BBRMSD, Å	Total time, ns		Best result reached at, ns	
	(run 1)	(run 2)			(run 1)	(run 2)	(run 1)	(run 2)
AMBER	0.81	0.72	0.77	0.72	1887	1993	1328	1911
CHARMM	0.77	0.60	0.68	0.60	1676	1497	1535	1260
GROMOS	0.75	0.86	0.80	0.75	2023	2032	1862	560
OPLS	1.90	0.38	1.14	0.38	1819	1744	793	1707

Table 5: Trajectories which contain smallest AARMSD distance to the NMR conformation of 1YRF folding with the GPA*. Total time represents the total elapsed time over all simulations.

	RMSD, Å		Mean RMSD, Å	Best RMSD, Å	Total time, ns		Best result reached at, ns	
	(run 1)	(run 2)			(run 1)	(run 2)	(run 1)	(run 2)
AMBER	1.34	n/a	1.34	1.34	11589	n/a	11186	n/a
CHARMM	2.73	n/a	2.73	2.73	9884	n/a	6338	n/a
GROMOS	3.83	0.98	2.41	1.02	12142	592	8305	517
OPLS	1.71	n/a	1.71	1.71	11053	n/a	9358	n/a

Table 6: Trajectories which contain smallest BBRMSD distance to the NMR conformation of 1YRF folding with the GPA*. Total time represents the total elapsed time over all simulations.

	BBRMSD, Å		Mean BBRMSD, Å	Best BBRMSD, Å	Total time, ns		Best result reached at, ns	
	(run 1)	(run 2)			(run 1)	(run 2)	(run 1)	(run 2)
AMBER	0.86	n/a	0.83	1.83	11 589	n/a	11 186	n/a
CHARMM	2.05	n/a	2.05	2.05	9884	n/a	6337	n/a
GROMOS	3.47	0.39	1.93	0.39	12 142	592	8305	517
OPLS	0.89	n/a	0.89	0.89	11 053	n/a	9358	n/a

Table 7: Trajectories which contain smallest AARMSD and BBRMSD distance to the NMR conformation of 1GB1 folding with the GPA*. Total time represents the total elapsed time over all simulations.

	AARMSD, Å	BBRMSD, Å	Total time, ns	Best result reached at, ns
AMBER	2.73	1.75	12 289	11 959
CHARMM	4.16	2.80	15 427	14 230
GROMOS	3.04	1.87	14 545	12 637
OPLS	3.82	2.66	12 085	10 753

The full trajectories comparison

In Figures 27 and 28 we see the dihedral angle distance (ANGL) metric's version of the best trajectory for the 1L2Y protein and GROMOS force field. In Figures 29 and 30 we see the same protein and force field, but the RMSD metric version of the best trajectory. Figure 28 and 30 show a comparison of each metric and the potential energy of the protein. Note that fluctuations between RMSD and ANGL are different, which means that to reduce the RMSD even more, protein needs to increase the ANGL distance and vice versa. However, when we compare both metrics with the protein's potential energy, we see that neither metric is perfect, and, while they have similar trends, fluctuations in metrics' values and potential energy are far from matching at all. Figure 32 shows a comparison between all four force fields during the second run of the 1L2Y protein in terms of the RMSD metric. The increase in RMSD at several steps during execution of the AMBERff99SB-ILDN force field (AMBER) and OPLS trajectories, indicate passing of the energy barrier. Additionally, in Figure 31 we present several examples of the comparison between the best achieved RMSD and past distance from the origin. Finally, we generated thousands of plots which compare different metrics with themselves and potential energy from this analysis and obviously cannot ask the reader to compare them. In the following subsections we summarize the common trends.

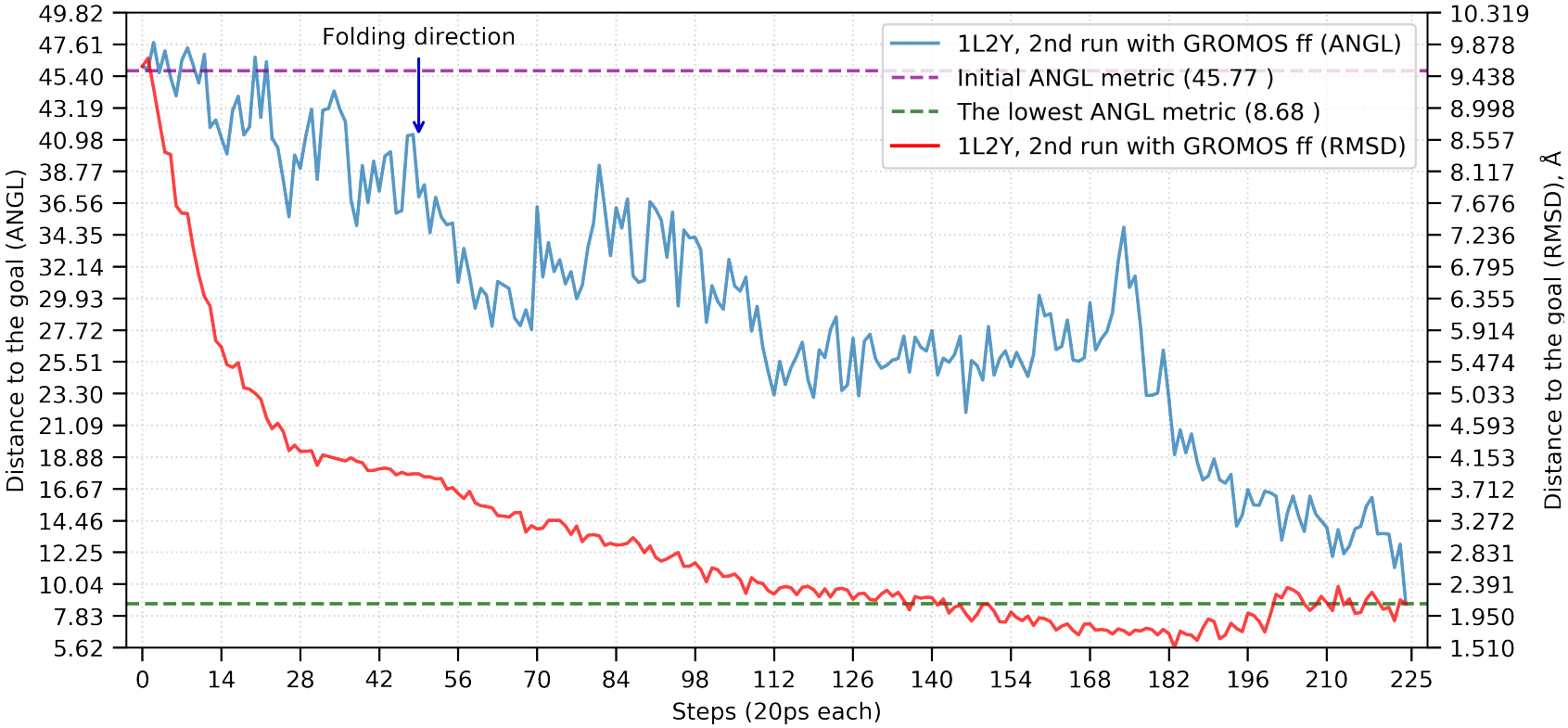


Figure 27: ANGL metric's version of the shortest trajectory during the 1L2Y protein second run with the GROMOS force field. Right axis represents the RMSD values.

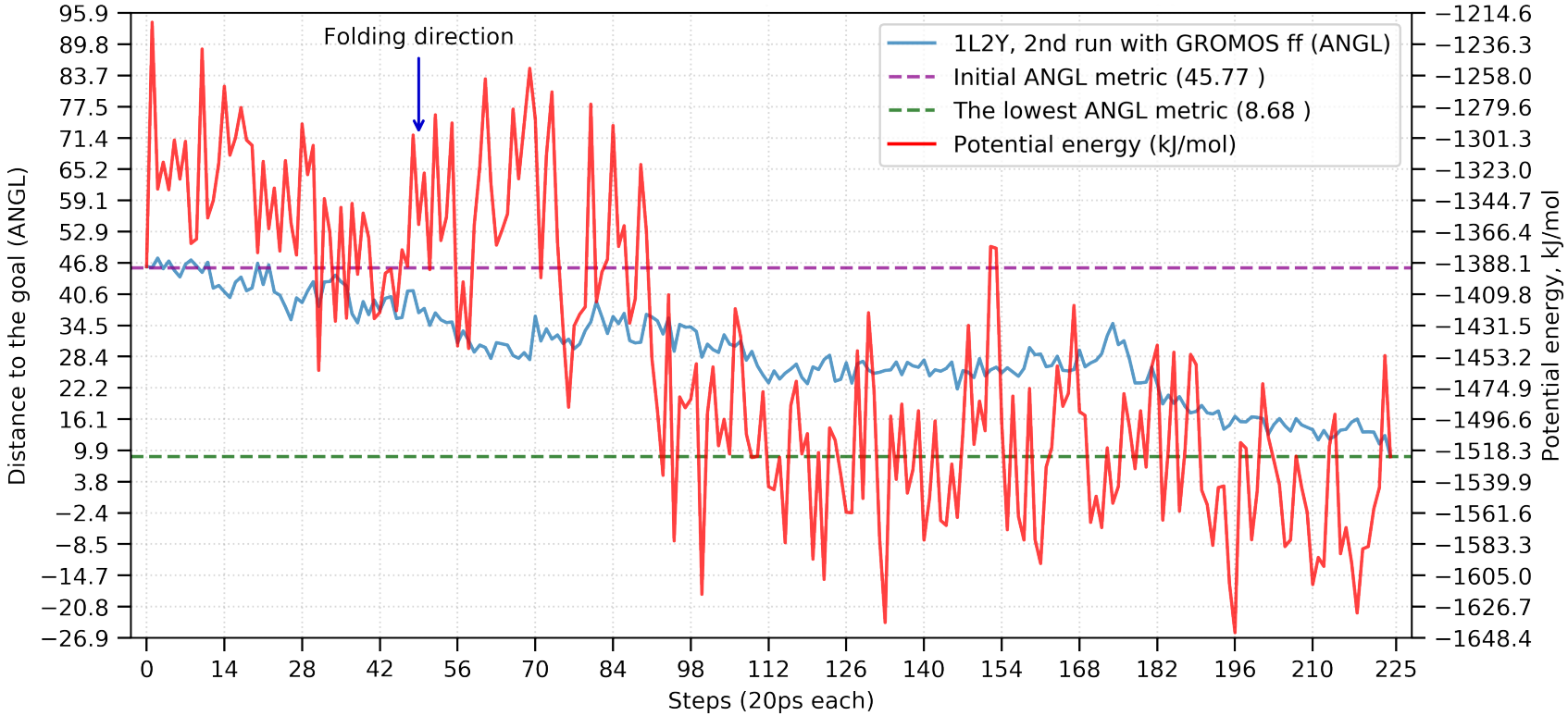


Figure 28: ANGL metric's version of the shortest trajectory during the 1L2Y protein second run with the GROMOS force field. Right axis represents protein's potential energy.

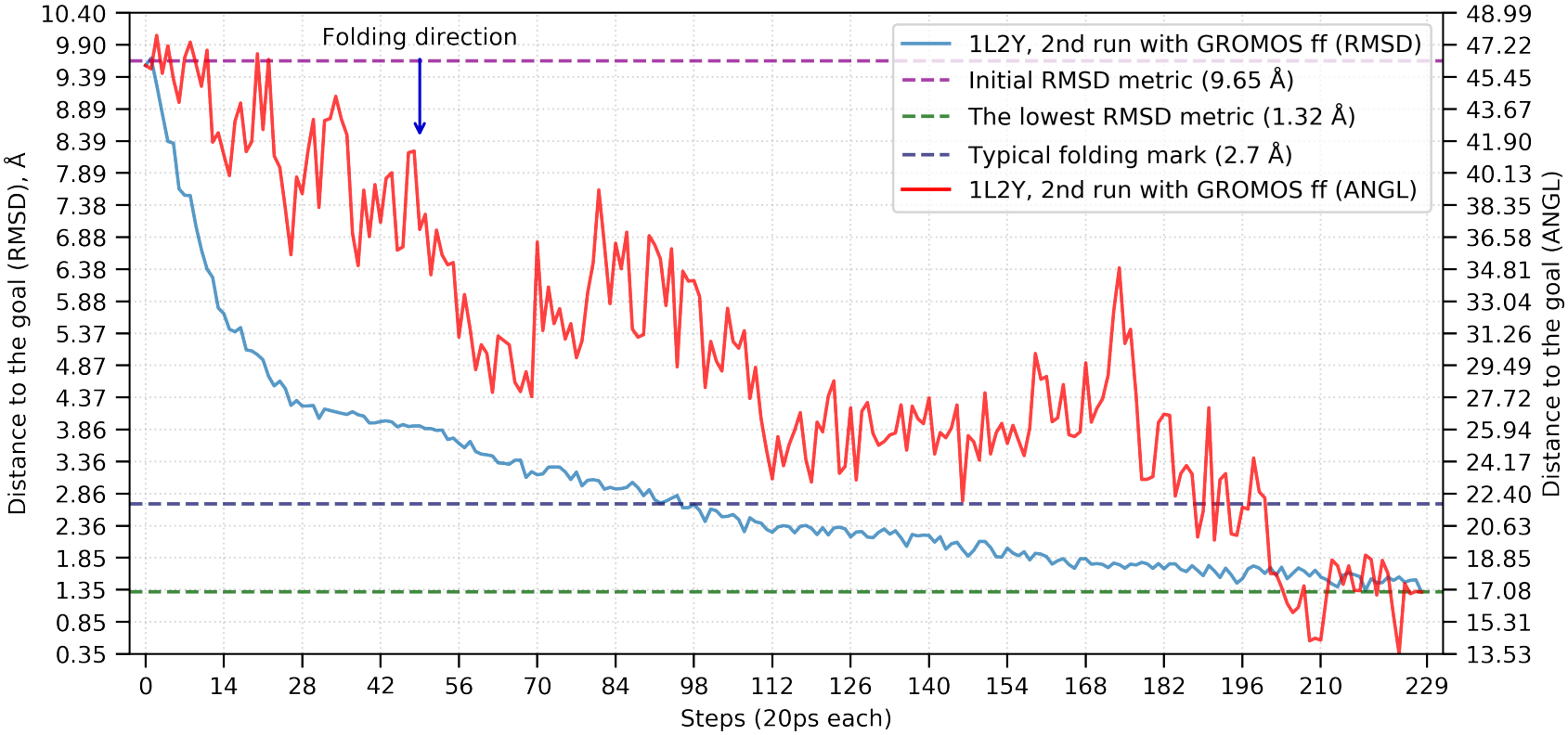


Figure 29: RMSD metric's version of the shortest trajectory during the 1L2Y protein second run with the GROMOS force field. Right axis represents the ANGL values.

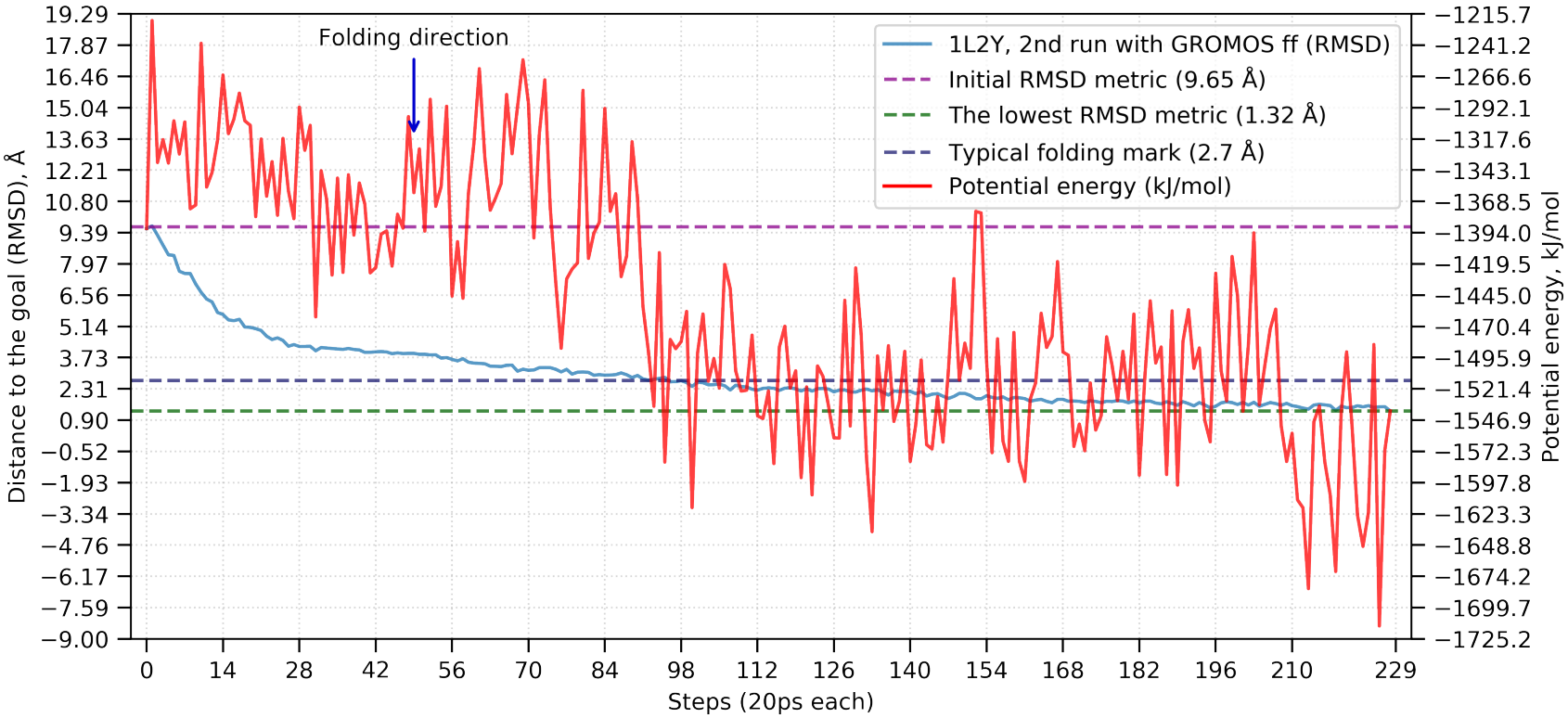


Figure 30: RMSD metric's version of the shortest trajectory during the 1L2Y protein second run with the GROMOS force field. Right axis represents protein's potential energy.

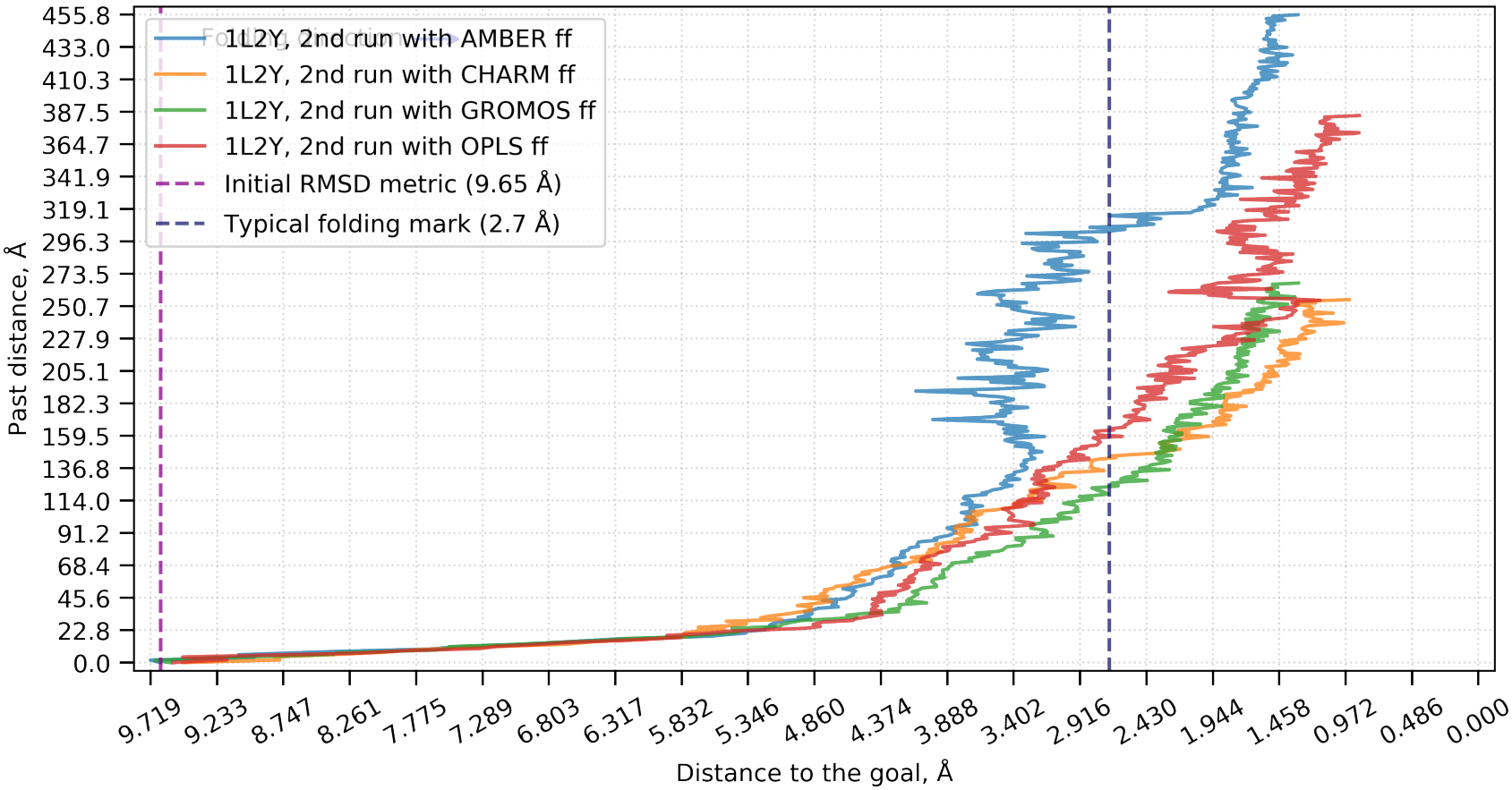


Figure 31: RMSD metric's version of the shortest trajectories during the 1L2Y protein second run as compared to the distance traveled from the origin (initial unfolded conformation).

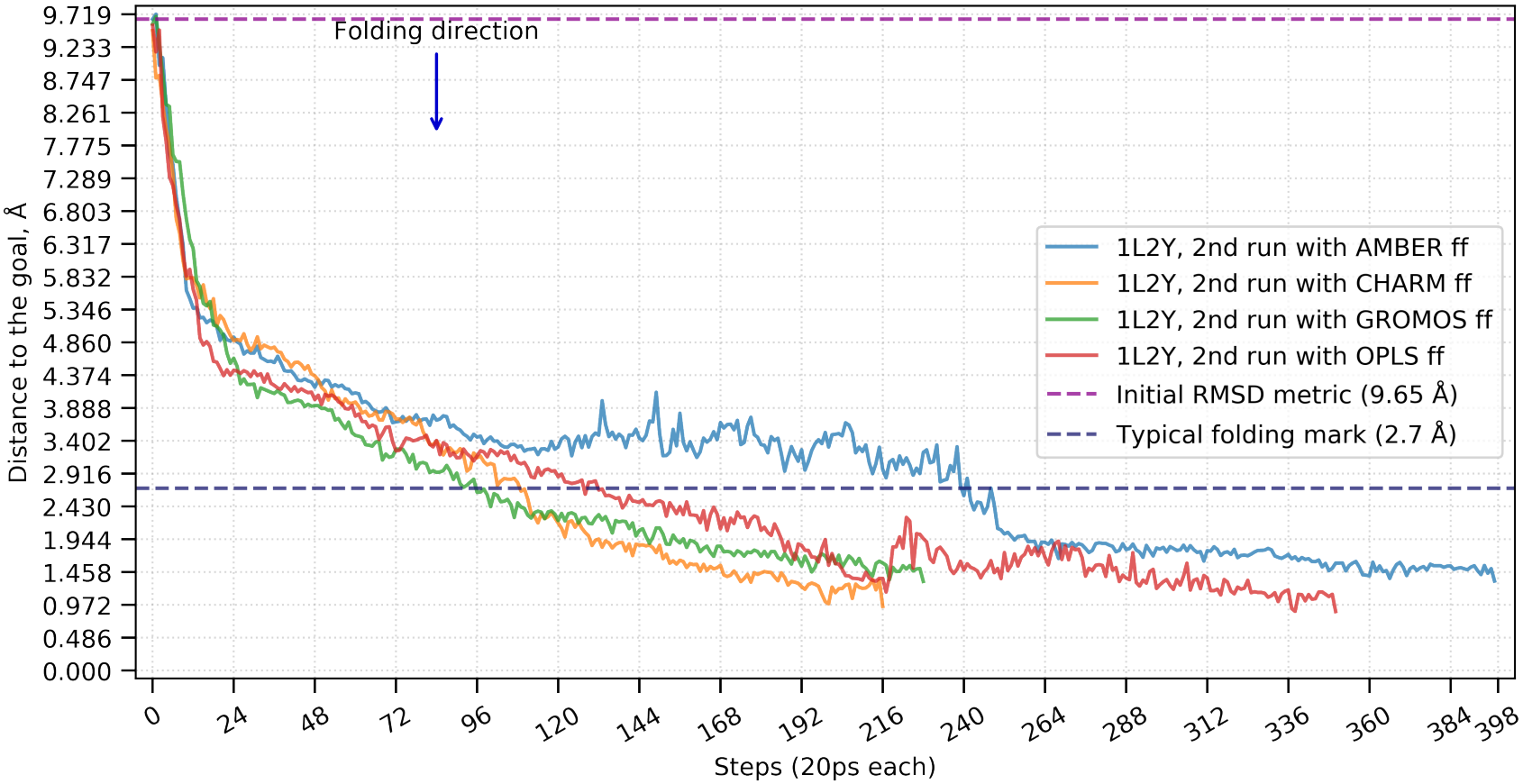


Figure 32: RMSD metric's version of the shortest trajectories during the 1L2Y protein second run for all four force fields.

Comparison of results by protein

General results across all force fields The analysis show that results vary among different force fields, as expected. For example, 1L2Y OPLS had the Sworst average RMSD of 1.766 Å, but the best RMSD among all runs of 0.871 Å at the same time. 1YRF's simulation with the GROMOS force field during the first run gave 3.864 Å which was the worst result, but during second run GROMOS produced 1.018 Å. Different force fields appear to have different strengths (OPLS favors more beta-sheets than other force fields) and our results suggest that all force fields tested can fold proteins quickly using GPA*.

Results for 1L2Y For both normalized and unnormalized approaches, RMSD and ANGL performed much better than the other metrics as given by the percent promotions for each metric. The hydrogen bonds contact map distance agreement (ANDH) was dominating the other metrics in the normalized version. Additionally, the RMSD version of the best trajectory had a slightly higher correlation with potential energy. contact map distance agreement (AND) and ANGL were slightly worse than RMSD, but contact map distance disagreement (XOR) had almost always the worst correlation with the potential energy. However, ANGL metric had the highest determination coefficient with the potential energy, while XOR had the worst determination coefficient with the potential energy.

Results for 1YRF For both normalized and unnormalized approaches, RMSD and ANGL performed much better than the other metrics as given by the promotions during the particular metric. ANDH was dominating the other metrics in the normalized version. It is hard to say which metric's version of the best trajectory had the higher correlation to the potential energy because of similarly high values, but AND's version

on average had slightly higher correlation while RMSD's version had slightly lower correlation. The determination coefficients with potential energy were higher for AND and lower for RMSD.

Results for 1GB1 In unnormalized case, RMSD and ANGL were much better than any other metrics. AND and XOR resulted in the worst perfomance, when combined with GROMOS and OPLS they did not achieve a single promotion. ANDH was in the middle for AMBER and CHARMM36-nov2018 force field (CHARMM), but GROMOS and OPLS did not have any outstanding results. In nomalized case, ANGL was better than RMSD. ANDH was the best when ran with the CHARMM and second when ran with the AMBER force fields.

Comparison by force field

General results across all proteins AMBER and CHARMM had almost identical behavior with promotion steps. OPLS was very close to them as well, but sometimes resulted in a different order of metrics. GROMOS more often had a different order of metrics. We did not find any pattern in potential energy correlation across different force fields. Besides being consistent, AMBER was producing conformation with very low RMSD distance to the NMR conformation which.

General results by metric

We do not provide analysis of the potential energy correlation for immunoglobulin binding domain of streptococcal protein G (1GB1) since all values were negative with no visual pattern which indicates that they are too different.

Results for RMSD 1L2Y favored this metric more for both normalized and unnormalized number of promotions. Trajectory of 1L2Y with this metric had generally one

of the best correlation coefficients, but determination coefficients were either the best or the worst. This metric was the best for 1YRF only in the unnormalized case. The best trajectory for 1YRF according to current metric had generally one of the worst correlation and determination coefficients. This metric was the best for 1GB1 in unnormalized case, but only second in normalized case.

RMSD was doing well with all studied force fields, except the 1YRF / GROMOS where it was worse than ANGL. Simulation with the GROMOS force field after normalization showed better results of ANGL than RMSD. In most it had either the best or the worst correlation and determination with potential energy without visible pattern that depends on the force field.

Results for ANGL This metric was typically the second best (after RMSD) in generation of the promotional steps for both runs of the 1L2Y protein regardless of the normalization. ANGL's version of the best trajectory correlation with the potential energy is slightly worse than RMSD, but the determination coefficient with the potential energy was the highest. This metric was the best for 1YRF in the normalized case and the second best (after RMSD) in the unnormalized case. This metric's version of the best trajectory correlation and determination with the potential energy had a very high variation, so we did not see any pattern. ANGL metric was the second for 1GB1 in unnormalized case and the first in normalized case.

This metric had the second place with all studied force fields, except the GROMOS where sometimes (1YRF normalized and unnormalized) it was the first. Often it was either the second or the first in correlation and determination without a visible pattern that depends on the force field.

Results for ANDH Current metric was third in generation of the promotional steps for both runs of the 1L2Y protein regardless of the normalization. ANDH's version of

the best trajectory correlation for both runs was slightly worse than ANGL and AND but the determination coefficient with the potential energy was much worse than ANGL. The same situation was with 1YRF protein. This metric's version of the best trajectory correlation was second but stable, however determination coefficient with the potential energy was third, worse than AND. This metric was the in the middle for 1GB1 for both normalized and unnormalized cases.

It was the third metric (after ANGL) for all force field, except the GROMOS where it often exchanged the third place with XOR. Often it was either the second or the third in correlation and determination without a visible pattern that depends on the force field.

Results for AND This metric was one of the worst metrics in generation of the promotional steps for 1L2Y regardless of the normalization. AND metric's version of the best trajectory for 1L2Y had a slightly better correlation and determination with the potential energy than ANDH during both runs. Current metric was one of the worst metrics in generation of the promotional steps for 1YRF regardless of the normalization. This metric's version of the best trajectory for 1YRF had one of the highest correlation and determination coefficients with the potential energy. AND metric was the one of the worst for 1GB1 in both normalized and unnormalized cases.

This metric was the worst regardless of the force field. Often it was either the second or the third in correlation and determination without a visible pattern that depends on the force field.

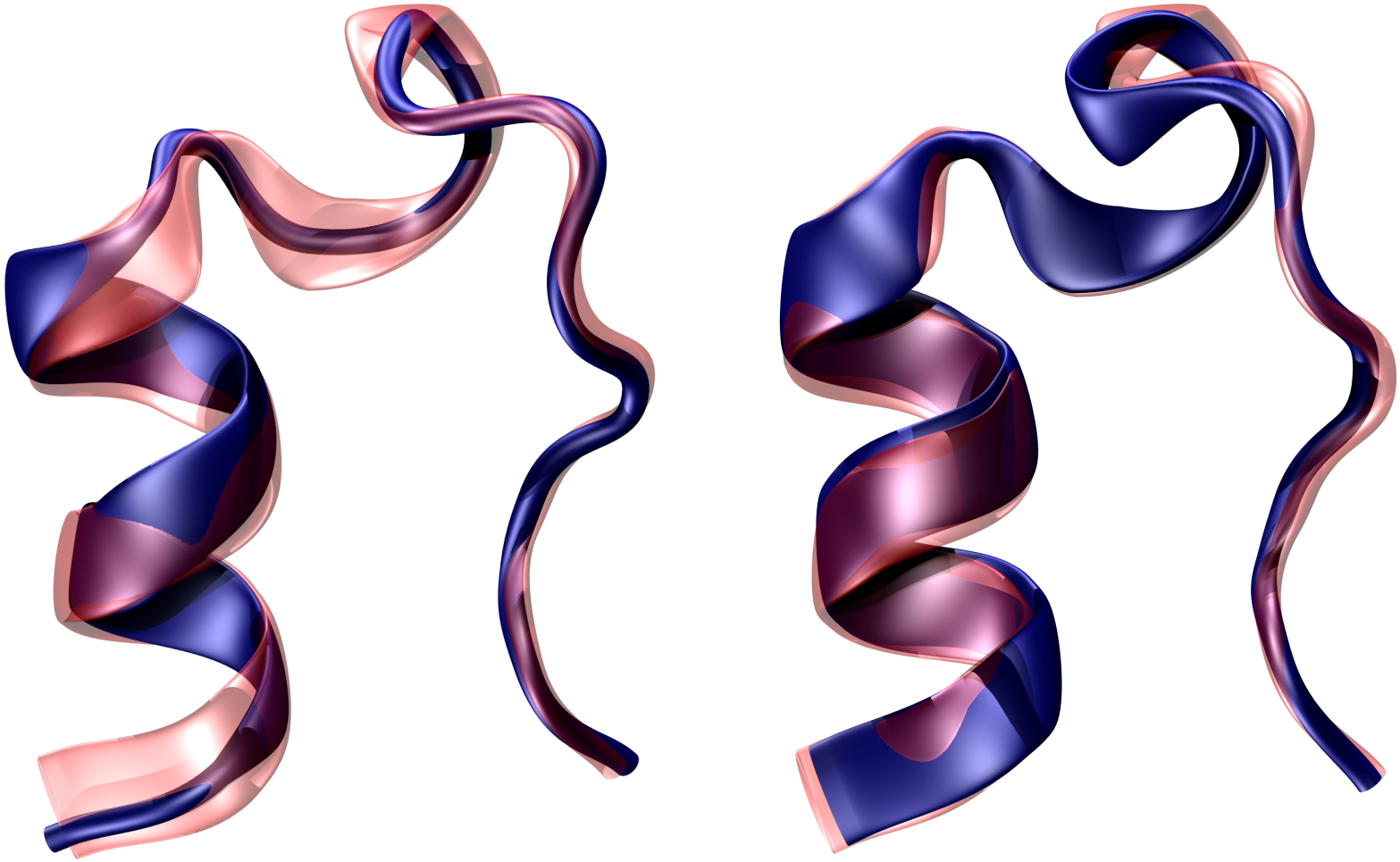
Results for XOR This metric was one of the worst metrics for 1L2Y regardless of the normalization. XOR metric's version of the best trajectory for 1L2Y had the worst correlation and determination coefficients with the potential energy. For 1YRF it was slightly better than AND. Current metric's version of the best trajectory for 1YRF had too much variation in correlation and determination coefficients with the potential energy

to determine any patterns. XOR metric was the worst for 1GB1 in both normalized and unnormalized cases.

Overall, this metric was fourth out of five studied metrics. However, GROMOS often had it to be the third. Often it was either the second or the third in correlation and determination without a visible pattern that depends on the force field.

Demonstration of the Metric Utility

In the Methods section we mentioned that RMSD is not the only metric used to measure the distance between the two conformations. Figure 33 includes the closest to the NMR structure conformations by according to RMSD (A) and ANGL (B). While the conformation (A) has a smaller RMSD (1.051 Å), visually conformation B is much better (1.074 Å). AARMSD between the two conformations is 1.730 Å, BBRMSD between them is 0.763 Å. We have to mention that 1L2Y had several slightly different NMR conformations and it is very possible that both proteins represent two different conformations, however, we used only one NMR conformation as a goal. AARMSD between the NMR conformations was 1.2 - 2.3 Å and BBRMSD between them was 0.33 - 1.4 Å.



(A) RMSD: 1.051 Å

(B) RMSD: 1.436 Å

Figure 33: Example of different metrics "smallest" distance. Red color is the NMR structure, blue color is the current conformation. (A) represents the best trajectory according to the RMSD metric. (B) represents the best trajectory according to the ANGL metric.

The REMD results and comparison with GPA*

Our key results listed in the Table 8 show that Replica-Exchange Molecular Dynamics (REMD) with a similar number of MD steps was not able to achieve lower RMSD distance to the NMR conformation, and resulted in much longer folding trajectories. Furthermore, GPA* *average results* in most of the cases were close to the *best* REMD values.

Another interesting fact is that while with GPA* RMSD was going straight down (Figure 36), REMD resulted in very short bursts of low RMSD (Figure 34 and 35). Additionally, GPA* by design was able to recreate a compact sequence of frames that lead from the fully unfolded state to the state with the smallest RMSD or any other metrics, which is much easier to analyze and amenable to additional calculations such as umbrella sampling which is needed for calculation of Gibbs free energies (Kumar et al., 1992).

Table 8: GPA* and REMD comparison of RMSD for the AMBER force field. Time represents total time of all simulations

		GPA*			REMD		
	Initial RMSD, Å	Best RMSD, Å	Average RMSD, Å	Time, ns	Best RMSD, Å	Average RMSD, Å	Time, ns
1L2Y (run 1)	8.97	1.29	3.85	1887	3.46	7.00	1940
1L2Y (run 2)	8.97	1.32	3.21	1993	2.91	6.85	2000
1YRF	10.79	1.39	5.49	11589	2.62	8.67	10000
1GB1	13.01	2.73	6.87	12289	7.14	12.02	18000

1L2Y

Even for the worst of the two GPA* runs, the RMSD result (1.322 Å, Table 3) was much lower than the best REMD result (2.914 Å, Table 8) with the same force field. It is interesting to mention, that in order to reach 2.914 Å, GPA* needed to perform only 85.86 ns of 1887.04 ns of total time during the first run, or 195.02 ns of 1992.8 ns total

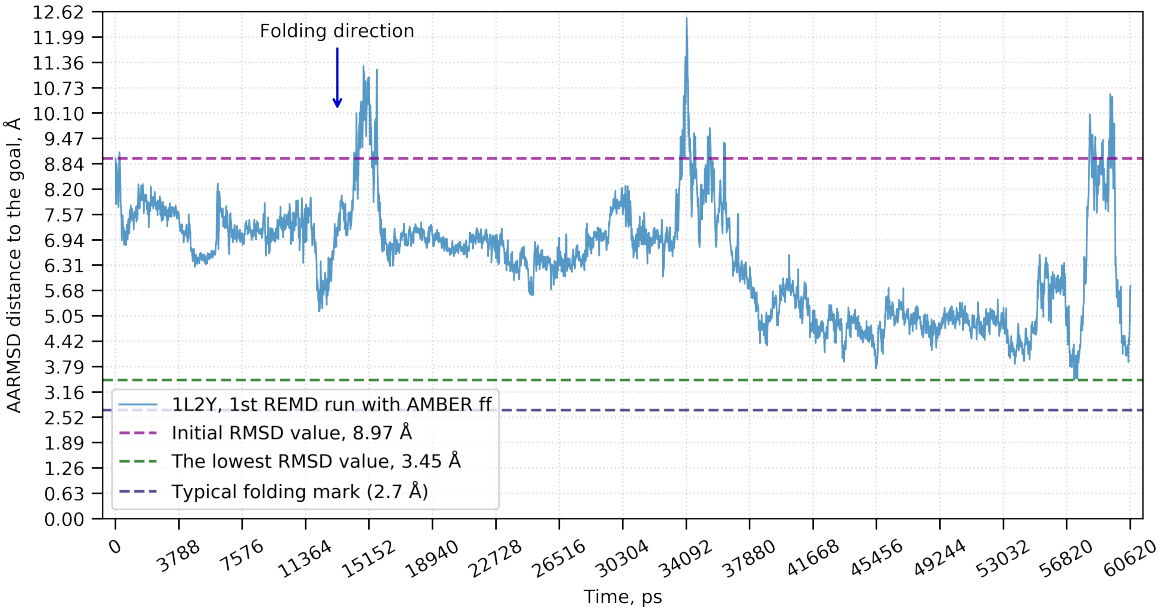


Figure 34: Behavior of the REMD algorithm while folding 1L2Y during the first run with AMBER force field.

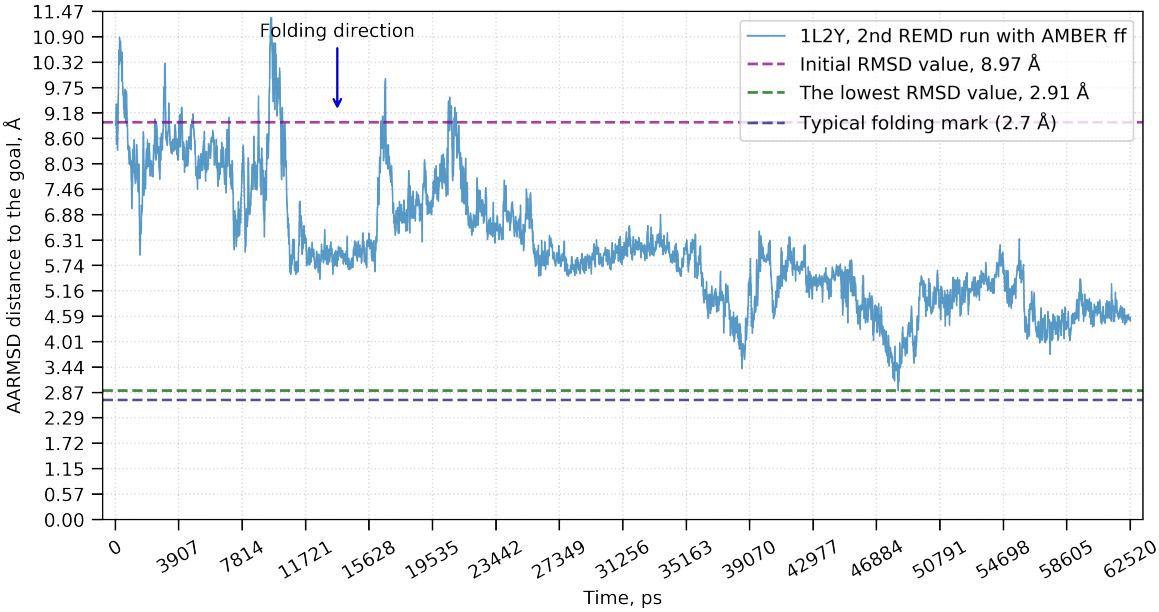


Figure 35: Behavior of the REMD algorithm while folding 1L2Y during the second run with AMBER force field.

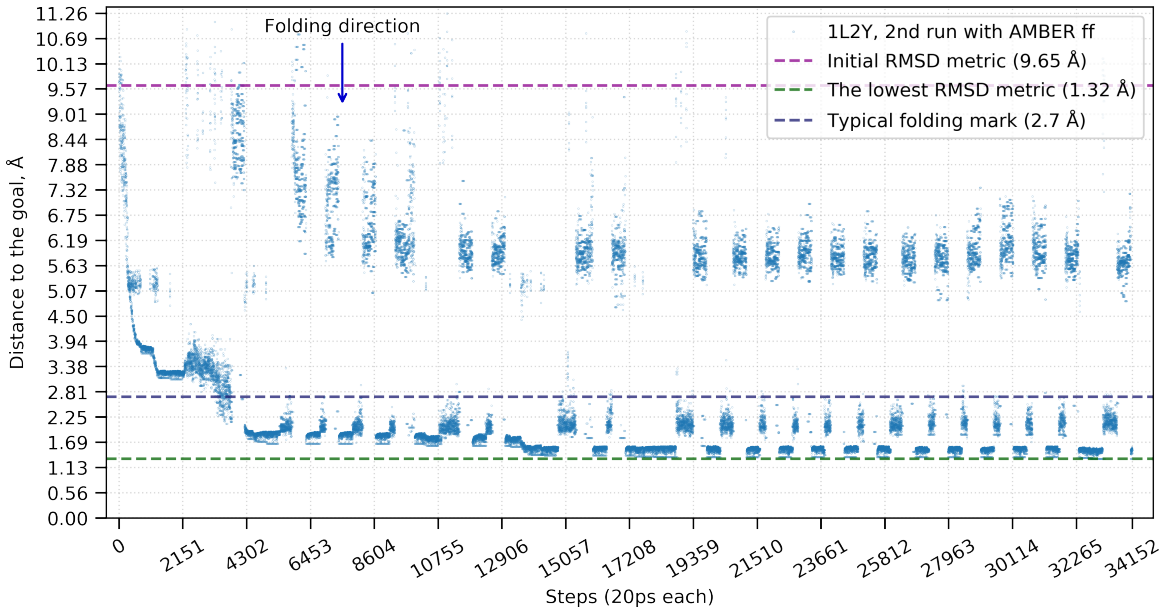


Figure 36: Behavior of the GPA* algorithm while folding 1L2Y during the second run with AMBER force field. Second run was selected as the one with worse of two runs.

time during the second run. Therefore, GPA* exhibited 2.4 times smaller AARMSD while having 6.4 times shorter trajectories than REMD with the same force field.

1YRF

The GPA* RMSD result (1.385 Å, Table 5) was much lower than the best REMD (2.615 Å, Table 8) with the same force field. It is interesting to mention, that in order to reach 2.615 Å, GPA* needed to perform only 8948.6 ns of 11589 ns of total time. An additional 2237.62 ns were needed to reach the lowest result. Therefore, GPA* exhibited 1.9 times smaller AARMSD while having 22.4 times shorter trajectories than REMD with the same force field.

1GB1

The GPA* RMSD result (3.935 Å, Table 7) was much lower than the best REMD (7.139 Å, Table 8) with the same force field. It is interesting to mention, that in order to reach 7.139 Å, GPA* needed to perform only 183.4 ns of 12288.76 ns of total time. An additional 11775.14 ns steps were needed to reach the lowest result, which matched our termination criterion in the Methods section. Therefore, GPA* exhibited 2.6 times smaller AARMSD while having 8.6 times shorter trajectories than REMD with the same force field.

Table 9, 10, and 11 show that GPA* was not only able to generate much shorter trajectories (17-30 times) compared to REMD, but also spent less total computational time to achieve the same RMSD to the NMR conformation. There was one exception though: 1YRF total time needed to achieve the same RMSD result was only 0.8 of the total REMD computational time. Figure 37 shows that GPA* reached an energy barrier of 2.7 Å at 200 ns and spent most of the time trying to pass it, which occurred around 8948.6 ns. If not for this barrier, 1YRF would achieve almost the same RMSD distance to the NMR 38.6 times faster compared to REMD.

Therefore, GPA* can still exhibit such anomalies even if they are much less likely. This is expected since our sampling mechanism remains to be the standard MD.

Table 9: Comparison of the shortest AARMSD distances to the NMR structure obtained with GPA* and REMD. Length represents the length of the folding trajectory

	REMD		GPA*		RMSD	Length
	Best RMSD, Å	Reached by, ns	Best RMSD, Å	Reached by, ns	Ratio	Ratio
1L2Y	2.91	48.2	1.22	7.6	2.4	6.4
1YRF	2.62	241.4	1.34	10.8	1.9	22.4
1GB1	7.14	183.4	2.73	21.4	2.6	8.6

Table 10: Comparison of the common smallest AARMSD distances to the NMR structure reached with GPA* and REMD. Length represents the length of the folding trajectory

	REMD		GPA*	
	Common RMSD, Å	Reached by, ns	Reached by, ns	Ratio
1L2Y	2.91	48.2	2.72	17.7
1YRF	2.62	241.4	7.96	30.3
1GB1	7.14	183.4	0.08	2292.5

Table 11: Total simulation time spent before the common smallest AARMSD distances to the NMR structure were reached.

	REMD			GPA*	
	Total replicas	Reached at, ns	Total time, ns	Total time, ns	Ratio
1L2Y	30	48.2	1446.6	85.9	16.9
1YRF	32	241.4	7725.4	8948.6	0.9
1GB1	36	183.4	6602.4	60.5	109

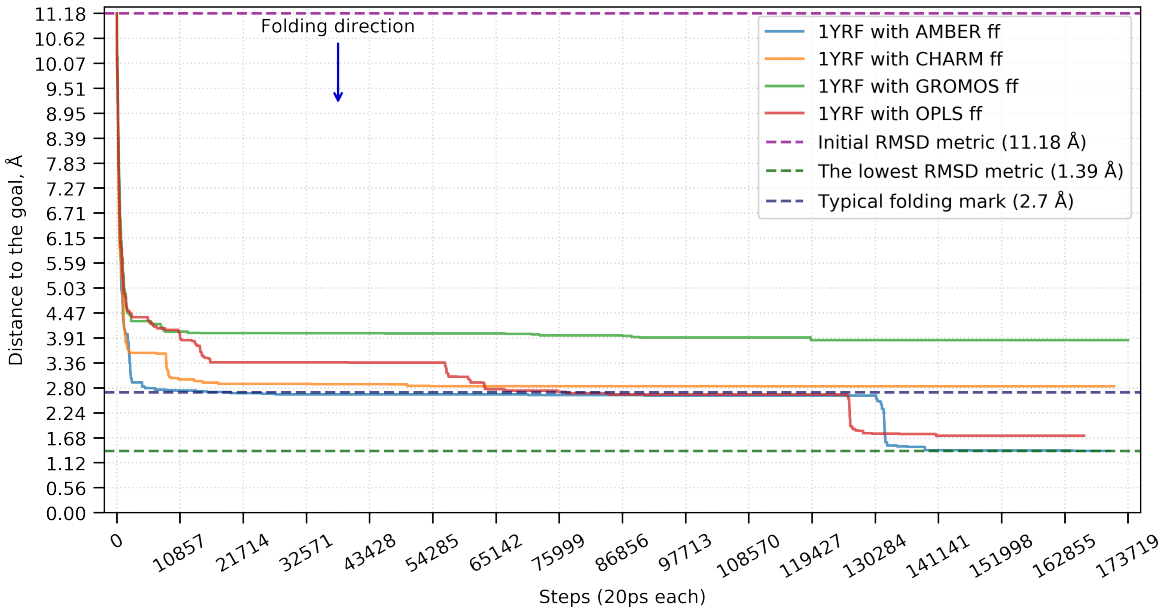
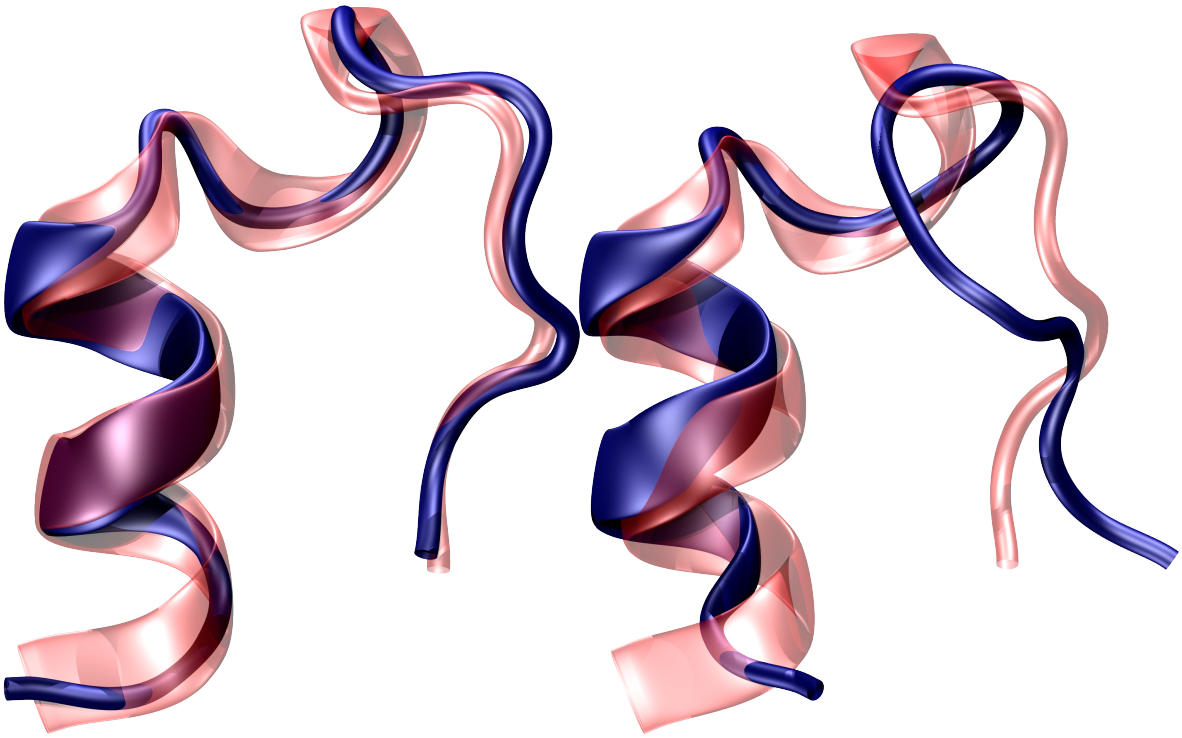


Figure 37: GPA* performance of the RMSD during the 1YRF folding process.

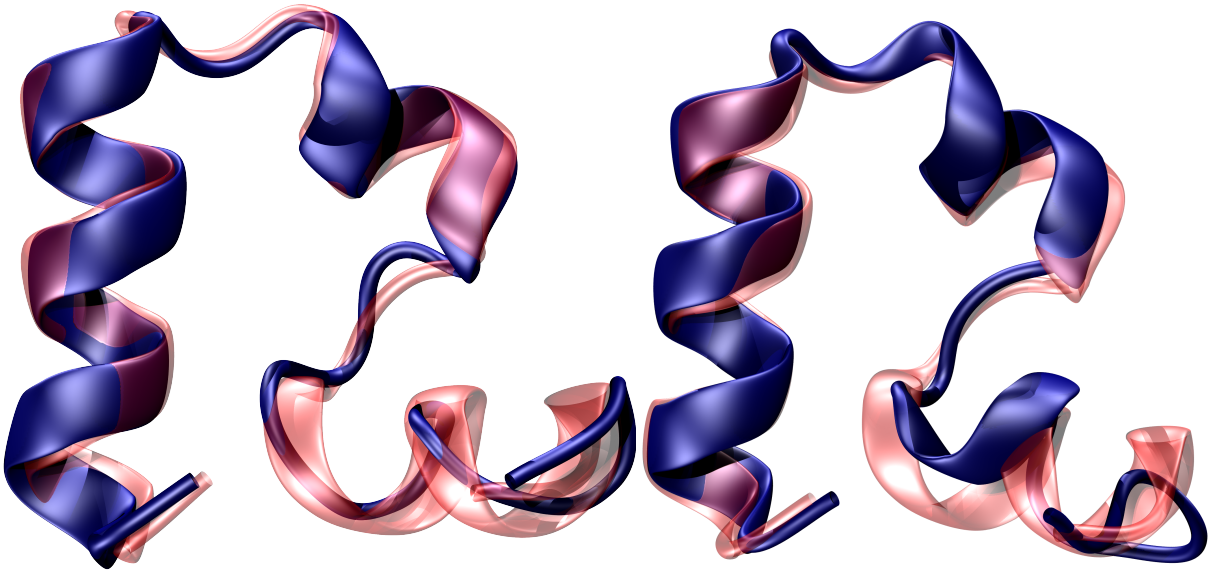
Figures 38 and 39 show the final conformations of GPA* (left) and REMD (right). Red color in both figures represents the NMR structure. While both methods contain most secondary structures, GPA* in all cases shows more complete secondary structures.

The complete list of REMD results which contain the best RMSD achieved by each replica, can be found in Table 33.



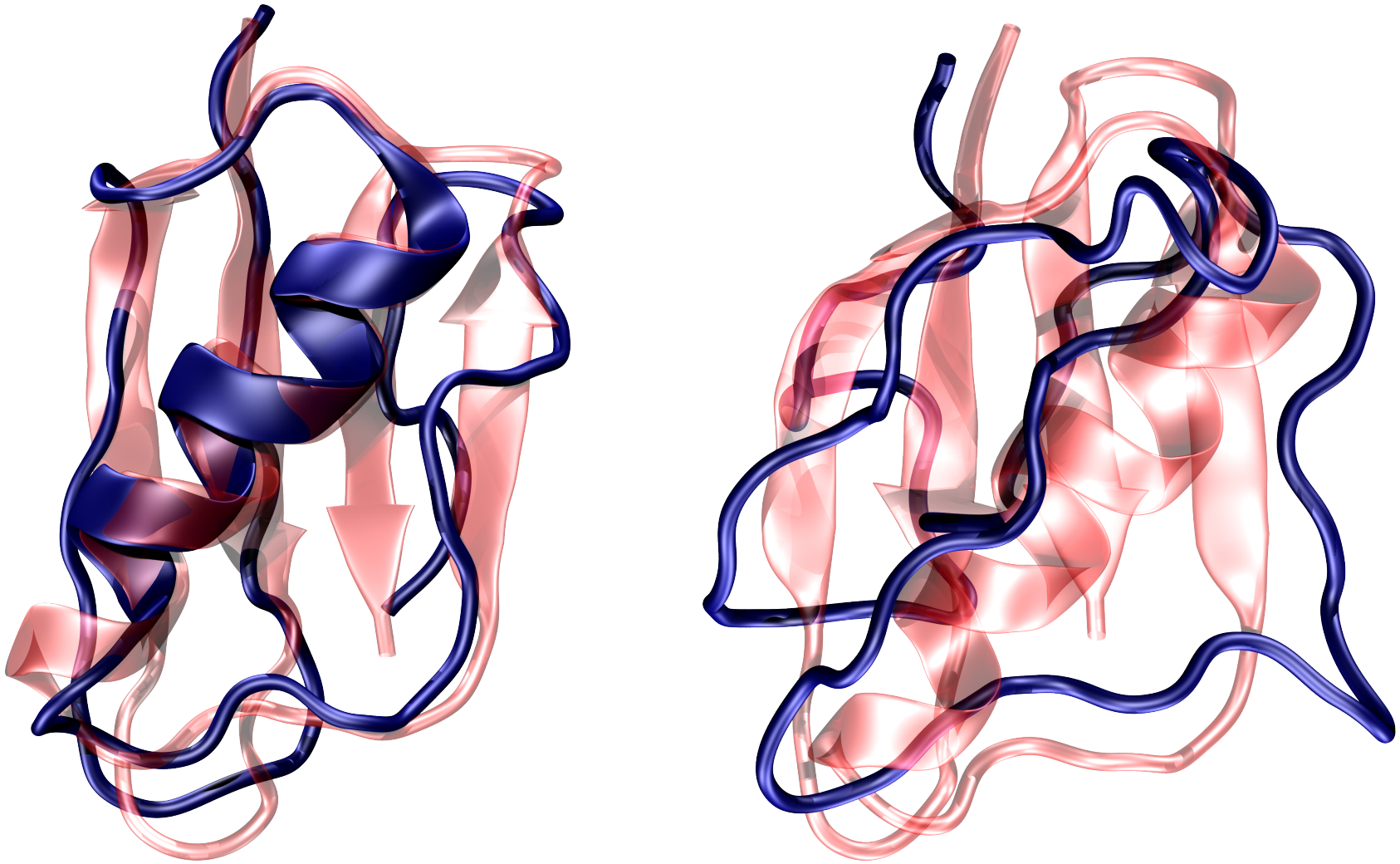
(A) GPA*. RMSD: 1.285 Å (B) REMD. RMSD: 2.914 Å

Figure 38: Best achieved conformations of 1L2Y achieved with GPA* (A) and REMD (B) obtained with the AMBER force field



(A) GPA*. RMSD: 1.342 Å (B) REMD. RMSD: 2.615 Å

Figure 39: Best achieved conformations of 1YRF achieved with GPA* (A) and REMD (B) obtained with the AMBER force field



(A) GPA*. RMSD: 2.726 Å

(B) REMD. RMSD: 7.139 Å

Figure 40: Best achieved conformations of 1GB1 achieved with GPA* (A) and REMD (B) obtained with the AMBER force field

SMD results and comparison with GPA*

For all proteins the smallest RMSD was achieved with 80-90 kJ/mol of pulling force (Table 12). The lower values were generally not as productive. However, even 10-20 kJ/mol was enough to fold the protein to a reasonable RMSD distance from the NMR conformation. If the force was too high, proteins got pulled together too quickly without time to adapt the sidechain orientations and were stuck in an artificial energy well, where the force field and harmonic potential were pulling in opposite directions. It might be desirable to implement an algorithm that can apply a variable force depending on the protein conformation thus reducing the amount of the artificial bias introduced by the method, but discussion is outside the scope of this work. While we were working with proteins which have a very simple folding trajectory, sequences of the conformations during the folding process by Steered Molecular Dynamics (SMD) were not visually similar to any generated by GPA* trajectories. This indicates that the artificial force significantly modifies the folding trajectory. Furthermore, Figure 41 and 42 show that even high force magnitudes may hit artificial energy barriers which prevent further folding. Detailed results about each force used during the simulation may be found in Table 34.

Table 12: SMD simulation smallest RMSD distance to the NMR conformation for the 1L2Y, 1YRF, and 1GB1 proteins. Simulation duration: 2 ns.

	Initial	Final		Force
	AARMSD, Å	AARMSD, Å	BBRMSD, Å	kJ/mol
1L2Y	8.970	1.044	0.572	90
1YRF	10.794	0.629	0.254	80
1GB1	13.014	1.495	0.942	80

Because of the principles behind SMD, we had no doubt that SMD would be able to generate the folded conformation, especially with higher forces. However, since this method is typically followed by the umbrella sampling (Kästner, 2011) procedure which typically takes much longer, it would be more correct to compare the total time of the

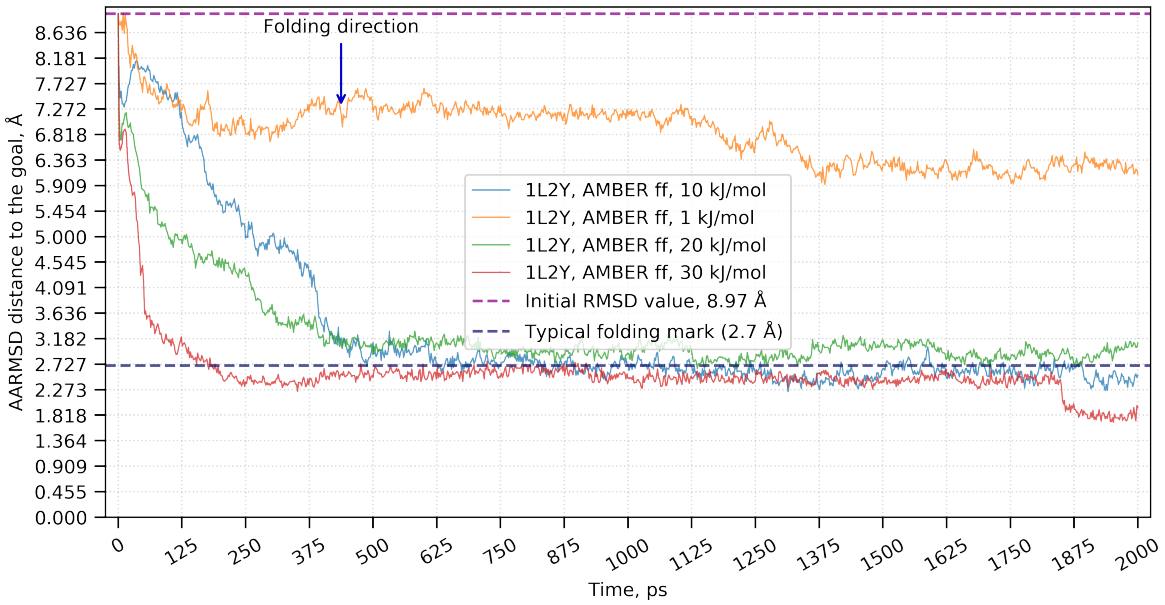


Figure 41: RMSD values during the SMD folding of the 1L2Y with lower force values.

GPA* and SMD with successive umbrella sampling which we plan to perform in future work.

FRODAN results and comparison with GPA*

Table 13: Frodan RMSD distance to the NMR structure for 1L2Y, 1YRF, and 1GB1 proteins

	Initial AARMSD, Å	AARMSD, Å	BBRMSD, Å
1L2Y	8.970	0.770	0.611
1YRF	10.794	0.898	0.454
1GB1	13.014	2.559	1.660

All three runs were able to achieve very low RMSD values, but not all formed the secondary structure. With a simple protein like 1L2Y the final conformation looked complete, however, with increase of complexity of the protein, the algorithm gave less and less correct secondary structure. Furthermore, the folding trajectory was very different from all the trajectories we saw with GPA*.

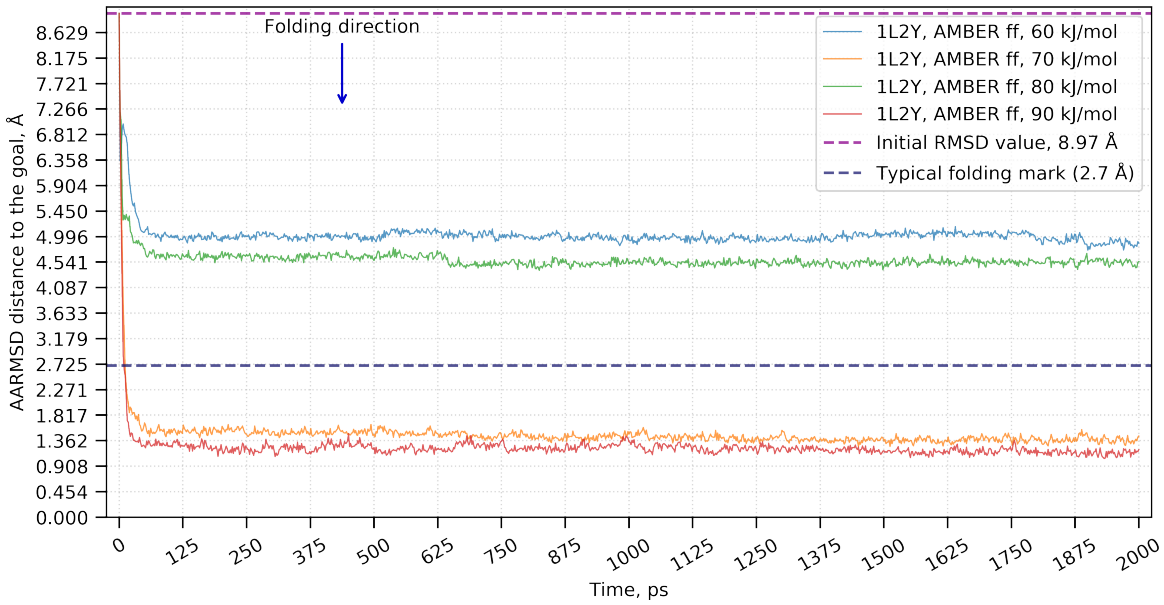


Figure 42: RMSD values during the SMD folding of the 1L2Y with higher force values.

Figure 43 and 44 show that every iteration resulted in RMSD decrease.

While all tested proteins reached very low AARMSD and BBRMSD values, during the visual inspections we found that some secondary structures were not formed:

1L2Y folding process finished with conformation which was had only 3_{10} helix formed.

1YRF folding process finished with assembly one full alpha helix, of almost full second alpha helix, and no third alpha helix.

1GB1 folding process finished with assembly almost full alpha-helix, and two out of four beta sheets were formed partially.

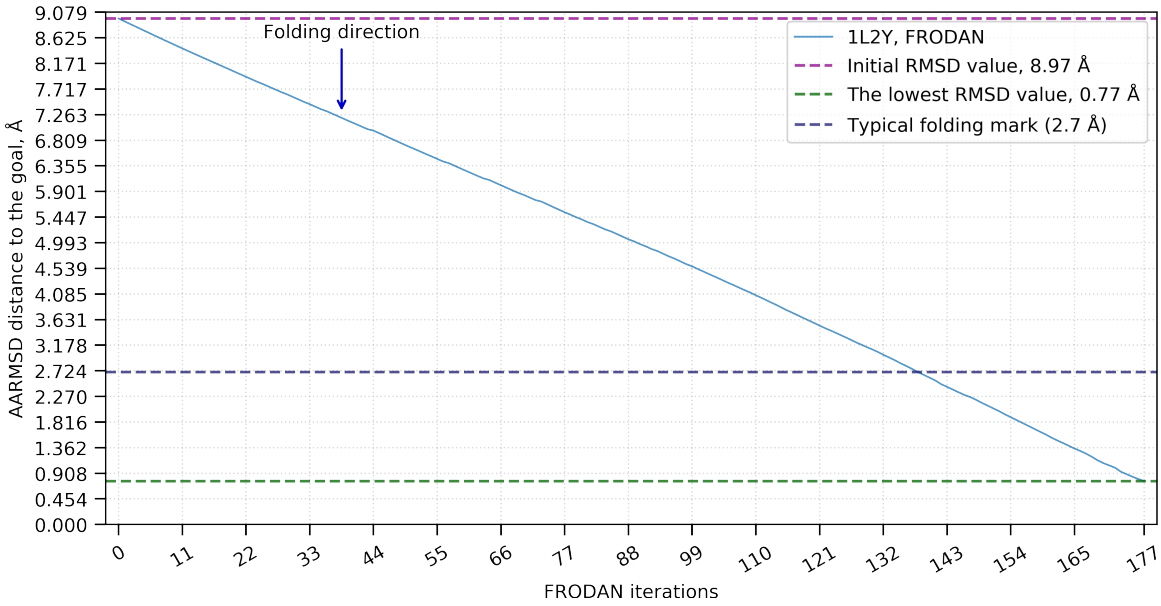


Figure 43: AARMSD values during the FRODAN folding of the 1L2Y.

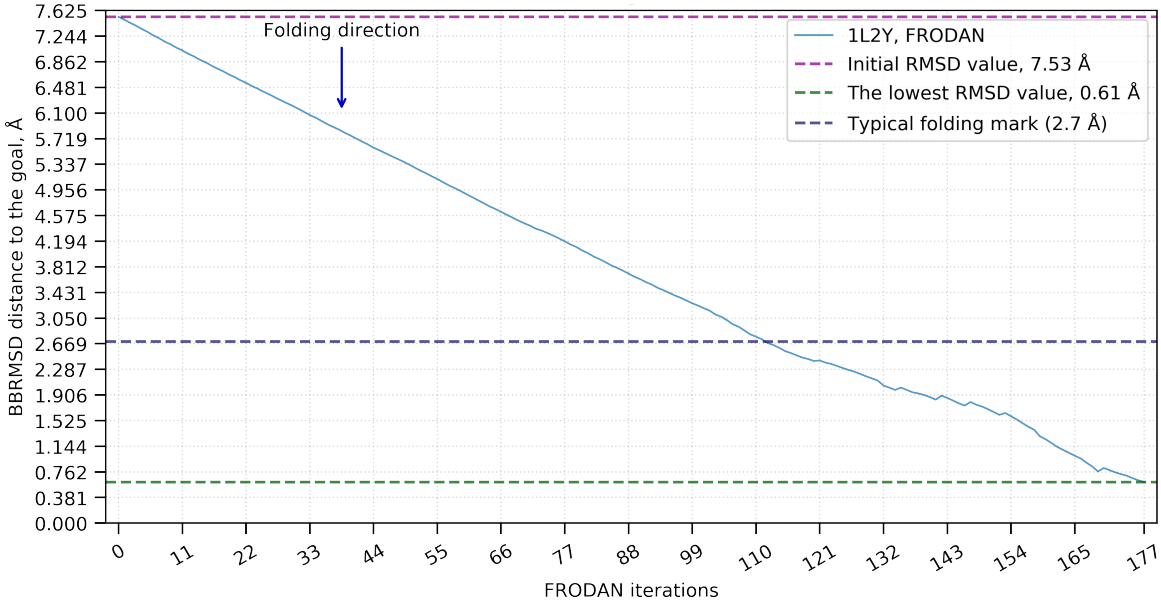


Figure 44: BBRMSD values during the FRODAN folding of the 1L2Y.

DISCUSSION

We proposed a novel approach to perform efficient and more natural Molecular dynamics (MD) simulations and demonstrated the folding process on several proteins with different properties. The results demonstrated that Greedy-proximal A* (GPA*) was able to perform faster folding than the Replica-Exchange Molecular Dynamics (REMD) approach. But, what is more important is that such a result was achieved without adding any artificial energy bias. Additionally, our algorithm was not only able to generate much more compact trajectories with much closer Root-mean-squared deviation (RMSD) distance to the nuclear magnetic resonance (NMR) structure, but also spent less computational time. Because of the low computational overhead, our algorithm can be efficiently scaled to make use of very large cluster systems by computing a large number of independent steps in parallel thus folding proteins in much shorter time compared to REMD or multi-replicate MD.

Our algorithm generates a trajectory that consists only of steps needed to perform the transitions between the folded and unfolded states. Such a trajectory would allow biochemists to study bottlenecks in the folding process which is crucial for drug design which affect protein folding. Furthermore, it is very easy to generate many trajectories that lead to the folded state after one simulation and apply statistical approaches to find the folding patterns. Future experiments with umbrella sampling are planned to identify any potential benefits for calculating ΔG as well.

Code is posted on `github.com/fio2003/GPA_star` under the MIT license and allows anyone to use, study, or improve the code quickly and freely. Postprocessing scripts shipped with the main code allow processing of the data and plotting thousands of graphs which show specific folding patterns or compare distance metric values to various potential energy values to see how particular metrics work with studied proteins or force fields.

Our algorithm was able to produce trajectories which were 47-6799 times faster than previously reported. Additionally, we did not have to tune any of the parameters such as duration of the single simulation, best metric usage only, number of seeds, maximum duration of one metric, etc. With continued tuning we expect the approach to achieve even better folding times.

Understanding how the motions of protein transform one structure to another can help us understand how to engineer or improve the protein to do better in biofuels or to understand drug design.

drug design: if we have proteins that are involved in the signal processing, which can prohibit or enhance the signal, examining the motion of the protein in presence of the drug and without it can give us an idea of how efficient the drug is. Our algorithm allows to perform such a study efficiently by understanding the energy barriers that may appear or disappear after adding or removing the drug. For example, we can run GPA* protein with the drug and without the drug, and compare two pathways. Since our algorithm stores all unsuccessful steps, we can track the probability of transitions between the states. When we combine a protein with the drug, lower (higher) probability will indicate higher (lower) energy barrier. Furthermore, we can inspect the folding pathway for presence of the new energy barriers.

CHAPTER III.

CLUSTERING

INTRODUCTION

Another common application of Molecular dynamics (MD) simulations is the future experimental validation and energy landscape exploration for studying metastable conformations and the transitions between them (Phillips, 2012; Bowman and Pande, 2010). While the problem of capturing metastable states may often be successfully resolved within the timescale of the simulation, finding those states is often performed with automated techniques such as clustering (Bhowmik and Ramanathan, 2018; Sittel and Stock, 2016). Although there are many clustering algorithms available, not all of them can be successfully applied to high-dimensional data such as MD simulations (Steinbach et al., 2004). In particular, recent work from the clustering literature (Sakuraba et al., 2010) shows that many high-dimensional data sets explore a mixture of independent subspaces and previous clustering studies of MD data have ignored such effects. In this study we explore the application of the subspace clustering techniques to MD simulation data and compare the performance with traditional Spectral clustering (SPC) algorithms (Ng et al., 2002) and demonstrate when and why such approaches may be superior to traditional techniques.

BACKGROUND

Trajectory Clustering

A Molecular dynamics (MD) trajectory can be viewed as a set of frames where each frame represents a molecular conformation at a particular time. Such conformation is a set of positions of atoms that form protein which can be viewed as multidimensional space with $3N-6$ degrees of freedom (DOF) - x, y, z for each particle minus 3 for translation and 3 for rotation of the protein as a whole. In order to simplify the notation we will simplify notation by treating each atom as a point in the space. Then, the problem of clustering can be formed as follows: we want to divide a given set of points $X = \{\boldsymbol{x}_j \in \mathbb{R}^d\}_{j=1}^N$ into n groups in such a way that each group contains a subset of points that share similar qualities unique to each particular group. Various approaches to solving this problem have been developed and applied to molecular dynamics simulations:

1. Best and Hege (Best and Hege, 2002) used intramolecular distances to form a similarity graph and for two-part partitioning.

2. Karpen et al. (Karpen et al., 1993) used dihedral angles of the backbone and side-chain groups to cluster MD data of a tri-ribonucleotide and create the "conformational states" and transitions between them using a neural network.

3. Huang et al. (Huang et al., 2017) used perron-cluster analysis to study the decahedron to icosahedron transition in Pt nanoparticles

4. Phillips et al. (Phillips et al., 2011) applied Spectral clustering to intrinsically disordered FG-nucleoporins.

5. Rauscher et al. (Rauscher and Pomès, 2010) followed up with a detailed study of elastin-like disordered proteins.

Even though clustering has been a common analysis technique for the postprocessing folding trajectories (Rajan et al., 2010), to our knowledge, recent clustering algorithms

such as subspace clustering (Elhamifar and Vidal, 2012) have not been applied to molecular dynamics simulation data. Subspace methods assume that a mixture of different processes may contribute to an overall data set, and multi-replicate simulations common in the field may exhibit such properties.

Spectral Clustering

Concept

Spectral clustering is an algorithm commonly used to cluster data points generated by non-linear processes (Von Luxburg, 2007). In this approach, X is represented as a similarity graph G and is partitioned in such a way that points within one group share a high weight (connectivity), while points in different groups share a very low weight. To achieve the goal mentioned above, we let graph $G = (V, E, W)$ be an undirected graph with a set of vertices $V = \{v_1, v_2, \ldots, v_N\}$, set of edges $E = \{e_1, e_2, \ldots, e_{N^2}\}$, and weights $W = \{w_{ij}\}_{N \times N}$, $w \geq 0$ shared by every two vertices. A value of $w_{ij} = 0$ means that there is no connection between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ vectors. Each cluster group mentioned above can be described as sets of points A in V that preserve these properties: $A_i \cup A_j = \oslash$, where $i \neq j$ and $A_1 \cup A_2 \cup \ldots A_n = V$.

In order to clarify the property of similarity within the spectral approach, a brief review of several of the most popular methods for constructing similarity graphs is given below:

1. *Epsilon neighborhood* (Von Luxburg, 2007) - points are treated as similar only if their pairwise distance is smaller than a cutoff parameter, epsilon, thus creating an unweighted graph.

2. *k-nearest neighbors (KNN) graph* (Von Luxburg, 2007) - for each vertex, v_i, k vertices, v_j, are selected with the highest similarity (weight) only if v_i is also among the KNN of v_j, resulting in a mutual KNN graph.

3. *Fully connected graph* (Von Luxburg, 2007) - all vertices in V are connected with a similarity function that encodes all connections.

The Gaussian similarity function (GSF) is one of the most commonly used functions for further refinement of the neighborhood graph (Arya et al., 1998) and is defined as $s(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\sigma^2)$, where σ is a user-defined parameter which determines the rate of decrease in similarity for all points. Selecting an appropriate value for σ can be computationally expensive and time-consuming (Vladymyrov and Carreira-perpinan, 2013). A more advanced method for choosing σ, entropic affinities, that overcomes the restriction of picking a fixed σ in the GSF has been developed (Vladymyrov and Carreira-perpinan, 2013; Hinton and Roweis, 2003).

Construction of Graph Laplacians

Once the GSF has been applied to the neighborhood graph, the graph is transformed into the Laplacian form (Weisstein, 2014). There exist several ways to define the graph Laplacians. One of the most common forms used for clustering is the normalized symmetric graph (Von Luxburg, 2007), which is defined as:

$$L = D^{-1/2} \times W^* \times D^{-1/2}, \tag{1}$$

where W^* is the similarity matrix formed from the neighborhood graph, with elements defined as $w_{ij} = s(\mathbf{x_i}, \mathbf{x_j})$ and D is a diagonal matrix of size N, where $d_i = \sum_j w_{ij}$. Properties and proofs concerning this Laplacian can be found in (Von Luxburg, 2007).

Approximate Normalized Cut

The next step after construction the graph Laplacian is an application of singular value decomposition (SVD) (Golub and Reinsch, 1971) to the graph Laplacian in order to perform an approximate k-way normalized cut (Shi and Malik, 2000). We save k rows from $V' \in \mathbb{R}^{n \times k}$ (unitary matrix that contains right singular vectors as rows) that correspond to the top k eigenvectors and then construct the matrix $Y' \in \mathbb{R}^{n \times k}$ from V' by normalizing the row sums to have norm 1. The matrix Y' represents a nonlinear projection of the data within which the clustering problem may be solved using linear clustering algorithms like k-means (Arthur and Vassilvitskii, 2007). The complete procedure for spectral clustering is shown in Algorithm 3.

Algorithm 3 Spectral clustering algorithm

1: **procedure** SPECTRAL CLUSTERING(S, k) ▷ Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct
2: Construct a similarity graph by one of the ways described above (page 91).
3: Compute the normalized Laplacians L using equation 1.
4: Compute the first k eigenvectors $\mathbf{v_1}, \ldots, \mathbf{v_k}$ of L.
5: Let $V' \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $v_1, \ldots, v_k$ as columns. Construct matrix $Y' \in \mathbb{R}^{n \times k}$ from V' by normalizing the row sums to have norm 1, that is $y_{ij} = v_{ij} / (\sum_k v_{ik}^2)^{1/2}$.
6: Cluster the points ($\mathbf{u_i}$) into clusters $F_1, \ldots, F_k$ with k-means algorithm.
7: **return** Clusters $A_1, \ldots, A_k$ with $A_j = \{j | \boldsymbol{y}_i \in F_j\}$.
8: **end procedure**

Data Subspaces

Clustering of the points in X may be challenging since the points may be positioned among a set of m (affine) subspaces $\{S_\ell\}_{\ell=1}^{m}$ in d dimensions.

Figure 45 demonstrates an intersection of two sets of points (purple and green) that reside in different subspaces, often assumed to have been generated by two distinct latent (unknown) processes. Clustering data from different subspaces (purple and green) can

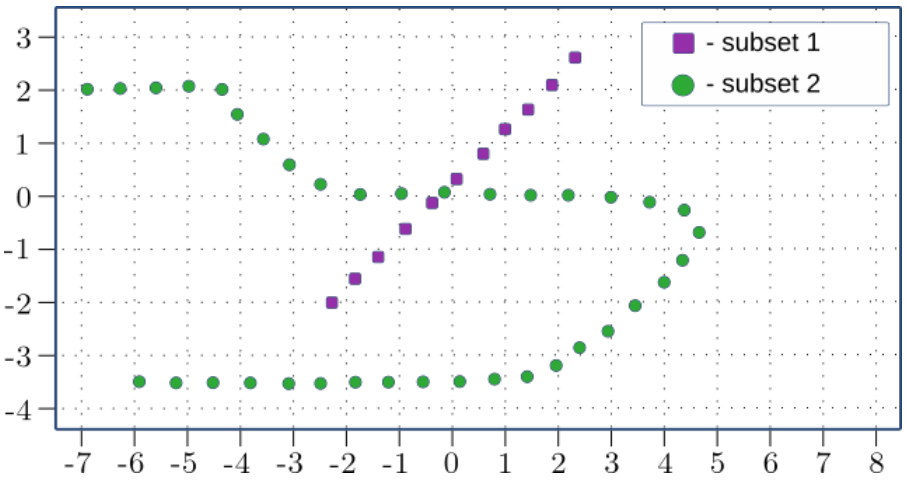


Figure 45: Example of data that resides in two subspaces.

prove challenging for spectral methods to separate into the desired clusters since the similarity graph will connect points near the region of intersection. Latent processes may also be nonlinear (green) or affine (purple) in nature, but neither the presence of subspaces nor the (non)linearity is usually known *a priori* for general data sets. For example, let's say that X resides in d dimensions with m affine subspaces $\{S_\ell\}_{\ell=1}^{m}$. We may often need to cluster X according to the subspaces $\{S_\ell\}_{\ell=1}^{m}$ to obtain the desired results. In the case of $d = 1$, the problem reduces to the solution of the well-known and easily solved principal components analysis (PCA) (Bryant and Yarnold, 1995). However, the problem described above becomes significantly more difficult with the growth of both m and d (Elhamifar and Vidal, 2012). Perhaps more importantly, many problems invoke non-affine (nonlinear) subspaces which may limit the applicability of certain clustering algorithms. In particular, although spectral clustering may solve nonlinear problems, it unfortunately cannot handle clustering within multiple subspaces (Vidal, 2010). For example, in Figure 45 the intersection at (0,0) will not be separated and, most likely, will merge the two data sets in such a way that the normalized cut will not separate the two processes as desired.

Entropic Affinities

As was mentioned in "Concept", selecting sigma (σ) for the GSF may be challenging and time consuming (Von Luxburg, 2007). This is especially true when dealing with large data sets of nonuniform density (Vladymyrov and Carreira-perpinan, 2013). When applied to data exhibiting both dense and sparse regions, the chosen sigma may keep too many points in dense regions and too few in sparse regions. Entropic affinities promise to overcome such an inconvenience by selecting sigma values for each point with respect to a desired perplexity by implicitly defining a continuously differentiable function in the bounded input space (Vladymyrov and Carreira-perpinan, 2013; Hinton and Roweis, 2003).

For an a posterior distribution of an isotropic kernel density estimator of width σ_i defined on X, let's define a discrete distribution $p_j(\boldsymbol{x}_i; \sigma_i)$ with probabilities for $i, j = 1, \ldots, N$ (Vladymyrov and Carreira-perpinan, 2013):

$$p_j(x_i; \sigma_i) = \frac{exp\left(-\left\|\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\sigma_i}\right\|^2\right)}{\sum_{k=1, k!=i}^{N} exp\left(-\left\|\frac{\boldsymbol{x}_i - \boldsymbol{x}_k}{\sigma_i}\right\|^2\right)} \tag{2}$$

In this case each σ_i is being set individually for each point, x_i, to a value such that the entropy of the distribution, $p_j(x_i; \sigma_i)$, equals $log(K)$, where K is a user-set parameter called "perplexity" (Vladymyrov and Carreira-perpinan, 2013). Given both the theoretical and practical advantages offered by entropic affinities, a comparison of their effectiveness for MD simulation data clustering with the fixed (average) σ approach is warranted.

Subspace Clustering

Another approach for clustering, which assumes the data lie in multiple (affine) subspaces was suggested by Elhamifar and Vidal (Elhamifar and Vidal, 2012). It uses

the following data property: any point in a union of subspaces can be represented as a linear combination of several points in the local neighborhood. The coefficients of this combination can be used to construct an affinity matrix due to the local relationships between points residing in the same affine subspace. Thus, X can be viewed as a self-expressive dictionary in which each point $\boldsymbol{x}_i \in \cup_{\ell=1}^{m} \mathcal{S}_\ell$, where $\mathcal{S}_\ell$ - subspace, $\ell = 1$ - relaxation type for efficient solution of the sparse optimization problem, can be written as a linear combination of other points

$$\boldsymbol{x}_i = X\boldsymbol{c}_i, \tag{3}$$

where $c_i \stackrel{\Delta}{=} [c_{i1}, c_{i2}, \ldots, c_{iN}]^\top$. In order to remove the trivial solution of describing a point as a linear combination of itself, an additional constraint $c_{ii} = 0$ is added.

In practice, the number of data points in a subspace S_ℓ is often higher than its dimension, which suggests that the representation of $\boldsymbol{x}_i$ in the dictionary X is not unique in general. This assumption leads to the conclusion that each $\boldsymbol{x}_i$, and consequently X, has a non-trivial null-space, giving rise to infinitely many representations of each data point.

The key observation in the proposed algorithm was that among all solutions of Equation 3, there exists a sparse solution, $\boldsymbol{c}_i$, whose nonzero entries correspond to data points from the same subspace as $\boldsymbol{x}_i$. Such a solution would be referred as a subspace-sparse representation (Elhamifar and Vidal, 2012).

A data point $\boldsymbol{x}_i$ that lies in the d_ℓ dimensional subspace S_ℓ can be described as a linear combination of d_ℓ other points from $\mathcal{S}_\ell$. A sparse representation of a data point gives the opportunity to find points from the same subspace where the number of non-zero elements relates to the dimension of the underlying subspace. A system of equations similar to Equation 3 may contain an infinite number of solutions, which can be restricted by minimization of an appropriate objective function. An example of such a restriction

with the ℓ_q-norm of the solution is shown below in Equation 4:

$$min\|\boldsymbol{c}_i\|_q \text{ s.t. } \boldsymbol{x}_i = X\boldsymbol{c}_i,\ c_{ii} = 0 \tag{4}$$

Different choices of q give rise to different solutions (Elhamifar and Vidal, 2012). Usually, by decreasing the value of q from infinity toward zero, the sparsity of the solution increases (Elhamifar and Vidal, 2012). The extreme case of $q = 0$ corresponds to the general NP-hard problem (Elhamifar and Vidal, 2012; Knuth, 1974) (at least as hard as problems which can be approximated within every constant in polynomial time (Amaldi and Kann, 1998)) of finding the sparsest representation of the given point and is not being considered since we are interested in the efficient way to find a nontrivial sparse representation of $\boldsymbol{x}_i$ in the dictionary X. Minimization of the tightest convex relaxation of the ℓ_1-norm (q=1) is considered as sufficient, which can be solved efficiently using convex programming tools (Boyd and Vandenberghe, 2004). Equation 4 can be rewritten in matrix form for all data points $i = 1, \ldots, N$ as :

$$min\|\boldsymbol{C}\|_1 \text{ such that } X = XC,\ \ diag(C) = 0, \tag{5}$$

where $C \triangleq [c_1 c_2 \ldots c_N] \in \mathbb{R}^{N\times N}$ is the matrix whose ith column corresponds to the sparse representation of $\boldsymbol{x}_i, \boldsymbol{c}_i$, and $diag(\boldsymbol{C}) \in \mathbb{R}^N$ is the vector of the diagonal elements of C. Ideally, the solution of Equation 5 corresponds to a sparse subspace representations of the data points, which is used next to derive the clustering of the data.

Algorithm 4 Sparse Subspace Clustering

1: **procedure** SUBSPACE CLUSTERING(S, k) ▷ A set of points $\{\boldsymbol{x}_i\}_{i=1}^{N}$ lying in a union of m linear subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^{m}$.
2: Solve the sparse optimization program.
3: Normalize the columns of $\boldsymbol{C}$ as $c_i \leftarrow \frac{c_i}{\|c_i\|_\infty}$.
4: Compute the first k eigenvectors $\mathbf{v_1}, \ldots, \mathbf{v_k}$ of L.
5: Form a similarity graph with N nodes representing the data points. Set the weights on the edges between the nodes by $\boldsymbol{W'} = |\boldsymbol{C}| + |\boldsymbol{C}|^\top$.
6: Apply spectral clustering described in Algorithm 3 to the similarity graph $\boldsymbol{W'}$.
7: **return** SpectralClustering($\boldsymbol{W'}, k$). ▷ see Algorithm 3
8: **end procedure**

To perform clustering, first a weighted graph is constructed (Elhamifar and Vidal, 2012) $\mathcal{G} = (V, E, \boldsymbol{W'})$, where $\boldsymbol{W'} \in \mathbb{R}^{N \times N}$ is a non-negative symmetric similarity matrix representing the weights of the edges. The similarity matrix $\boldsymbol{W'}$, and thus the similarity graph $\mathcal{G}$, contains nodes that correspond to the points of the same subspace connected to each other, and there are no edges between nodes that correspond to the points in different subspaces. Recall that construction of common graph Laplacians requires a symmetric affinity matrix, while the sparse representation from the convex optimization does not guarantee symmetry. One possible symmetrization is $\boldsymbol{W'} = |\boldsymbol{C}| + |\boldsymbol{C}|^\top$, which can be described analogously: if node i is connected to node j with weight w, then j should have a connection to i with the same weight. The complete procedure for the Subspace Clustering (SSC) is shown in Algorithm 4. A problematic assumption made using this approach is that the data consists of only *affine* subspaces. We addressed this limitation in the Methodology section below.

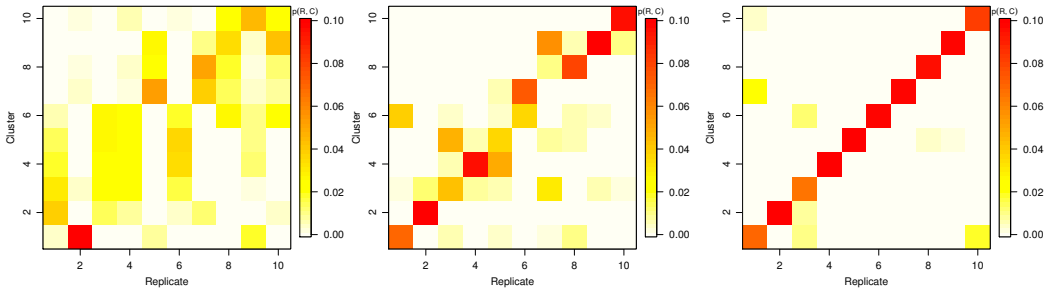
Normalized Mutual Information

Since we need some tool to quantitatively measure each algorithm's ability to separate data into clusters, we utilize the normalized mutual information (NMI) (Estévez et al., 2009). Mutual information reflects the dependence of two variables, in our case - simulation replicate number (R) and cluster number (F):

$$NMI(R;F) = \frac{\sum_{f \in F} \sum_{r \in R} p(r,c) log_2 \left(\frac{p(r,f)}{p_1(r) p_2(f)} \right)}{argmax \left(\sum (p(r) log_2 p(r); \sum (p(c) log_2 p(f)) \right)}, \quad (6)$$

where $p(r,c)$ is the joint probability distribution of the two random variables R and F; $p_1(r)$ and $p_2(f)$ are the marginal probability distributions of R and F respectively.

Examples of different NMI values may be found in Figure 46.



(A) NMI: 0.1575 (B) NMI: 0.4495 (C) NMI: 0.6360

Figure 46: Three examples of cluster-replicate joint probability distributions for low (0.1575, A), medium (0.4495, B), and high (0.6360, C) NMI values.

These plots were built with data that can be found in the Results section, but are used here to illustrate the use of NMI for determining clustering quality for MD simulations. In Figure 46 the plot A demonstrates the joint probability distribution from the Table 18, row 9, column 4, the plot B refers to the Table 16, row 3, column 1, and the plot C refers to the Table 17, row 11, column 3. For the *high* NMI example (plot C) we may easily say which data set (replicate simulation) is referred to by a particular cluster, for the medium NMI example (Figure 46.B) we clearly see the trend but it is not always possible to make an assumption about cluster-replicate relationships like we did for example with high

NMI value. Finally, the example of low NMI (Figure 46.B) demonstrate an example where the simulation data are too similar so that the algorithm cannot distinguish between the simulations. The critical case of an NMI of 1.0 would exhibit a plot with a filled diagonal that reflects an exact mapping between the input data sets: one per cluster. However, for NMI close to 0.0 we would see an even distribution among the clusters, meaning that the particular algorithm sees no difference between the input data sets. NMI therefore is a good summary criterion of the overall effectiveness of sampling for protein ensembles generated using MD.

Kmeans clustering algorithm

Kmeans is a very popular unsupervised clustering algorithm (Lloyd and Stuart, 1982). Here we will provide an overview of it and encourage to read the original Lloyd and Stuart paper (Lloyd and Stuart, 1982) or improved version called kmeans++ proposed by Arthur et al. (Arthur and Vassilvitskii, 2007). In kmeans, k stands for the number of clusters we want to divide a set of points $\boldsymbol{x}$, where x_i is a n-dimensional point. The algorithm tries to divide $\boldsymbol{x}$ into sets $\boldsymbol{A}$, such as the distance (typically the sum of squares or variation) between all points inside A_i is minimal:

$$\underset{s}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in A_i} ||x - \mu_i||^2 = \underset{s}{\operatorname{argmin}} \sum_{i=1}^{k} |A_i| Var A_i \tag{7}$$

, where μ_i is the mean of points in A_i. In the original kmeans algorithm (Lloyd and Stuart, 1982) (Lloyd algorithm), the initial cluster centers are selected randomly from the set $\boldsymbol{x}$. kmeans++ optimizes the initial cluster selection by selecting points that are far from each other, thus reducing the number of iterations needed for convergence.

An illustration of the algorithm can be viewed on Figure 47. Unfortunately kmeans algorithm heavily depends on the initial cluster centers and is unable to the nonlinearity and affinity.

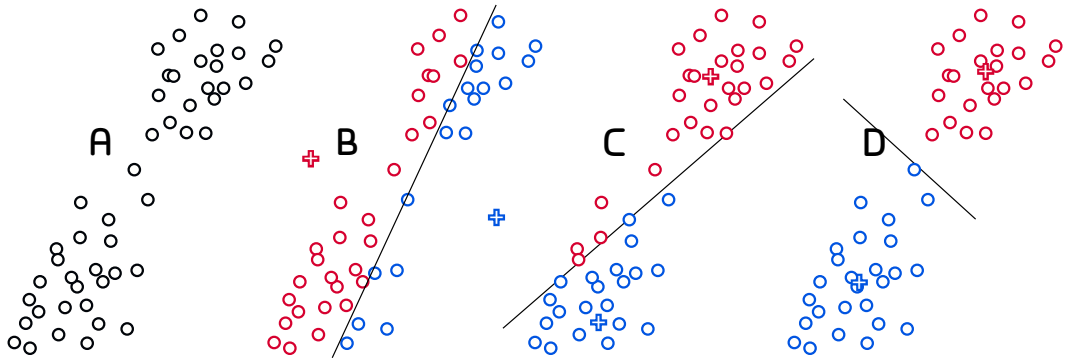


Figure 47: Example of the kmeans clustering algorithm. A - represents unlabeled initial data, B - initial labeling according to the cluster centers (+), C - labeling according to the recomputed cluster centers, D - final clusters.

METHODOLOGY

Clustering methodology

As follows from the descriptions above, that the connecting link to combination of both approaches (Spectral clustering (SPC) and Subspace Clustering (SSC)) can be found at the steps just prior to singular value decomposition (SVD). Among different approaches, we suggest that the dot product (SDS) and element-wise product (SES) multiplications of both the affinity matrices (standard spectral and standard subspace) may produce normalized mutual information (NMI) increase. In other words, only weighted connections between points which exist in *both* the standard similarity graph (nonlinear) *and* the subspace-sparse graph (affine subspace) should be preserved. Therefore, a general, efficient algorithm may be defined as follows:

1. Compute optimization coefficients C.
2. Compute affinity matrix S.
3. Construct matrix $M = SC \cdot C$ (SDS) or $M = SC * C$ (SES)
4. Construct graph Laplacians.
5. Perform singular vector decomposition.
6. Run k-means algorithm.

Geometric Rationale

The standard Gaussian affinity graph will connect points within close proximity to one-another, but exclude those far apart as shown in red (see Figure 48). This property preserves the relationships between points along nonlinear manifolds. The yellow area where the blue and green data sets intersect is however problematic since

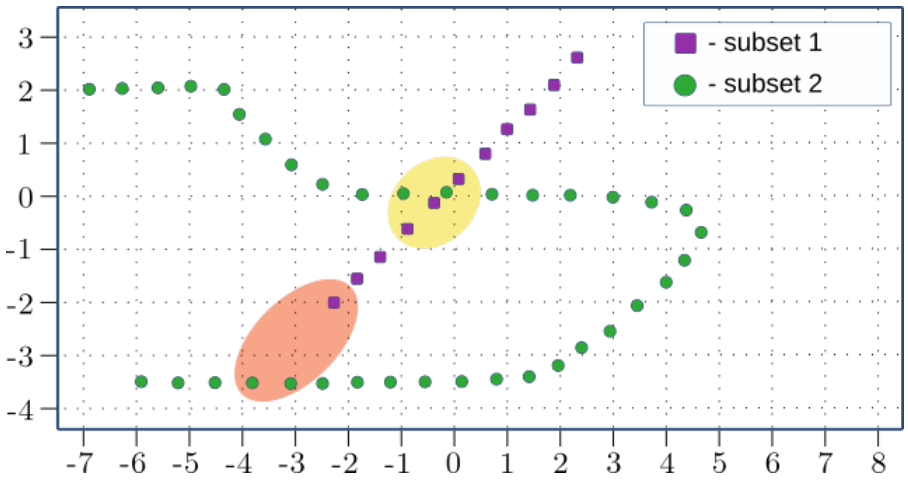


Figure 48: Example of a complex manifold with structures considered challenging for the standard clustering algorithms. Red area shows a region that is difficult for the subspace algorithms to separate. Yellow area indicates a region challenging for the spectral clustering to separate.

the local connectivity blurs the relationship between different manifolds and connects the manifolds together. The subspace method treats all blue points and part of the green points (red area) as one subspace since they are observed to lie in the same affine subspace (along the same line in the ambient space). However, the boundary points in the yellow area are more separated between subspaces than when using the Gaussian affinity function. This geometric interpretation suggests that a combination of the Gaussian affinities and subspace algorithm coefficients may result in a better separation of points in both the red and yellow areas. In both approaches, we express all points as vectors of n weights. When both methods "agree" (both result in the same connections) to connect certain points, such connections result in higher weights in the final connectivity matrix. When they disagree, the connectivity coefficients will be lower and result in breaking unreliable links (spurious connectivity relations created by either the standard spectral or subspace algorithms, independently).

Clustering implementation

We chose Python3 (Van Rossum and Drake, 2011) as the main implementation programming language for the algorithm. For solution of the sparse convex problem we tested different solvers, but only ECOS (Domahidi et al., 2013) was able to work with arrays of requested size and produced correct results consistently. Since parameter tuning requires multiple executions of the algorithm, we developed a strategy of spreading the load among the nodes on a cluster. We selected the client-server model, where the client is the clustering software and the server is a dispatcher that tracks which tasks are executed at the moment. Since our clustering algorithm consists of two separate parts, we can reuse the previous computations to speed up the overall process.

Figure 49 and Algorithm 5 describe the client-server interaction.

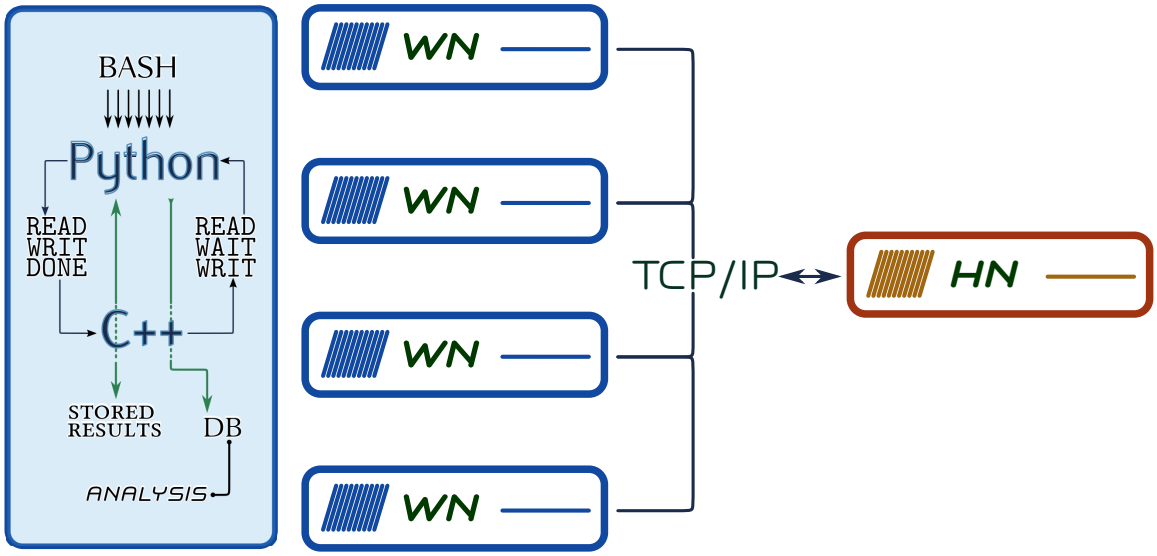


Figure 49: Pyssc architecture. HN - head node, WN - work nodes

Client-server communications are performed through TCP/IP (Forouzan, 2010) communications with the assumption that the client knows the exact IP address of the server. The server is implemented in C++ and uses the system call, *epoll*, to track multiple file descriptions within $O(1)$ time. Example of messages sent between client and server

Algorithm 5 Client-server communications

client A is executed with specific clustering parameters (method, solver, KNN number, etc)
client A generates the digest of input parameters with the MD5 (Rivest, 1992) algorithm
client A searches the file which contains result values obtained during prior runs with the digest as name

if the file with such a name exists **then**
    client A asks the server to read it
else
    client A asks the server to create it
end if
server checks whether there is another client B which is generating such a file:
if the file is being generated by the client B **then**
    client A is advised to not create the file, but wait until signaled and read it
    client B performs the computation, stores the file, and signals the server that the file was generated
else
    client A is advised to perform the computation and store partial results in the file
end if
server informs all other clients who was waiting to read the file, that the file is ready to be read
client A stores results of clustering in the database

are described in Figure 50. We used SQLite (Team, 10) as a database for the results storage and further analysis.

Additional version information about libraries used in the experiment:

1. Numpy (Oliphant, 06): 1.13.1
2. Scipy (Jones et al., 01): 0.19.1
3. sparsesvd (KARDOŠ, 2010): 0.2.2
4. CVXPY (Diamond and Boyd, 2016): 0.4.10 with solver ECOS

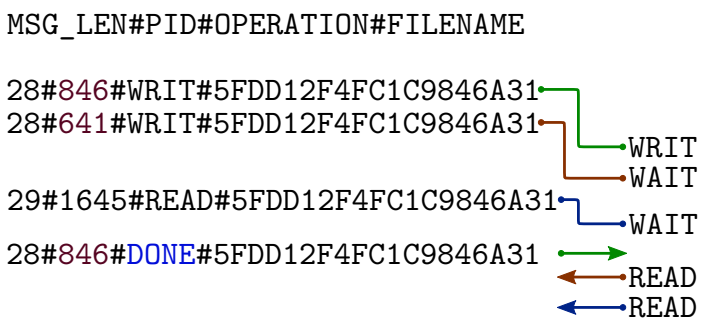


Figure 50: Example of message passing between client and server. # - delimiter between parts of the message. First part is total message length.

5. Pandas (McKinney, 2010): 0.20.3

6. Matplotlib (Hunter, 2007): 2.0.2

Data Preparation

We studied two general types of proteins: natively folded proteins (NFP) and intrinsically disordered proteins (IDP). NFP fold into stable conformers thus utilizing fewer available degrees of freedom and limiting variation in conformation over time. Proteins for the NFP group were picked in such a way that we have representation of two major structural classes: alpha-helical (Trp-Cage Miniprotein Construct TC5b (1L2Y) (Neidigh et al., 2002)) and beta-sheet (immunoglobulin binding domain of streptococcal protein G (1GB1) hairpin (Gronenborn et al., 1991)). 1L2Y and 1GB1 are well-known and widely used in Molecular dynamics (MD) simulations to demonstrate secondary and tertiary structures as well as fast folding. IDP do not tend to converge to some particular conformer, thus their simulations result in a broad variety of trajectories. Another significant difference compared to NFP is that despite both types demonstrating a high dimensional motion at higher temperatures, at physiological temperatures NFP tend to reduce dimensionality proportionally to folding progress, while IDP lack such behavior.

On the IDP side, NSP1 protein, tRNA m1A58 methyltransferase (5EQJ) (Goffeau et al., 1996) was selected as an example of a relaxed-coil structure which exhibits few meta-stable conformations and nucleoporin NUP116p protein (YJM1418) (Goffeau et al., 1996) as an example of a compact collapsed-coil structure with many meta-stable conformations (Yamada et al., 2010).

All simulation trajectories were obtained using GROningen MAchine for Chemical Simulations (GROMACS) 4.5.4 (Pall et al., 2014) with the AMBERff99SB-ILDN force field (AMBER) (Lindorff-Larsen et al., 2010; Case et al., 2018) force field and the TIP3P (MacKerell Jr et al., 1998) water model with 150 mMol $Na^{+}Cl^{-}$ added to neutralize the system. A better force field can be examined in the future, although we believe that this will not affect our results in a general way. We will evaluate overall clustering quality, which does not depend on the particular force field. For each protein we created 10 independent simulations with a duration of 350 ns each, but with different initial velocities. The temperature profile is specified in supplementary materials in section 'Temperature profile', and reflects heating each protein into a highly disordered shape and then monitoring its return to the native stable state.

1. $0 - 20\ ns$: 300 K to 600 K
2. $20 - 80\ ns$: 600 K
3. $80 - 100\ ns$: 600 K to 300 K
4. $100 - 350\ ns$: 300 K

which reflects heating each protein into a highly disordered shape and then monitoring its return to the native stable state. The integration time step was chosen as 2 fs. The first 100 ns (steps 1-3) were discarded as the equilibration phase (a special phase which allows to distribute the kinetic energy, introduced during initial heating, among all degrees of freedom (DOF)). After simulation we extracted the backbone structure from each

simulation frame. Prior to clustering, the set of coordinate frames was translated into the dihedral angle space and sin-cos embedding of the dihedral angles with the Molecular Dynamics Spectral Clustering Toolkit (MDSCTK) (Phillips et al., 2008).

We created three data sets in order to test how the data density would affect the final result: Dense (DN) - frames were taken every 10 ps resulting in 2501 points per simulation, then 10 simulations were concatenated to form a complete data set with the size of 25010 points, Sparse (SP) - frames were taken every 100 ps resulting in 251 points per simulation, then 10 simulations were concatenated to form a complete data set with size of 2510 points, and Super-sparse (SS) - frames were taken every 1000 ps resulting in 25 points per simulation, then 10 simulations were concatenated to form a complete data set with the size of 250 points.

Clustering Setup

k-nearest neighbors (KNN) were precomputed with MDSCTK and stored for future use. For experiments with plain Gaussian similarity function (GSF), sigma values were hand selected to achieve the top performance among different proteins, but all results were saved for future analysis. The sigma/perplexity selection strategy used was as follows: the range of values was 'scanned' for the best NMI values and then selected for a finer 'neighborhood' search. Note that not all sigma/perplexity values may be present in all experiments since for each protein different sigma/perplexity values resulted in high NMI. We applied a similar strategy for selecting KNN.

For the final clustering we ran the k-means++ (Arthur and Vassilvitskii, 2007) algorithm 80 times (due to its stochastic nature, discussed in subsection "Kmeans clustering algorithm"). Our experience suggests that this was more than enough since the maximum deviation was only 0.042 NMI and the average deviation was only 0.005 NMI. Other implementations of the k-means algorithm may require different number of executions to improve the stability or decrease overall computation time. NMI was computed and

stored after every iteration in order to derive the maximum, minimum, average, and median values for the particular test set. We used only the median data for the analysis in order to provide a fair analysis which does not depend on spikes of very high NMI which observed during the review of results of the hybrid approach. Additional results may be found online at: `https://github.com/fio2003/PYSSC/tree/master/pyssc_usage_and_raw_results/results_database/results.7z`. All code implementation along with more detailed results can be found at: `https://github.com/fio2003/PYSSC`. Our parallel scheduler which we used for running the analysis can be found at `https://github.com/fio2003/PYSSC_scheduler`.

The bare k-means clustering test was not included since this algorithm does not support any kind of nonlinearity in data.

Statistical Analysis

Analysis of the clustering results for the above experiments was performed as follows.

Overall performance analysis: We selected the highest NMI values among each group of algorithms, protein types, affinity types, and data densities (categories) and created the three tables (16, 17, and 18) which represent how the clustering quality depends on algorithms and proteins.

Graph segmentation analysis: We plotted the relationship between NMI values and perplexity, sigma and KNN for each category to analyze the unique properties. Each graph was divided into three segments. Each segment was later classified according nomenclature given in Tables 14 and 15. The two examples of such a classification as shown in Figures 51 and 52.

Segmented graph analysis: We used the previous classification to plot the relationship between the physical width of graphs described in the previous paragraph and NMI

values. Each graph was divided into 3×3 sectors: three for the NMI classification and three for the thickness classification. For each sector we counted the number of segments that fall into the sectors to show the relationship between the NMI values and variance for each category.

Boxplox analysis: Finally, we used violin plots and boxplots to better describe the distribution of NMI values inside each combination of categories.

All percentages shown are calculated as follows: the sum of elements classified as W, M, N is equal 100% and the sum of elements classified as "/", "-", "\" is also equal 100%. All others (CT, CS, S, defined in Tables 14 and 15) show percent of the maximum possible value.

Table 14: Graph width nomenclature used for analysis.

	Graph thickness
W	width more than 0.1 NMI.
M	width between 0.05 - 0.1 NMI.
N	width less than 0.05 NMI.
CT	flags that two adjacent segments were classified differently.

Table 15: Graph shape nomenclature used for analysis

	Graph shape behavior
/	grows more than 0.05 NMI per segment.
-	does not grow/fall more than 0.05 NMI.
\	falls/decreases more than 0.05 per segment.
S	significantly - more than 0.1 NMI per segment.
CS	changes live /\or \/, also called A and V shapes.

Additionally we will explain the meaning of the parameters defined in Tables 14 and 15.

As for the graph width - it is desirable to see more narrow parts of the graph since those indicate that the behavior depends mostly on the studied parameter. Wide graph

indicates that studied parameter may have a small impact on the resulting NMI, and lack consistency of the result. CT indicates that some other parameter's (which is not reviewed at the moment) impact depends on the studied parameter's value, which indicates that compared parameters have nonlinear relationship.

A growing or a falling behavior of the trend line can be treated as suggestion in parameter's values needed to obtain the best possible clustering results. S indicates that the studied parameter has high impact on the final result. CS is a saddle point indication which is either the best or the worst studied parameter's value. NMI value qualitative assessment:

high NMI : indicates that algorithm is able correctly classify the data, however, it can also indicate that the data was not uniform, so it was easy divide it into clusters.

low NMI : indicates that algorithm was not able to correctly classify the data or the data points formed close to the uniform distribution.

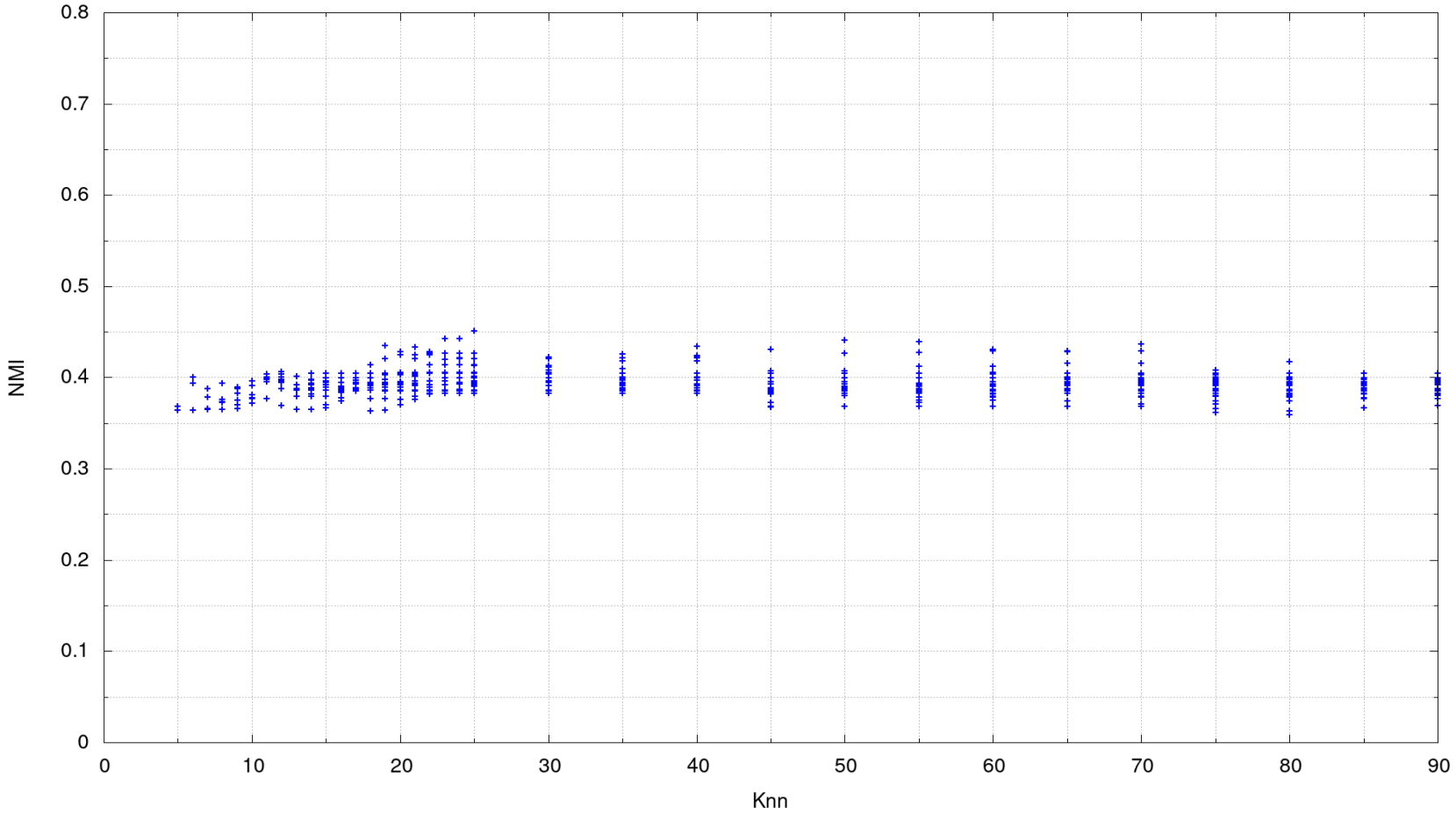


Figure 51: Example of a medium thickness, straight graph derived from the NMI/KNN results for the SPC algorithm with entropic affinities for super sparse data of the 1L2Y protein.

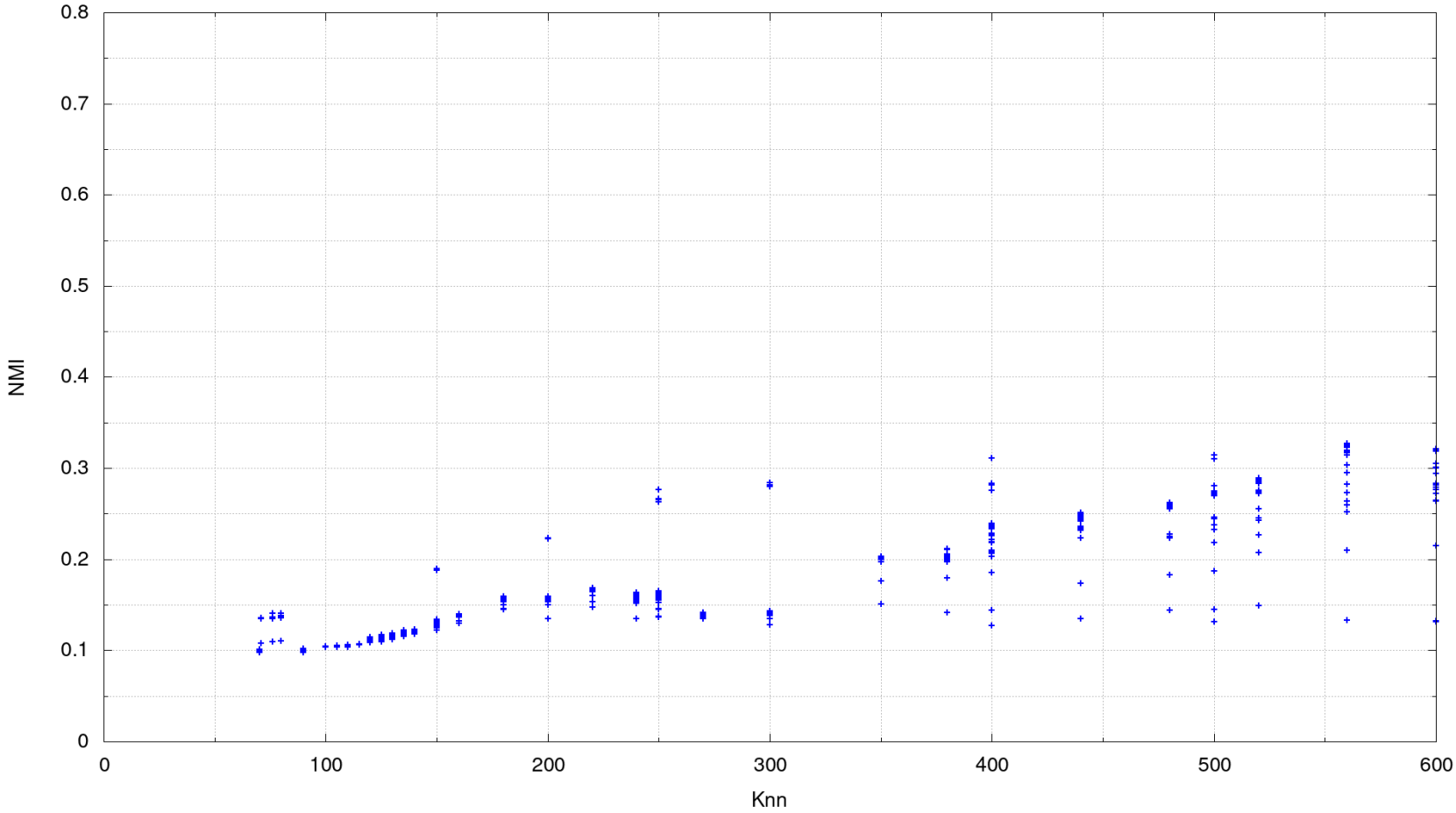


Figure 52: Example of a thickness that changes from medium to wide and has a growing trend; derived from the NMI/KNN results for the SES algorithm with the plain affinity for sparse data of the 5EQJ protein.

RESULTS

Clustering

All tables below contain the median normalized mutual information (NMI) results for the corresponding experiments. A brief description of meaning of the NMI values can be found in section "Statistical Analysis".

Overall Performance (Detailed)

For the dense data we used only the entropic affinities due to the prohibitive computational cost of exploring the parameter space for fixed sigma.

The Entropic Affinities Analysis for dense data (Table 16)

Discussion below is in reference to Table 16. Algorithms: For all cases the Spectral

Table 16: The best NMI values for each protein obtained with all algorithms using entropic affinities and the dense data set.

NFP		IDP		
1L2Y	1GB1	YJM1418	5EQJ	
0.4495	0.5053	0.7447	0.5114	SPC
0.2848	0.2772	0.4157	0.3128	SSC
0.3892	0.3991	0.5869	0.4148	SDS
0.3752	0.4695	0.5794	0.5052	SES

clustering (SPC) algorithm showed the highest NMI values while the Subspace Clustering (SSC) algorithm showed the lowest results.

Proteins: natively folded proteins (NFP) (0.4495 for Trp-Cage Miniprotein Construct TC5b (1L2Y) and 0.5053 for immunoglobulin binding domain of streptococcal protein G (1GB1)) and NSP1 protein, tRNA m1A58 methyltransferase (5EQJ) (0.5114) demonstrated similar results while nucleoporin NUP116p protein (YJM1418) demonstrated a significantly higher NMI value (0.7447).

The Entropic Affinities Analysis for sparse data (Table 17)

Discussion below is in reference to Table 17. Algorithms: element-wise product

Table 17: The best NMI values for each protein obtained with all algorithms for the sparse data set.

Entropic affinity

NFP		IDP		
1L2Y	1GB1	YJM1418	5EQJ	
0.4593	0.5048	0.7311	0.5545	SPC
0.4740	0.3665	0.6345	0.5182	SSC
0.4924	0.4662	0.7169	0.5432	SDS
0.5018	0.4901	0.7214	0.5890	SES

Plain affinity

NFP		IDP		
1L2Y	1GB1	YJM1418	5EQJ	
0.3100	0.2485	0.2864	0.3037	SPC
0.4740	0.3665	0.6345	0.5182	SSC
0.4881	0.4319	0.6360	0.5350	SDS
0.2685	0.3047	0.3395	0.3269	SES

(SES) demonstrated high NMI values for all proteins, but SPC was slightly better for 1GB1 and YJM1418. SSC performed the worst among algorithms for intrinsically disordered proteins (IDP) and 1GB1.

Proteins: 1L2Y and 1GB1 had almost identical NMI values - 0.518 for 1L2Y and 0.5048 for 1GB1. 5EQJ had slightly higher NMI - 0.589 than both NFP. YJM1418 had the highest NMI among all proteins - 0.7311.

Plain Affinities Analysis for sparse data (Table 17)

Discussion below is in reference to Table 17 on page 115. Algorithms: dot product (SDS) showed the highest NMI among algorithms for all proteins while SPC had the lowest NMI values for 1GB1 and IDP. SES showed the lowest NMI value for 1L2Y. Proteins: NFP had lower NMI values, while IDP had higher NMI values.

Entropic Affinities Analysis for super-sparse data (Table 18)

Discussion below is in reference to Table 18 on page 116.

Table 18: Best NMI values for each protein obtained with all algorithms for the super-sparse data set.

Entropic affinity

NFP		IDP		
1L2Y	1GB1	YJM1418	5EQJ	
0.4508	0.4592	0.5941	0.4797	SPC
0.4150	0.3861	0.6586	0.4685	SSC
0.4170	0.4017	0.6727	0.4624	SDS
0.4224	0.4095	0.6509	0.5128	SES

Plain affinity

NFP		IDP		
1L2Y	1GB1	YJM1418	5EQJ	
0.3496	0.3181	0.2989	0.1575	SPC
0.4150	0.3861	0.6586	0.4685	SSC
0.4273	0.4159	0.6759	0.4685	SDS
0.3487	0.3490	0.3210	0.3016	SES

Algorithms: SPC performed the best for NFP group with NMI values of 0.4508 and 0.4592 for 1L2Y and 1GB1 respectively, but demonstrated the worst NMI value of 0.5941 for YJM1418. SSC demonstrated the lowest NMI values for the NFP group resulting in 0.4150 and 0.3861 for 1L2Y and 1GB1 respectively. SDS demonstrated the highest NMI value for YJM1418 - 0.6727, but the lowest NMI value for 5EQJ. SES demonstrated the highest value for 5EQJ - 0.5128. Proteins: Both NFP showed similar values and within the IDP group, YJM1418 had the highest NMI value - 0.6727.

Plain Affinities Analysis for super-sparse data (Table 18)

Discussion below is in reference to Table 18. Algorithms: SPC demonstrated the worst NMI results for the IDP and 1GB1. SDS demonstrated the highest NMI values for all proteins. Subspace demonstrated the same (highest) NMI value for 5EQJ. SES demonstrated the worst NMI values for 1L2Y protein. Proteins: Like in the Sparse (SP)

data case, the NFP group demonstrated similar results and in the IDP group, YJM1418 had the highest value - 0.6759.

Overall Performance (General)

The analysis of results described above shows that SES depends more on SPC while SDS depends more on SSC. Entropic affinities generally demonstrated better NMI values for all algorithms with one exception: the NMI value obtained for YJM1418 in the super-sparse data with SDS and plain affinities was not significantly higher (0.6759) than the same combination with entropic affinities (0.6727). SPC combined with entropic affinities is the best for all proteins in the dense data, 1GB1 and YJM1418 for the sparse data, and the NFP group for the super-sparse data. SES demonstrated the best results for 1L2Y and for the sparse data and 5EQJ for the sparse and super-sparse data sets. For plain affinities SDS with sparse and super-sparse data showed the best results among all algorithms for all proteins. In general also, SPC was almost always the worst algorithm to use with plain affinities.

General Graph Segmentation Results

Before we start searching relations between different execution parameters, we urge the reader to check subsection "Statistical Analysis".

Sparsity

Graph thickness: *For KNN*, the sparser the data, the more narrow graphs are produced. *For perplexity/sigma* we see the opposite trend. Both show more CT with denser data. Graph shape: *For KNN*, all results were pretty much identical, but the denser data contained more CS. Angles were also smaller. *For perplexity/sigma*, denser data contained slightly fewer straight parts.

Algorithms

Graph thickness: *For KNN*, SDS and SES showed thinner shapes than SPC. *For perplexity/sigma* SPC showed thinner shapes, SDS second, and SES was the last in this regard, but SES thickness variation was less (30% for SES and 45% for SPC). Graph shape: *For KNN*, SPC had the most (83%) straight lines, while SES had the least number (43%). There was the opposite situation with the curvature, where SPC had a small curvature and SES had a sharp curvature. *For perplexity/sigma*, there were no significant differences except that SPC had the highest number of CS, but SES had the smallest.

Affinity

Graph thickness: *For KNN*, entropic affinities exhibited twice as many narrow parts compared to plain affinities, and a very similar situation with regard to changes, so the entropic affinities were more stable. *For perplexity/sigma* the situation was the same, 68% vs 7% narrow parts for the entropic and plain affinities, but 56% vs 17% for the changes. The situation with changes can be explained since often there was just an even distribution of points that did not give any information, but was not treated as a thickness change. Graph shape: *For KNN*, the entropic affinities produced more straight lines, less CS and significantly less curvature than the plain affinities. *For perplexity/sigma* the entropic affinities contained slightly fewer straight regions.

Protein type

Graph thickness: *For KNN*, both were similar, but NFP produced more narrow pieces and significantly less changes. *For perplexity/sigma*, big difference was only with changes 33% vs 50% for for NFP and IDP. NFP had a slightly more narrow parts. Graph shape: *For KNN*, NFP contained more straight regions and less CS. For perplexity/sigma NFP still contained a little more straight regions and less A and V shapes. A more detailed analysis of graph segmentation can be found in the supplementary materials.

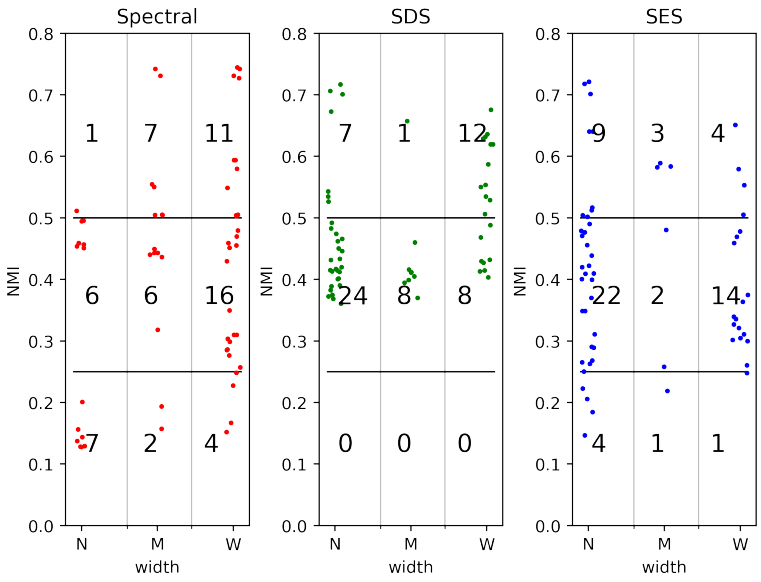


Figure 53: Relationship between the NMI values and variation for the SPC (left), SDS (middle), and SES (right) algorithms for the k-nearest neighbors (KNN) batch. Numbers in the graphs indicate the number of points in that sector.

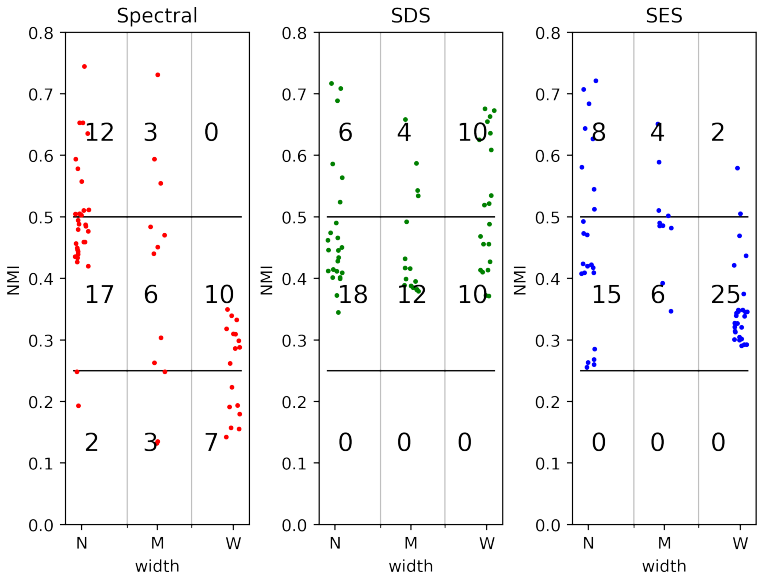


Figure 54: Relationship between the NMI values and variation for the SPC (left), SDS (middle), and SES (right) algorithms for the perplexity/sigma batch. Numbers in the graphs indicate the number of points in that sector.

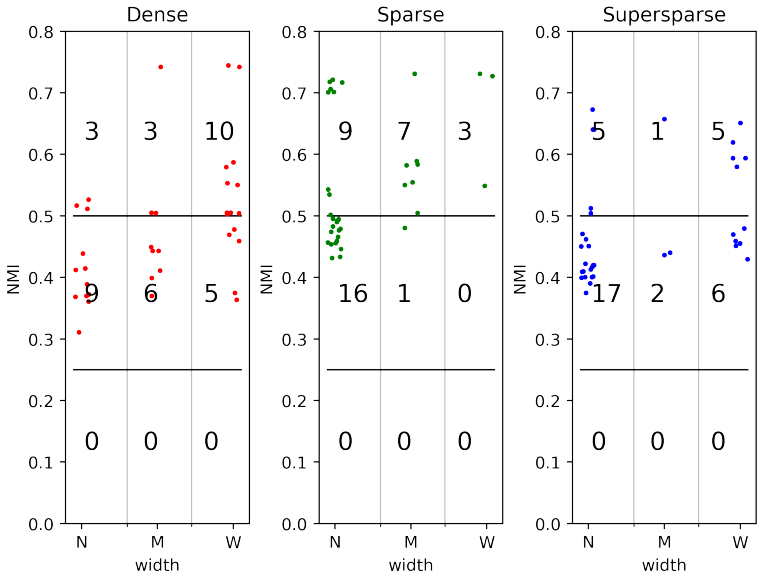


Figure 55: Relationship between the NMI values and variation for the Dense (left), Sparse (middle), and Super-sparse (right) data sets. Numbers in the graphs indicate the number of points in that sector.

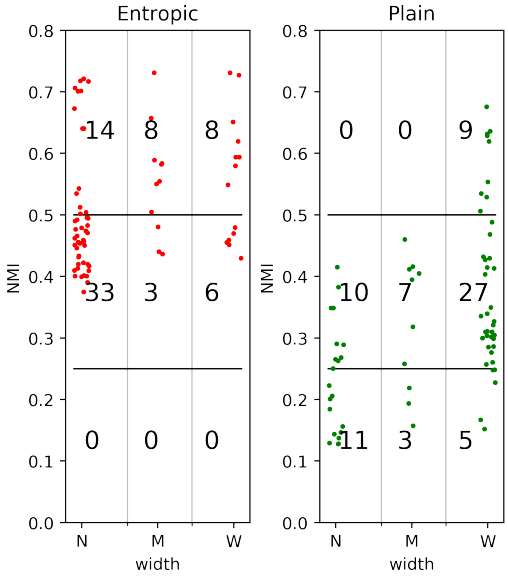


Figure 56: Relationship between the NMI values and variation for Entropic (left) and Plain (right) affinities. Numbers in the graphs indicate the number of points in that sector.

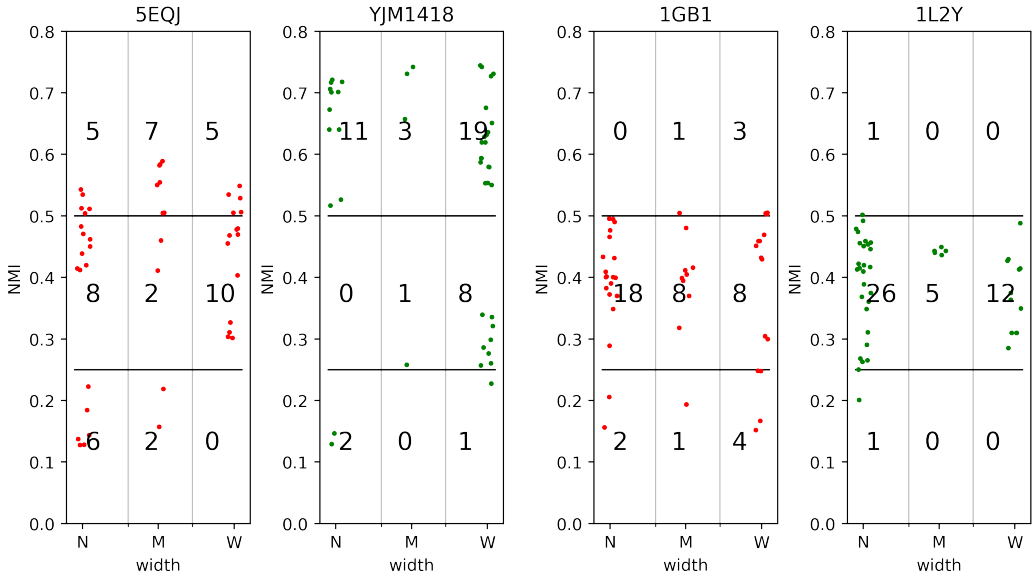


Figure 57: Relationship between the NMI values and variation for the studied proteins. Numbers in the graphs indicate the number of points in that sector.

Segmented Graph Analysis

Graphs displayed in Figures 53, 54, 55, 56, 57 on pages 119-121 have only categorical meaning. The points inside sectors were slightly (randomly) displaced along the x-axis to provide perspective concerning the number of points with similar NMI values.

Algorithm

We found that despite differences in behavior between the KNN and perplexity/sigma batches, we still can see a clear difference between algorithms inside each group. In the KNN case (Figure 53) SDS produced generally higher NMI values than SPC and SES produced more consistent results, most of which were described as 'narrow' (defined in Table 14). In the perplexity/sigma batch we may observe that both SDS and SES demonstrate generally higher NMI than SPC, but also have more variation which can be observed in Figure 54.

Density

Both sparse and super-sparse plots showed similar behavior for both KNN and perplexity/sigma. Denser data produced slightly higher NMI, which can be seen in Figure 55.

Affinity

We found that there is a clear difference between plain and entropic affinities. Entropic affinities demonstrate more narrow variation and higher NMI values than plain affinities (Figure 56). Graphs for perplexity/sigma (data not reported here, but can be found inside the git repository mentioned above) showed even stronger difference, describing most points with plain affinities as wide and most points with entropic affinities as narrow.

Proteins

Results for the KNN and perplexity/sigma batches were consistent and showed clear differences between IDP and NFP groups. The NFP group contained moderate NMI values while the IDP group contained higher NMI values. Inside the NFP group both proteins show similar behavior, while inside the IDP group YJM1418 demonstrated significantly higher NMI values than 5EQJ which can be seen in Figure 57. Consistency inside the NFP group was expected since they have very similar structure and tend to fold fast, producing similar trajectories during each simulation. Inconsistency inside the IDP group can also be explained upon a closer look at their known physical properties: YJM1418 tends to have many semi-folded shapes resulting in different trajectories that are easier to separate. On the other hand, 5EQJ tends to have many large-amplitude movements, and no particular semi-folded shapes which results in producing more chaotic trajectories that are harder to separate.

Boxplot Analysis

Dense

It is clear that SPC is a winner for this data set, having general performance much higher than SDS and SES (see Figure 58 on page 124). We may see that the variation among NFPs and 5EQJ is about the same, while YJM1418 has almost twice the variation. It is also clear that the NFPs had much lower NMI results than IDPs, especially YJM1418, as shown in Figure 59.

Sparse

For the entropic affinities, all three algorithms demonstrate a similar variance, but SES generally had higher NMI values. For plain affinities SPC had the smallest variance but also the smallest NMI values, while SDS had the highest NMI values, but also the

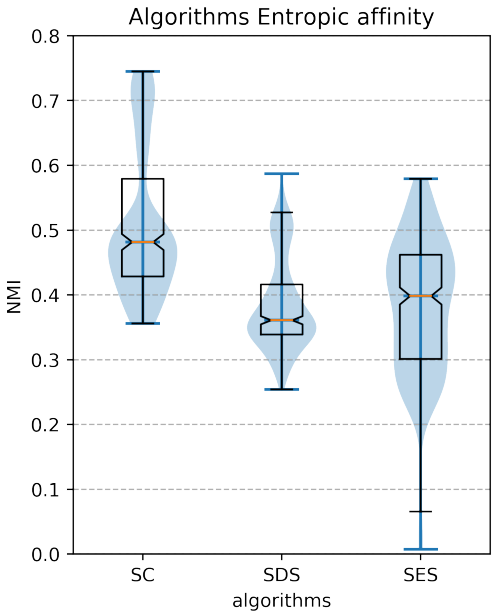


Figure 58: Relationship between the NMI values and variation for the SPC (left), SDS (middle), and SES (right) algorithms for the perplexity/sigma batch and dense data set.

highest variance. For the plain affinities SPC showed the worst NMI values, but the smallest variation. SDS and SES showed a similar variation, but SDS had a much higher NMI values (Figure 60). For sparse data and entropic affinities all proteins demonstrated small variance. NFP had similar NMI values, while 5EQJ had a slightly higher NMI value but the highest variation among all proteins and finally YJM1418 had the highest NMI values while keeping the average variance. For the plain affinities all proteins had a very high variance. It is interesting that 1L2Y had a much higher (median/average) NMI value than others (Figure 61).

Super-sparse

For the entropic affinities all three algorithms had a similar performance, but SPC had a slightly lower NMI values and SES had a slightly higher NMI values. For the

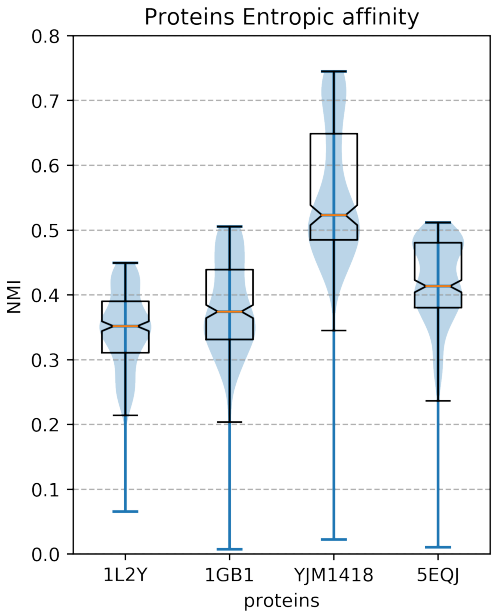


Figure 59: Relationship between the NMI values and variation for IDPs: 5EQJ, YJM1418 (left) and NFPs: 1GB1, 1L2Y (right) for the dense data set.

plain affinities while SPC had the smallest variation it also had the smallest NMI values. SDS had medium variation but significantly higher NMI values (see Figure 62). For the entropic affinities the NFP group demonstrated very similar results - low variance and lower NMI values. 5EQJ had a higher NMI value, but twice the variance, while YJM1418 had the highest NMI values and variance. For plain affinities all proteins demonstrated a very high variance, but 1L2Y had a slightly higher NMI value (see Figure 63).

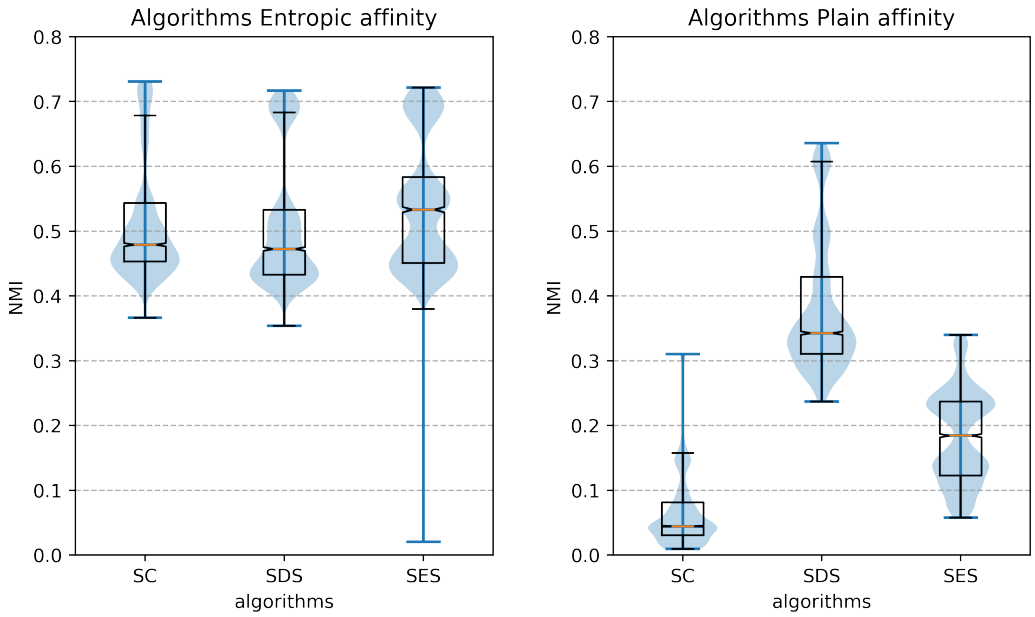


Figure 60: Relationship between the NMI values and variation for the SPC (left), SDS (middle), and SES (right) algorithms for the perplexity/sigma batch and the sparse data set.

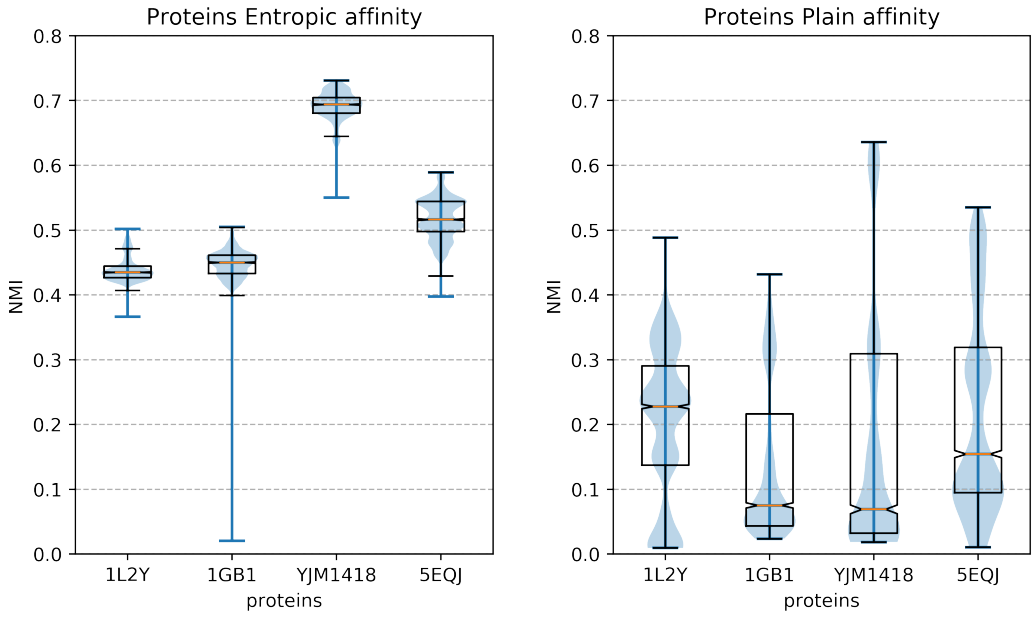


Figure 61: Relationship between the NMI values and variation for IDPs: 5EQJ, YJM1418 (left) and NFPs: 1GB1, 1L2Y (right) for the sparse data set.

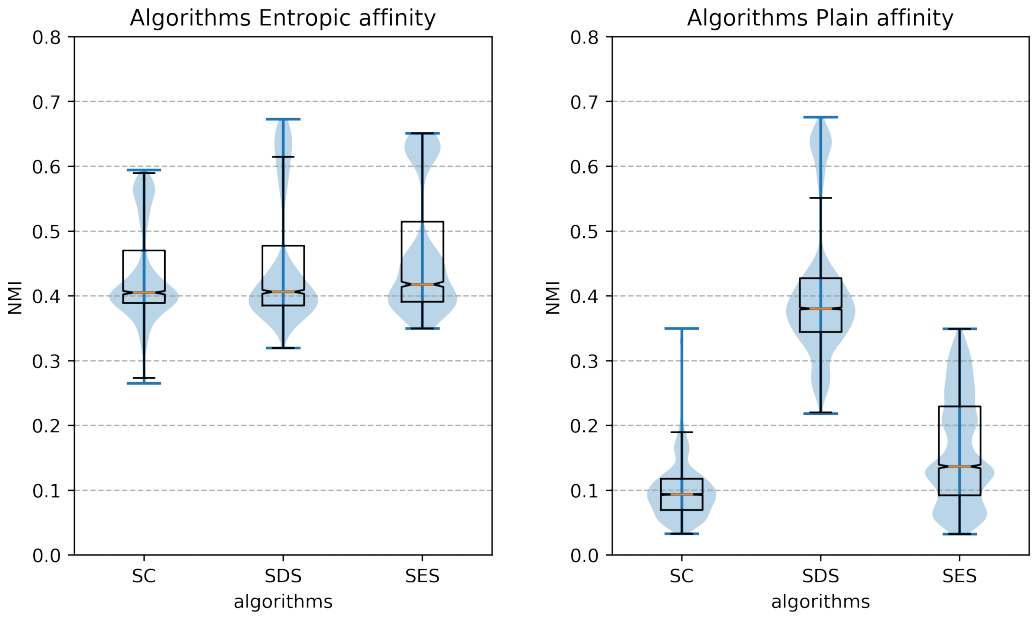


Figure 62: Relationship between the NMI values and variation for the SPC (left), SDS (middle), and SES (right) algorithms for the perplexity/sigma batch and the super-sparse data set.

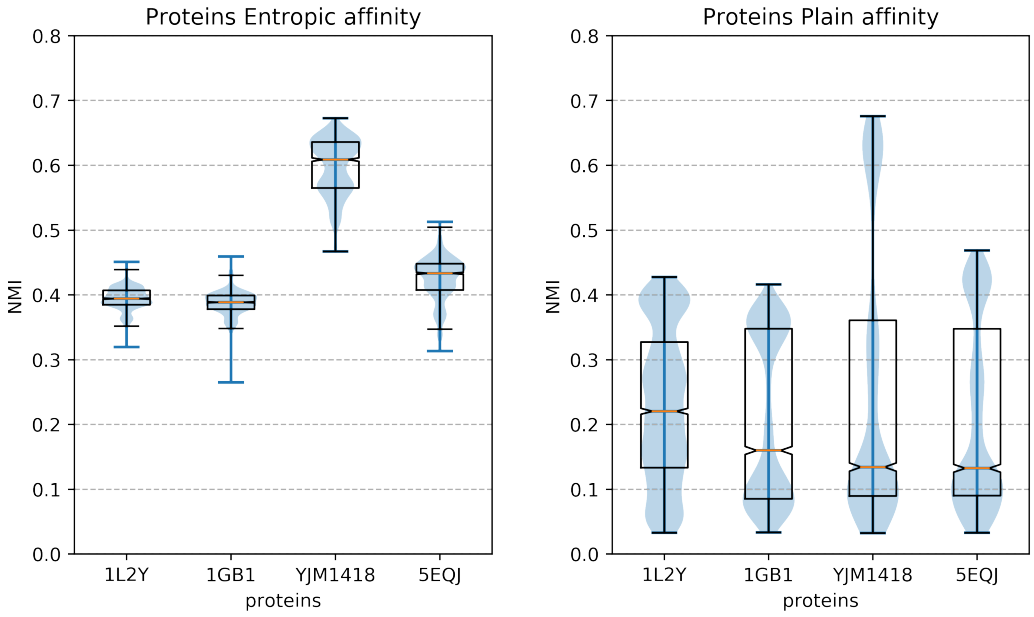


Figure 63: Relationship between the NMI values and variation for IDPs: 5EQJ, YJM1418 (left) and NFPs: 1GB1, 1L2Y (right) for the super-sparse data set.

DISCUSSION

We have performed a thorough analysis of the clustering results produced by Spectral clustering (SPC), Subspace Clustering (SSC), element-wise product (SES), and dot product (SDS) and their entropic affinities improvements on variable-density Molecular dynamics (MD) simulation data. The results section shows that entropic affinities significantly improve clustering quality and should be used instead of plain affinities for all algorithms. Hybrid solutions such as SES and SDS in most cases either improve clustering accuracy or stability of the clustering results.

We found that increasing data density significantly increases clustering time, but did not always produce better clustering accuracy. Since the entropic affinities approach is not necessarily the standard approach used in the field, our results indicate that the subspace clustering algorithm and both SDS and SES produced higher (55% more) normalized mutual information (NMI) values than SPC. Therefore, our approach of reusing results of the convex optimization solution is a geometrically well-motivated method for dealing with data displaying both subspace and nonlinear components. However, the entropic affinities results attest to the fact that much of the issue with clustering MD simulation data is due to nonuniform sampling. Additionally, it was clear that intrinsically disordered proteins (IDP) were easier to cluster than natively folded proteins (NFP) which was not surprising due to lack of simulation convergence. This result only bolsters the need for better clustering approaches such as SDS and SES. Although we concentrated on MD simulations, SDS and SES improvements should be similar for other data with similar properties. This can lead to better clustering results in areas that intensively use clustering techniques, such as text recognition, image processing, data science, etc. Although higher k generally resulted in higher NMI values, it also required more computational time. Analysis of minimum NMI results may be of interest since algorithmic stability may be more important for computationally demanding data sets.

CHAPTER IV.

SUMMARY OF CONTRIBUTIONS

In Chapter II we were able to create very fast folding trajectories for the three proteins (1L2Y, 1YRF, and 1GB1) using several different force fields (AMBER, CHARMM, GROMOS, and OPLS) and analyze their performance. We ran Replica-Exchange Molecular Dynamics (REMD) to compare with our algorithm and found that the Greedy-proximal A* (GPA*) technique not only resulted in shorter trajectories, smaller Root-mean-squared deviation (RMSD) distance to the nuclear magnetic resonance (NMR) structure without introductions of the artificial bias, but also used less computational time. We compared our trajectories to the most recently published results and found that trajectories generated with GPA* were 200-3000 times shorter.

Thus an efficient combination of the path-finding algorithms with standard Molecular dynamics (MD) indicates a possible application to other MD-related approaches like the stereochemical approach to allow efficient simulations of very large protein complexes which are not feasible to study with the regular MD because of the limited computational power of modern systems. Additionally, we discussed the problems of using only one metric to compare two conformations and proposed several new metrics to help mitigate this problem.

Such short trajectories, that were achieved without application of external force, in our opinion, are potentially the most probable in the natural environment. We are planning to apply the umbrella sampling approach in future studies to determine whether our method can also result in less total computational time and better approximation of experimental results. We hope that our approach will provide a more efficient way for scientific community will use our approach to study folding pathway events more productively.

Additionally, we plan to tune parameters such as the revised distance metrics set, duration of single simulations, number of seeds, metric usage order, maximum duration of one metric, determination of the energy barrier and trying avoidance strategies, etc.

While efficiency was improved using the parameters explored above, the further development of our approach may improve efficiency in various areas of research related to the study of proteins' conformation changes.

Adoption of the presented algorithm allows to perform either more simulations within the same time or simulate folding of larger proteins. While more trajectories allow catch sample rare events, larger proteins study would help generate folding pathways with respect to interactions absent in coarser methods.

In Chapter III we showed that the protein trajectory data has nonlinear and subspace properties which our approach could use to divide data into clusters with higher precision. We conclude that the protein trajectory data can be clustered more precisely thus improving the quality of the analysis.

Analysis of the results clearly showed that with the standard Gaussian affinities our hybrid approach outperformed past clustering algorithms for all tested proteins. Additionally, it cat improve the clustering results even with "preprocessed" entropic affinities. We hope that adoption of our algorithm by other scientists would result in less analysis error due to poor clustering algorithm performance.

Because of the client-server architecture implemented with the asynchronous execution, the best parameter search can be scaled without any significant overhead which will allow scientists to study best clustering parameters of known processes and apply these parameters to similar problems.

One potential future application could be to help in the search for local minima by GPA* by the determination of the cluster of nodes that are similar but have a common problem - inability to reduce distance metric to the NMR conformation. Such information may help reduce the chances of selecting nodes from such a cluster for future simulations, and selection of more promising nodes that are further from the energy well.

REFERENCES

Amaldi, E. and Kann, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science, 209(1-2)*, 237–260.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 1027–1035. Society for Industrial and Applied Mathematics.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM), 45(6)*, 891–923.

Aumasson, J.-P., Neves, S., Wilcox-O'Hearn, Z., and Winnerlein, C. (2013). Blake2: simpler, smaller, fast as md5. In *International Conference on Applied Cryptography and Network Security*, 119–135. Springer.

Balasubramanian, V., Bethune, I., Shkurti, A., Breitmoser, E., Hruska, E., Clementi, C., Laughton, C., and Jha, S. (2016). Extasy: Scalable and flexible coupling of md simulations and advanced sampling techniques. In *2016 IEEE 12th International Conference on e-Science (e-Science)*, 361–370. IEEE.

Berendsen, H. J., Postma, J. v., van Gunsteren, W. F., DiNola, A., and Haak, J. (1984). Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, *81(8)*, 3684–3690.

Bernardi, R. C., Melo, M. C., and Schulten, K. (2015). Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochimica et Biophysica Acta (BBA)-General Subjects*, *1850(5)*, 872–877.

Best, C. and Hege, H.-C. (2002). Visualizing and identifying conformational ensembles in molecular dynamics trajectories. *Computing in Science & Engineering*, *4(3)*, 68–75.

Bhowmik, D. and Ramanathan, A. (2018). Identifying metastable states of protein folding with deep clustering techniques. *Biophysical Journal*, *114(3)*, 42a.

Bottaro, S., Bengtsen, T., and Lindorff-Larsen, K. (2018). Integrating molecular simulation and experimental data: A bayesian/maximum entropy reweighting approach. *bioRxiv*, page 457952.

Bowman, G. R. and Pande, V. S. (2010). Protein folded states are kinetic hubs. *Proceedings of the National Academy of Sciences*, *107(24)*, 10890–10895.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. New York, NY: Cambridge University Press, USA.

Bradley, P., Malmström, L., Qian, B., Schonbrun, J., Chivian, D., Kim, D. E., Meiler, J., Misura, K. M., and Baker, D. (2005a). Free modeling with rosetta in casp6. *Proteins: Structure, Function, and Bioinformatics*, *61(S7)*, 128–134.

Bradley, P., Misura, K. M., and Baker, D. (2005b). Toward high-resolution de novo structure prediction for small proteins. *Science*, *309(5742)*, 1868–1871.

Bryant, F. B. and Yarnold, P. R. (1995). Principal-components analysis and exploratory and confirmatory factor analysis.

Callender, R. and Dyer, R. B. (2002). Probing protein dynamics using temperature jump relaxation spectroscopy. *Current opinion in structural biology*, *12(5)*, 628–633.

Car, R. and Parrinello, M. (1985). Unified approach for molecular dynamics and density-functional theory. *Physical Review Letters*, *55(22)*, 2471.

Carnevali, P., Tóth, G., Toubassi, G., and Meshkat, S. N. (2003). Fast protein structure prediction using monte carlo simulations with modal moves. *Journal of the American Chemical Society*, *125(47)*, 14244–14245.

Carugo, O. (2007). Statistical validation of the root-mean-square-distance, a measure of protein structural proximity. *Protein Engineering, Design & Selection*, *20(1)*, 33–37.

Case, D., Ben-Shalom, I., Brozell, S., Cerutti, D., III, T. C., Cruzeiro, V., Darden, T., Duke, R., Ghoreishi, D., Gilson, M., Gohlke, H., Goetz, A., Greene, D., Harris, R., Homeyer, N., Izadi, S., Kovalenko, A., Kurtzman, T., Lee, T., LeGrand, S., Li, P., Lin, C., Liu, J., Luchko, T., Luo, R., Mermelstein, D., Merz, K., Miao, Y., Monard, G., Nguyen, C., Nguyen, H., Omelyan, I., Onufriev, A., Pan, F., Qi, R., Roe, D., Roitberg, A., Sagui, C., Schott-Verdugo, S., Shen, J., Simmerling, C., Smith, J., Salomon-Ferrer, R., Swails, J., Walker, R., Wang, J., Wei, H., Wolf, R., Wu, X., Xiao, L., York, D., and Kollman, P. (2018). Amber 2018, University of California, San Francisco.

Chen, Y., Ding, F., Nie, H., Serohijos, A. W., Sharma, S., Wilcox, K. C., Yin, S., and Dokholyan, N. V. (2008). Protein folding: then and now. *Archives of Biochemistry and Biophysics*, *469(1)*, 4–19.

Chiu, T. K., Kubelka, J., Herbst-Irmer, R., Eaton, W. A., Hofrichter, J., and Davies, D. R. (2005). High-resolution x-ray crystal structures of the villin headpiece subdomain,

an ultrafast folding protein. *Proceedings of the National Academy of Sciences, 102(21),* 7517–7522.

Chowdhury, S., Lee, M. C., and Duan, Y. (2004). Characterizing the rate-limiting step of trp-cage folding by all-atom molecular dynamics simulations. *The Journal of Physical Chemistry B, 108(36),* 13855–13865.

Daggett, V. and Levitt, M. (1993). Protein unfolding pathways explored through molecular dynamics simulations. *Journal of Molecular Biology, 232(2),* 600–619.

Daidone, I., Amadei, A., Roccatano, D., and Di Nola, A. (2003). Molecular dynamics simulation of protein folding by essential dynamics sampling: folding landscape of horse heart cytochrome c. *Biophysical Journal, 85(5),* 2865–2871.

(dalke@ks.uiuc.edu), A. D.

Dama, J. F., Parrinello, M., and Voth, G. A. (2014). Well-tempered metadynamics converges asymptotically. *Physical Review Letters, 112(24),* 240602.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research, 17(83),* 1–5.

Dill, K. A. and MacCallum, J. L. (2012). The protein-folding problem, 50 years on. *Science, 338(6110),* 1042–1046.

Dinner, A. R. and Karplus, M. (1999). Is protein unfolding the reverse of protein folding? a lattice simulation analysis. *Journal of Molecular Biology, 292(2),* 403–419.

Domahidi, A., Chu, E., and Boyd, S. (2013). ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC),* 3071–3076.

Doran, J. E. and Michie, D. (1966). Experiments with the graph traverser program. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, *294(1437)*, 235–259.

Eleftheriou, M., Rayshubski, A., Pitera, J. W., Fitch, B. G., Zhou, R., and Germain, R. S. (2006). Parallel implementation of the replica exchange molecular dynamics algorithm on blue gene/l. In *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, 8–pp. IEEE.

Elhamifar, E. and Vidal, R. (2012). Sparse subspace clustering: Algorithm, theory, and applications. *CoRR*, abs/1203.1005.

Ensign, D. L., Kasson, P. M., and Pande, V. S. (2007). Heterogeneity even at the speed limit of folding: large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. *Journal of Molecular Biology*, *374(3)*, 806–816.

Espanol, P. and Warren, P. (1995). Statistical mechanics of dissipative particle dynamics. *EPL (Europhysics Letters)*, *30(4)*, 191.

Estévez, P. A., Tesmer, M., Perez, C. A., and Zurada, J. M. (2009). Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, *20(2)*, 189–201.

Farrell, D. W., Speranskiy, K., and Thorpe, M. (2010). Generating stereochemically acceptable protein pathways. *Proteins: Structure, Function, and Bioinformatics*, *78(14)*, 2908–2921.

Felner, A. (2011). Position paper: Dijkstra's algorithm versus uniform cost search or a case against dijkstra's algorithm. In *Fourth annual symposium on combinatorial search*.

Finkelstein, A. V. (1997). Can protein unfolding simulate protein folding? *Protein Engineering*, *10(8)*, 843–845.

Forouzan, B. (2010). *TCP/IP Protocol Suite*. McGraw-Hill, Inc., New York, NY, USA, 4 edition.

Freddolino, P. L., Park, S., Roux, B., and Schulten, K. (2009). Force field bias in protein folding simulations. *Biophysical Journal*, *96(9)*, 3772–3780.

Frishman, D. and Argos, P. (1995). Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Bioinformatics*, *23(4)*, 566–579.

Gidalevitz, T., Ben-Zvi, A., Ho, K. H., Brignull, H. R., and Morimoto, R. I. (2006). Progressive disruption of cellular protein folding in models of polyglutamine diseases. *Science*, *311(5766)*, 1471–1474.

Goffeau, A., Barrell, B. G., Bussey, H., Davis, R., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J., Jacq, C., Johnston, M., et al. (1996). Life with 6000 genes. *Science*, *274(5287)*, 546–567.

Golub, G. H. and Reinsch, C. (1971). Singular value decomposition and least squares solutions. In *Linear Algebra*, 134–151. Springer.

Gronenborn, A. M., Filpula, D. R., Essig, N. Z., Achari, A., Whitlow, M., Wingfield, P. T., and Clore, G. M. (1991). A novel, highly stable fold of the immunoglobulin binding domain of streptococcal protein G. *Science*, *253(5020)*, 657–661.

Groot, R. D. and Warren, P. B. (1997). Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of Chemical Physics*, *107(11)*, 4423–4435.

Gustafson, P. (1998). A guided walk metropolis algorithm. *Statistics and Computing*, *8(4)*, 357–364.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, *4(2)*, 100–107.

Hess, B., Bekker, H., Berendsen, H. J., and Fraaije, J. G. (1997). Lincs: a linear constraint solver for molecular simulations. *Journal of Computational Chemistry*, *18(12)*, 1463–1472.

Hinton, G. E. and Roweis, S. T. (2003). Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, 857–864.

Hoffmann, D. and Knapp, E.-W. (1996). Polypeptide folding with off-lattice monte carlo dynamics: the method. *European Biophysics Journal*, *24(6)*, 387–403.

Huang, J. and MacKerell Jr, A. D. (2013). Charmm36 all-atom additive protein force field: Validation based on comparison to nmr data. *Journal of Computational Chemistry*, *34(25)*, 2135–2145.

Huang, R., Lo, L.-T., Wen, Y., Voter, A. F., and Perez, D. (2017). Cluster analysis of accelerated molecular dynamics simulations: A case study of the decahedron to icosahedron transition in pt nanoparticles. *The Journal of Chemical Physics*, *147(15)*, 152717.

Huang, S.-Y. and Zou, X. (2010). Advances and challenges in protein-ligand docking. *International Journal of Molecular Sciences*, *11(8)*, 3016–3034.

Huang, Y., Zhang, X., Ma, Z., Li, W., Zhou, Y., Zhou, J., Zheng, W., and Sun, C. Q. (2013). Size, separation, structural order, and mass density of molecules packing in water and ice. *Scientific Reports*, 3:3005.

Humphrey, W., Dalke, A., and Schulten, K. (1996). VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9(3)*, 90–95.

Izrailev, S., Stepaniants, S., Isralewitz, B., Kosztin, D., Lu, H., Molnar, F., Wriggers, W., and Schulten, K. (1999). Steered molecular dynamics. In *Computational molecular dynamics: challenges, methods, ideas*, 39–65. Springer.

Jauch, R., Yeo, H. C., Kolatkar, P. R., and Clarke, N. D. (2007). Assessment of casp7 structure predictions for template free targets. *Proteins: Structure, Function, and Bioinformatics*, *69(S8)*, 57–67.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python. [Online; accessed 09/06/2019].

Jorgensen, W. L., Maxwell, D. S., and Tirado-Rives, J. (1996). Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, *118(45)*, 11225–11236.

Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, *22(12)*, 2577–2637.

KARDOŠ, L. (2010). *Numerická knihovna pro Python*. PhD thesis, Masarykova univerzita, Fakulta informatiky.

Karpen, M. E., Tobias, D. J., and Brooks III, C. L. (1993). Statistical clustering techniques for the analysis of long molecular dynamics trajectories: analysis of 2.2-ns trajectories of ypgdv. *Biochemistry*, *32(2)*, 412–420.

Karplus, K., Barrett, C., and Hughey, R. (1998). Hidden markov models for detecting remote protein homologies. *Bioinformatics (Oxford, England)*, 14(10), 846–856.

Karplus, M., Ichiye, T., and Pettitt, B. (1987). Configurational entropy of native proteins. *Biophysical Journal*, *52(6)*, 1083–1085.

Karplus, M. and Kuriyan, J. (2005). Molecular dynamics and protein function. *Proceedings of the National Academy of Sciences of the United States of America*, *102(19)*, 6679–85.

Kästner, J. (2011). Umbrella sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *1(6)*, 932–942.

Kawai, H., Kikuchi, T., and Okamoto, Y. (1989). A prediction of tertiary structures of peptide by the monte carlo simulated annealing method. *Protein Engineering, Design and Selection*, *3(2)*, 85–94.

Klepeis, J. L., Lindorff-Larsen, K., Dror, R. O., and Shaw, D. E. (2009). Long-timescale molecular dynamics simulations of protein structure and function. *Current opinion in structural biology*, *19(2)*, 120–127.

Klepeis, J. L., Wei, Y., Hecht, M. H., and Floudas, C. A. (2005). Ab initio prediction of the three-dimensional structure of a de novo designed protein: A double-blind case study. *Proteins: Structure, Function, and Bioinformatics*, *58(3)*, 560–570.

Knuth, D. E. (1974). Postscript about np-hard problems. *ACM SIGACT News*, *6(2)*, 15–16.

Kolinski, A. and Skolnick, J. (1994). Monte carlo simulations of protein folding. i. lattice model and interaction scheme. *Proteins: Structure, Function, and Bioinformatics*, *18(4)*, 338–352.

Kufareva, I. and Abagyan, R. (2011). Methods of protein structure comparison. In *Homology Modeling*, 231–257. Springer.

Kühn, O. and Wöste, L. (2007). *Analysis and control of ultrafast photoinduced reactions, volume 87*. Springer Science & Business Media.

Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H., and Kollman, P. A. (1992). The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *Journal of Computational Chemistry, 13(8)*, 1011–1021.

Kurant, M., Markopoulou, A., and Thiran, P. (2010). On the bias of bfs (breadth first search). In *2010 22nd International Teletraffic Congress (lTC 22)*, 1–8. IEEE.

Lindorff-Larsen, K., Piana, S., Dror, R. O., and Shaw, D. E. (2011). How fast-folding proteins fold. *Science, 334(6055)*, 517–520.

Lindorff-Larsen, K., Piana, S., Palmo, K., Maragakis, P., Klepeis, J. L., Dror, R. O., and Shaw, D. E. (2010). Improved side-chain torsion potentials for the amber ff99sb protein force field. *Proteins: Structure, Function, and Bioinformatics, 78(8)*, 1950–1958.

Liu, S.-Q., Ji, X.-L., Tao, Y., Tan, D.-Y., Zhang, K.-Q., and Fu, Y.-X. (2012). Protein folding, binding and energy landscape: A synthesis. In *Protein Engineering*. InteChOpen.

Liu, Y., Prigozhin, M. B., Schulten, K., and Gruebele, M. (2014). Observation of complete pressure-jump protein refolding in molecular dynamics simulation and experiment. *Journal of the American Chemical Society, 136(11)*, 4265–4272.

Lloyd and Stuart (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory, 28(2)*, 129–137.

MacKerell Jr, A. D., Bashford, D., Bellott, M., Dunbrack Jr, R. L., Evanseck, J. D., Field, M. J., Fischer, S., Gao, J., Guo, H., Ha, S., et al. (1998). All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B, 102(18*, 3586–3616.

Madkour, A., Aref, W. G., Rehman, F. U., Rahman, M. A., and Basalamah, S. (2017). A survey of shortest-path algorithms. *arXiv preprint arXiv:1705.02044*.

Maple, J. R., Dinur, U., and Hagler, A. T. (1988). Derivation of force fields for molecular mechanics and dynamics from ab initio energy surfaces. *Proceedings of the National Academy of Sciences*, *85(15)*, 5350–5354.

Maximova, T., Moffatt, R., Ma, B., Nussinov, R., and Shehu, A. (2016). Principles and overview of sampling methods for modeling macromolecular structure and dynamics. *PLoS Computational Biology*, *12(4)*, e1004619.

Mayor, U., Johnson, C. M., Daggett, V., and Fersht, A. R. (2000). Protein folding and unfolding in microseconds to nanoseconds by experiment and simulation. *Proceedings of the National Academy of Sciences*, *97(25)*, 13518–13522.

McKinney, W. (2010). Data structures for statistical computing in python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, 51–56.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association*, 44(247), 335–341.

Nau, D. S. (1983). Expert computer systems. *Computer*, *16(2)*, 63–85.

Neidigh, J. W., Fesinmeyer, R. M., and Andersen, N. H. (2002). Designing a 20-residue protein. *Nature Structural and Molecular Biology*, 9(6), page 425.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 849–856.

of Vision (TM) Raytracer., P. (2004).

Ołdziej, S., Czaplewski, C., Liwo, A., Chinchio, M., Nanias, M., Vila, J., Khalili, M., Arnautova, Y., Jagielska, A., Makowski, M. o., et al. (2005). Physics-based protein-structure prediction using a hierarchical protocol based on the unres force field: assessment in two blind tests. *Proceedings of the National Academy of Sciences, 102(21)*, 7547–7552.

Oliphant, T. (2006–). NumPy: A guide to NumPy. USA: Trelgol Publishing. [Online; accessed 09/06/2019].

Oostenbrink, C., Villa, A., Mark, A. E., and Van Gunsteren, W. F. (2004). A biomolecular force field based on the free enthalpy of hydration and solvation: the gromos force-field parameter sets 53a5 and 53a6. *Journal of Computational Chemistry, 25(13)*, 1656–1676.

Pall, S., Abraham, M. J., Kutzner, C., Hess, B., and Lindahl, E. (2014). Tackling exascale software challenges in molecular dynamics simulations with gromacs. In *International Conference on Exascale Applications and Software*, 3–27. Springer.

Parker, S. G., Bigler, James, Dietrich, Andreas, Friedrich, Heiko, Hoberock, Jared, Luebke, David, McAllister, David, McGuire, Morgan, Morley, Keith, Robison, Austin, et al. (2010). Optix: a general purpose ray tracing engine. In *Acm transactions on graphics (tog), volume 29*, page 66. ACM.

Patriksson, A. and van der Spoel, D. (2008). A temperature predictor for parallel tempering simulations. *Physical Chemistry Chemical Physics, 10(15)*, 2073–2077.

Peter, E. K. and Shea, J.-E. (2017). An adaptive bias–hybrid md/kmc algorithm for protein folding and aggregation. *Physical Chemistry Chemical Physics, 19(26)*, 17373–17382.

Phillips, J., Lau, E., Colvin, M., and Newsam, S. (2008). *Analyzing dynamical simulations of intrinsically disordered proteins using spectral clustering*, 17–24.

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R. D., Kale, L., and Schulten, K. (2005). Scalable molecular dynamics with namd. *Journal of Computational Chemistry*, *26(16)*, 1781–1802.

Phillips, J. L. (2012). Validation of computational approaches for studying disordered and unfolded protein dynamics using polymer models.

Phillips, J. L., Colvin, M. E., and Newsam, S. (2011). Validating clustering of molecular dynamics simulations using polymer models. *BMC Bioinformatics*, *12(1)*, page 445.

Qiu, D., Shenkin, P. S., Hollinger, F. P., and Still, W. C. (1997). The gb/sa continuum model for solvation. a fast analytical method for the calculation of approximate born radii. *The Journal of Physical Chemistry A*, *101(16)*, 3005–3014.

Rajan, A., Freddolino, P. L., and Schulten, K. (2010). Going beyond clustering in md trajectory analysis: an application to villin headpiece folding. *PLoS One*, *5(4)*, e9890.

Ramachandran, G. N. (1963). Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.*, *7*, 95–99.

Ramey, C. and Fox, B. (2003). *Bash reference manual*. Network Theory Limited.

Rappé, A. K., Casewit, C. J., Colwell, K., Goddard III, W. A., and Skiff, W. M. (1992). Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, *114(25)*, 10024–10035.

Rauscher, S. and Pomès, R. (2010). Molecular simulations of protein disorder. *Biochemistry and Cell Biology*, *88(2)*, 269–290.

Reddy, H. (2013). Pathfinding—dijkstra's and a* algorithm's. *Int. J. IT Eng*, 1–15.

Rivest, R. (1992). The md5 message-digest algorithm.

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Sakuraba, S., Joti, Y., and Kitao, A. (2010). Detecting coupled collective motions in protein by independent subspace analysis. *The Journal of Chemical Physics, 133(18)*, page 11B604.

Šali, A. and Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology, 234(3)*, 779–815.

Scheraga, H. A., Khalili, M., and Liwo, A. (2007). Protein-folding dynamics: overview of molecular simulation techniques. *Annu. Rev. Phys. Chem., 58*, 57–83.

Selkoe, D. J. (2003). Folding proteins in fatal ways. *Nature, 426(6968)*, page 900.

Settanni, G. and Fersht, A. R. (2008). High temperature unfolding simulations of the trpz1 peptide. *Biophysical Journal, 94(11)*, 4444–4453.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107.

Sittel, F. and Stock, G. (2016). Robust density-based clustering to identify metastable conformational states of proteins. *Journal of Chemical Theory and Computation, 12(5)*, 2426–2435.

Snow, C. D., Sorin, E. J., Rhee, Y. M., and Pande, V. S. (2005). How well can simulation predict protein folding kinetics and thermodynamics? *Annu. Rev. Biophys. Biomol. Struct., 34*, 43–69.

Soltani, A. R., Tawfik, H., Goulermas, J. Y., and Fernando, T. (2002). Path planning in construction sites: performance evaluation of the dijkstra, a*, and ga search algorithms. *Advanced Engineering Informatics, 16(4)*, 291–303.

Stefani, M. (2008). Protein folding and misfolding on surfaces. *International Journal of Molecular Sciences, 9(12)*, 2515–2542.

Steinbach, M., Ertöz, L., and Kumar, V. (2004). The challenges of clustering high dimensional data. In *New Directions in Statistical Physics*, 273–309. Springer.

Steinberg, I. Z. and Scheraga, H. A. (1963). Entropy changes accompanying association reactions of proteins. *Journal of Biological Chemistry, 238(1)*, 172–181.

Sticklen, M. B. (2008). Plant genetic engineering for biofuel production: towards affordable cellulosic ethanol. *Nature Reviews Genetics, 9(6)*, page 433.

Stone, J. E. (1998). An efficient library for parallel ray tracing and animation.

Sugita, Y. and Okamoto, Y. (1999). Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters, 314(1-2)*, 141–151.

Team, S. D. (2010–). Sqlite. [Online; accessed 09/06/2019].

Ulmschneider, J. P., Ulmschneider, M. B., and Di Nola, A. (2006). Monte carlo vs molecular dynamics for all-atom polypeptide folding simulations. *The Journal of Physical Chemistry B, 110(33)*, 16733–16742.

van Gunsteren, W. F., Daura, X., Hansen, N., Mark, A. E., Oostenbrink, C., Riniker, S., and Smith, L. J. (2018). Validation of molecular simulation: an overview of issues. *Angewandte Chemie International Edition, 57(4)*, 884–902.

Van Rossum, G. and Drake, F. L. (2011). *The python language reference manual*. Network Theory Ltd.

Vidal, R. (2010). A tutorial on subspace clustering. 28.

Vladymyrov, M. and Carreira-perpinan, M. (2013). Entropic affinities: Properties and efficient numerical computation. In Dasgupta, S. and Mcallester, D., editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, 477–485. JMLR Workshop and Conference Proceedings.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17(4)*, 395–416.

Wales, D. (2003). *Energy landscapes: Applications to clusters, biomolecules and glasses*. Cambridge University Press.

Wang, J., Cieplak, P., and Kollman, P. A. (2000). How well does a restrained electrostatic potential (resp) model perform in calculating conformational energies of organic and biological molecules? *Journal of Computational Chemistry*, *21(12)*, 1049–1074.

Weisstein, E. W. (2014). Laplacian matrix. from mathworld–a wolfram web resource.

Wirth, A. J., Liu, Y., Prigozhin, M. B., Schulten, K., and Gruebele, M. (2015). Comparing fast pressure jump and temperature jump protein folding experiments and simulations. *Journal of the American Chemical Society*, *137(22)*, 7152–7159.

Yamada, J., Phillips, J. L., Patel, S., Goldfien, G. A., Calestagne-Morelli, A., Huang, H., de la Reza, R., Acheson, J. F., Krishnan, V., Newsam, S. D., Gopinathan, A., Lau, E. Y., Colvin, M. E., Uversky, V. N., and Rexach, M. F. (2010). A bimodal distribution of two distinct categories of intrinsically disordered structures with separate functions in fg nucleoporins. *Molecula*, *9* 10, 2205–2224.

Yang, Y., Faraggi, E., Zhao, H., and Zhou, Y. (2011). Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics*, *27(15)*, 2076–2082.

Yon, J. (2001). Protein folding: a perspective for biology, medicine and biotechnology. *Brazilian Journal of Medical and Biological Research, 34(4),* 419–435.

Yu, G. and Yang, J. (1998). On the robust shortest path problem. *Computers & Operations Research, 25(6),* 457–468.

Appendices

APPENDIX A

Extra Tables for Chapter II

Table 19: Configuration properties of the GPA* experiment for all proteins MD simulation with all force fields.

Protein	N_{aa}	N_{atoms}	N_{wat}	Na^+	Cl^-	Force field	Water model
1L2Y	20	304	2730	84	85	AMBER	TIP3P
1L2Y	20	304	2730	84	85	CHARMM	TIP3P
1L2Y	20	304	2730	84	85	OPLS	TIP3P
1L2Y	20	198	2678	80	81	GROMOS	SPCE
1GB1	56	855	3364	111	107	AMBER	TIP3P
1GB1	56	855	3364	111	107	CHARMM	TIP3P
1GB1	56	855	3364	111	107	OPLS	TIP3P
1GB1	56	562	3391	111	107	GROMOS	SPCE
1YRF	76	582	2350	75	77	AMBER	TIP3P
1YRF	76	582	2350	75	77	CHARMM	TIP3P
1YRF	76	582	2350	75	77	OPLS	TIP3P
1YRF	76	383	2263	70	72	GROMOS	SPCE

Table 20: Temperature distribution for REMD experiment of 1L2Y folding with AMBER, CHARMM, and OPLS force fields.

	Temperature, K	μ, kJ/mol	σ, kJ/mol	μ_{12}, kJ/mol	σ_{12}, kJ/mol	P_{12}
1	300.00	-105568	326.95			
2	302.87	-104964	328.31	603.6	463.34	0.2500
3	305.77	-104356	329.68	608.3	465.27	0.2500
4	308.69	-103743	331.05	613.0	467.21	0.2500
5	311.63	-103125	332.44	617.7	469.16	0.2501
6	314.59	-102502	333.84	622.6	471.13	0.2499
7	317.57	-101875	335.25	627.4	473.12	0.2501
8	320.58	-101242	336.67	632.3	475.12	0.2500
9	323.62	-100605	338.10	637.2	477.14	0.2500
10	326.67	-99963	339.55	642.1	479.17	0.2500
11	329.75	-99316	341.00	647.3	481.22	0.2499
12	332.86	-98663	342.47	652.2	483.29	0.2500
13	335.98	-98006	343.94	657.3	485.37	0.2500
14	339.13	-97344	345.43	662.5	487.46	0.2500
15	342.31	-96676	346.93	667.6	489.58	0.2500
16	345.51	-96003	348.44	672.7	491.71	0.2501
17	348.74	-95326	349.97	678.1	493.85	0.2500
18	351.99	-94643	351.50	683.4	496.01	0.2499
19	355.26	-93955	353.05	688.7	498.19	0.2499
20	358.56	-93261	354.61	694.1	500.39	0.2500
21	361.90	-92560	356.18	699.4	502.60	0.2500
22	365.25	-91855	357.77	704.9	504.84	0.2500
23	368.63	-91145	359.36	710.5	507.09	0.2500
24	372.04	-90429	360.97	716.0	509.35	0.2500
25	375.48	-89707	362.59	721.6	511.64	0.2500
26	378.93	-88980	364.23	727.3	513.94	0.2499
27	382.42	-88247	365.87	732.8	516.26	0.2500
28	385.94	-87509	367.53	738.6	518.60	0.2500
29	389.48	-86764	369.21	744.3	520.95	0.2500
30	393.05	-86015	370.89	750.3	523.33	0.2499
31	396.65	-85258	372.59	756.0	525.72	0.2500
32	400.00	-84553	374.17	705.0	528.04	0.2947

Table 21: Configuration properties of the REMD experiment for 1L2Y protein MD simulation with AMBER, CHARMM, OPLS force fields.

Variable	Value
P_{des}	0.25
Temperature range, K	300 - 400
Number of water molecules	2730
Number of protein atoms	304
Number of hydrogens in protein	~156
Number of constraints	~304
Number of vsites	~0
Number of DOF	~25178
Energy loss due to constraints	1.26 (kJ/mol K)

Table 22: Configuration properties of the REMD experiment for 1L2Y protein MD simulation with GROMOS force field.

Variable	Value
P_{des}	0.25
Temperature range, K	300 - 400
Number of water molecules	2678
Number of protein atoms	198
Number of hydrogens in protein	~102
Number of constraints	~198
Number of vsites	~0
Number of DOF	~24498
Energy loss due to constraints	0.82 (kJ/mol K)

Table 23: Temperature distribution for REMD experiment of 1L2Y folding with GROMOS force field.

	Temperature, K	μ, kJ/mol	σ, kJ/mol	μ_{12}, kJ/mol	σ_{12}, kJ/mol	P_{12}
1	300.00	-101548	322.51			
2	302.90	-100952	323.86	596.2	457.05	0.2499
3	305.83	-100351	325.22	600.8	458.97	0.2500
4	308.78	-99746	326.60	605.6	460.91	0.2499
5	311.76	-99136	327.98	610.2	462.86	0.2501
6	314.76	-98520	329.38	615.0	464.83	0.2501
7	317.78	-97901	330.79	620.0	466.81	0.2499
8	320.82	-97276	332.20	624.9	468.81	0.2500
9	323.89	-96646	333.63	629.7	470.82	0.2501
10	326.98	-96011	335.07	634.7	472.85	0.2500
11	330.10	-95371	336.53	639.7	474.89	0.2500
12	333.25	-94726	337.99	644.7	476.96	0.2500
13	336.41	-94076	339.47	649.9	479.04	0.2499
14	339.61	-93421	340.95	654.9	481.13	0.2500
15	342.82	-92761	342.45	660.1	483.24	0.2500
16	346.07	-92094	343.97	665.3	485.37	0.2500
17	349.34	-91424	345.49	670.5	487.52	0.2500
18	352.63	-90748	347.02	676.0	489.68	0.2499
19	355.95	-90067	348.57	681.2	491.86	0.2500
20	359.30	-89380	350.13	686.6	494.05	0.2500
21	362.67	-88688	351.70	692.0	496.27	0.2500
22	366.07	-87990	353.28	697.4	498.50	0.2501
23	369.50	-87287	354.88	703.0	500.75	0.2500
24	372.94	-86580	356.48	708.6	503.01	0.2500
25	376.42	-85866	358.10	714.2	505.29	0.2499
26	379.92	-85147	359.73	719.8	507.59	0.2500
27	383.46	-84421	361.38	725.5	509.91	0.2500
28	387.02	-83690	363.04	731.3	512.25	0.2500
29	390.62	-82953	364.71	737.1	514.60	0.2499
30	394.23	-82210	366.40	742.9	516.98	0.2500
31	397.89	-81461	368.10	748.7	519.37	0.2500
32	400.00	-81027	369.09	433.6	521.27	0.5564

Table 24: Temperature distribution for REMD experiment of 1YRF folding with GROMOS force field.

	Temperature, K	μ, kJ/mol	σ, kJ/mol	μ_{12}, kJ/mol	σ_{12}, kJ/mol	P_{12}
1	300.00	-90135	299.54			
2	303.15	-89583	300.90	552.1	424.58	0.2501
3	306.32	-89026	302.27	556.9	426.51	0.2500
4	309.52	-88465	303.66	561.5	428.46	0.2501
5	312.75	-87898	305.06	566.3	430.43	0.2501
6	316.01	-87327	306.47	571.1	432.41	0.2501
7	319.29	-86751	307.89	576.1	434.41	0.2500
8	322.58	-86175	309.31	581.1	436.43	0.2499
9	325.92	-85589	310.75	585.9	438.45	0.2500
10	329.29	-84998	312.21	591.0	440.50	0.2499
11	332.68	-84402	313.68	596.0	442.57	0.2500
12	336.11	-83801	315.16	600.9	444.66	0.2501
13	339.57	-83195	316.66	606.1	446.76	0.2501
14	343.05	-82583	318.16	611.4	448.89	0.2499
15	346.57	-81967	319.68	616.5	451.03	0.2500
16	350.11	-81345	321.22	621.8	453.19	0.2500
17	353.69	-80718	322.76	627.2	455.37	0.2500
18	357.29	-80085	324.33	632.5	457.56	0.2500
19	360.93	-79448	325.90	637.9	459.78	0.2500
20	364.59	-78804	327.48	643.4	462.01	0.2499
21	368.29	-78156	329.08	648.9	464.26	0.2500
22	372.02	-77501	330.70	654.4	466.54	0.2500
23	375.79	-76841	332.33	660.0	468.83	0.2500
24	379.58	-76176	333.97	665.7	471.14	0.2500
25	383.41	-75504	335.62	671.4	473.47	0.2500
26	387.27	-74826	337.30	677.0	475.82	0.2501
27	391.17	-74143	338.98	682.8	478.20	0.2501
28	395.10	-73454	340.68	688.7	480.59	0.2501
29	399.06	-72759	342.39	694.7	483.01	0.2500
30	400.00	-72593	342.80	165.7	484.51	0.8368

Table 25: Configuration properties of the REMD experiment for 1YRF protein MD simulation with GROMOS force field.

Variable	Value
P_{des}	0.25
Temperature range, K	300 - 400
Number of water molecules	2263
Number of protein atoms	383
Number of hydrogens in protein	~197
Number of constraints	~383
Number of vsites	~0
Number of DOF	~21133
Energy loss due to constraints	1.59 (kJ/mol K)

Table 26: Configuration properties of the REMD experiment for 1YRF protein MD simulation with AMBER, CHARMM, OPLS force fields.

Variable	Value
P_{des}	0.25
Temperature range, K	300 - 400
Number of water molecules	2350
Number of protein atoms	585
Number of hydrogens in protein	~300
Number of constraints	~585
Number of vsites	~0
Number of DOF	~22320
Energy loss due to constraints	2.43 (kJ/mol K)

Table 27: Temperature distribution for REMD experiment of 1YRF folding with AMBER, CHARMM, and OPLS force fields.

	Temperature, K	μ, kJ/mol	σ, kJ/mol	μ_{12}, kJ/mol	σ_{12}, kJ/mol	P_{12}
1	300.00	-97355	307.84			
2	303.07	-96790	309.20	566.1	436.31	0.2499
3	306.17	-96219	310.58	570.8	438.25	0.2500
4	309.30	-95644	311.97	575.4	440.21	0.2500
5	312.46	-95063	313.38	580.3	442.19	0.2500
6	315.64	-94478	314.79	585.1	444.18	0.2500
7	318.85	-93889	316.22	590.0	446.19	0.2500
8	322.09	-93292	317.66	594.8	448.21	0.2500
9	325.35	-92693	319.11	599.9	450.26	0.2500
10	328.63	-92089	320.57	604.9	452.32	0.2499
11	331.95	-91479	322.04	609.9	454.39	0.2500
12	335.28	-90867	323.52	614.9	456.49	0.2500
13	338.65	-90247	325.02	620.0	458.59	0.2501
14	342.05	-89621	326.53	625.2	460.72	0.2500
15	345.48	-88991	328.06	630.4	462.86	0.2500
16	348.93	-88355	329.59	635.8	465.03	0.2499
17	352.42	-87714	331.14	641.0	467.21	0.2500
18	355.93	-87068	332.70	646.4	469.41	0.2500
19	359.48	-86416	334.28	651.8	471.63	0.2500
20	363.05	-85759	335.87	657.1	473.87	0.2500
21	366.65	-85097	337.47	662.7	476.12	0.2500
22	370.29	-84428	339.09	668.1	478.40	0.2501
23	373.95	-83755	340.72	673.8	480.70	0.2500
24	377.64	-83075	342.36	679.4	483.01	0.2500
25	381.37	-82390	344.01	685.2	485.34	0.2500
26	385.13	-81699	345.68	690.7	487.69	0.2501
27	388.91	-81003	347.37	696.7	490.06	0.2500
28	392.73	-80301	349.07	702.6	492.46	0.2499
29	396.59	-79592	350.78	708.3	494.87	0.2500
30	400.00	-78964	352.30	627.9	497.15	0.3243

Table 28: Temperature distribution for REMD experiment of 1GB1 folding with AMBER, CHARMM, and OPLS force fields.

	Temperature, K	μ, kJ/mol	σ, kJ/mol	μ_{12}, kJ/mol	σ_{12}, kJ/mol	P_{12}
1	300.00	-139714	368.51			
2	302.57	-139038	369.88	677.1	522.12	0.2500
3	305.16	-138356	371.26	681.7	524.06	0.2500
4	307.76	-137670	372.64	686.3	526.02	0.2501
5	310.39	-136978	374.04	691.2	527.99	0.2500
6	313.03	-136283	375.45	696.1	529.97	0.2499
7	315.69	-135581	376.86	700.8	531.96	0.2501
8	318.37	-134876	378.29	705.6	533.97	0.2501
9	321.07	-134165	379.73	710.5	536.00	0.2501
10	323.78	-133449	381.17	715.7	538.04	0.2499
11	326.52	-132729	382.63	720.6	540.09	0.2500
12	329.27	-132003	384.09	725.4	542.15	0.2501
13	332.05	-131271	385.57	730.6	544.24	0.2500
14	334.84	-130536	387.06	735.7	546.33	0.2500
15	337.62	-129804	388.54	740.9	548.44	0.2500
16	340.45	-129058	390.04	746.0	550.54	0.2499
17	343.30	-128308	391.56	751.2	552.68	0.2500
18	346.17	-127551	393.09	756.3	554.83	0.2500
19	349.07	-126789	394.63	761.7	557.00	0.2499
20	351.98	-126022	396.18	766.9	559.18	0.2500
21	354.91	-125250	397.74	772.3	561.38	0.2500
22	357.86	-124472	399.31	777.7	563.60	0.2500
23	360.84	-123689	400.89	783.1	565.83	0.2500
24	363.83	-122900	402.49	788.5	568.08	0.2501
25	366.84	-122106	404.09	794.1	570.34	0.2499
26	369.88	-121306	405.71	799.5	572.61	0.2500
27	372.94	-120501	407.33	805.2	574.91	0.2500
28	376.01	-119692	408.97	810.9	577.22	0.2500
29	379.11	-118876	410.62	816.5	579.53	0.2500
30	382.22	-118054	412.28	822.2	581.88	0.2500
31	385.37	-117226	413.95	827.8	584.23	0.2500
32	388.53	-116393	415.63	833.7	586.61	0.2499
33	391.72	-115553	417.33	839.4	589.00	0.2501
34	394.93	-114708	419.04	845.3	591.40	0.2500
35	398.16	-113856	420.76	851.2	593.83	0.2500
36	400.00	-113372	421.74	484.3	595.74	0.5634

Table 29: Configuration properties of the REMD experiment for 1GB1 protein MD simulation with AMBER, CHARMM, OPLS force fields.

Variable	Value
P_{des}	0.25
Temperature range, K	300 - 400
Number of water molecules	3364
Number of protein atoms	855
Number of hydrogens in protein	~439
Number of constraints	~855
Number of vsites	~0
Number of DOF	~31986
Energy loss due to constraints	3.55 (kJ/mol K)

Table 30: Configuration properties of the REMD experiment for 1GB1 protein MD simulation with GROMOS force field.

Variable	Value
P_{des}	0.25
Temperature range, K	300 - 400
Number of water molecules	3391
Number of protein atoms	562
Number of hydrogens in protein	~289
Number of constraints	~562
Number of vsites	~0
Number of DOF	~31643
Energy loss due to constraints	2.34 (kJ/mol K)

Table 31: Temperature distribution for REMD experiment of 1GB1 folding with GROMOS force field.

	Temperature, K	μ, kJ/mol	σ, kJ/mol	μ_{12}, kJ/mol	σ_{12}, kJ/mol	P_{12}
1	300.00	-134825	366.53			
2	302.57	-134151	367.89	675.3	519.32	0.2499
3	305.15	-133471	369.26	680.0	521.24	0.2500
4	307.76	-132786	370.64	684.6	523.19	0.2500
5	310.38	-132096	372.03	689.5	525.15	0.2500
6	313.03	-131402	373.43	694.1	527.12	0.2501
7	315.69	-130703	374.84	698.9	529.10	0.2501
8	318.37	-129999	376.26	703.8	531.10	0.2500
9	321.07	-129290	377.68	708.7	533.12	0.2501
10	323.78	-128576	379.12	713.8	535.14	0.2499
11	326.52	-127858	380.57	718.7	537.18	0.2500
12	329.27	-127134	382.03	723.6	539.24	0.2501
13	332.05	-126404	383.50	728.7	541.31	0.2500
14	334.84	-125671	384.98	733.8	543.40	0.2500
15	337.62	-124940	386.45	739.0	545.49	0.2500
16	340.45	-124196	387.95	744.1	547.58	0.2499
17	343.30	-123448	389.45	749.2	549.71	0.2500
18	346.17	-122693	390.97	754.4	551.85	0.2500
19	349.06	-121934	392.51	759.8	554.01	0.2499
20	351.98	-121169	394.05	764.9	556.18	0.2501
21	354.91	-120398	395.60	770.3	558.36	0.2500
22	357.86	-119622	397.16	775.7	560.57	0.2500
23	360.84	-118841	398.74	781.1	562.79	0.2500
24	363.83	-118054	400.32	786.4	565.02	0.2501
25	366.84	-117262	401.92	792.1	567.27	0.2500
26	369.88	-116464	403.53	797.5	569.53	0.2501
27	372.94	-115661	405.14	803.1	571.82	0.2500
28	376.01	-114854	406.77	808.8	574.11	0.2500
29	379.11	-114040	408.41	814.4	576.42	0.2500
30	382.23	-113219	410.06	820.1	578.75	0.2500
31	385.37	-112393	411.73	825.7	581.09	0.2501
32	388.54	-111562	413.40	831.6	583.46	0.2500
33	391.70	-110732	415.07	837.5	585.83	0.2499
34	394.91	-109889	416.77	843.1	588.21	0.2500
35	398.14	-109040	418.48	849.1	590.62	0.2500
36	400.00	-108550	419.47	489.4	592.52	0.5577

Table 32: Ambient noise values computed during the GPA* start

	RMSD, Å	ANGL	AND, contancts	ANDH, contancts	XOR, contacts
1L2Y run 1					
AMBER	0.65	7.763	44	304	576
CHARMM	0.902	9.042	32	339.2	584
GROMOS	0.908	10.088	3.2	84.8	163.2
OPLS	0.913	9.603	21.6	324.8	555.2
GROMOS	0.908	10.088	3.2	84.8	163.2
1L2Y run 2					
AMBER	0.65	7.763	44.0	304	576
CHARMM	0.902	9.041	32.0	339.2	584
GROMOS	0.908	10.082	3.2	84.8	163.2
OPLS	0.913	9.603	21.6	324.8	555.2
1YRF					
AMBER	0.721	11.293	41.6	608	998.4
CHARMM	0.578	7.414	52	643.2	1056
GROMOS	0.728	13.251	1.6	185.6	284.8
OPLS	0.65	12.108	40	660.8	1056
GROMOS	0.728	13.13	1.6	192	304
1GB1					
AMBER	0.787	20.617	120.8	1380.8	2204.8
CHARMM	0.897	21.429	72.8	1331.2	2072
GROMOS	0.841	26.325	4.4	464	710
OPLS	0.926	27.924	146.4	1542.4	2534.4

Table 33: REMD results for 1L2Y, 1YRF, and 1GB1. Time column shows at what time in replica the lowest RMSD was spotted.

	1L2Y, run 1		1L2Y, run 2		1YRF		1GB1	
Replica	RMSD, Å	Time, ps	RMSD, Å	Time, ps	RMSD, Å	Time, ps	RMSD, Å	Time, ps
1	4.480	42 160	3.969	40 880	3.474	327 620	7.897	291 180
2	5.035	21 780	4.533	5520	2.932	241 340	7.789	222 800
3	4.288	23 100	4.931	42 240	3.884	301 380	7.863	263 260
4	5.950	23 460	4.120	14 200	4.484	120 840	7.765	167 380
5	5.485	27 840	3.228	30 940	4.156	120 920	7.770	177 840
6	5.113	30 020	2.914	48 220	4.490	121 280	7.757	177 340
7	5.630	53 280	4.609	52 060	3.097	245 500	7.794	198 860
8	3.559	11 580	4.688	27 300	4.194	120 960	7.678	178 360
9	3.632	43 160	4.803	44 960	4.620	120 640	7.788	163 220
10	4.559	12 120	5.855	53 580	4.777	286 460	7.847	177 660
11	4.365	11 680	5.453	2480	2.754	241 400	7.742	177 460
12	4.427	10 640	4.201	24 060	2.884	245 340	7.978	175 640
13	4.917	10 120	4.849	46 860	3.064	241 000	7.737	177 540
14	4.945	57 020	4.882	22 360	4.518	120 540	7.926	454 400
15	4.853	56 280	4.691	2760	3.126	244 960	7.653	177 920
16	5.058	44 860	4.154	57 580	2.615	241 420	7.767	202 060
17	5.032	51 400	5.384	36 560	3.707	330 500	7.813	263 620
18	4.827	45 260	5.275	36 160	3.499	332 320	7.712	263 760
19	4.731	25 580	4.521	60 560	3.040	244 940	7.630	263 720
20	5.058	60 560	4.685	41 560	3.342	241 220	7.139	18 340
21	4.572	17 000	5.352	32 300	4.655	120 580	7.674	177 420
22	4.694	22 280	3.829	56 880	2.989	245 040	7.748	178 380
23	5.123	55 620	4.960	22 160	2.883	245 240	7.949	262 720
24	5.222	34 140	5.228	60 600	3.595	195 540	7.538	202 080
25	4.973	60 600	5.257	1100	3.684	239 980	7.840	202 280
26	5.301	59 140	3.380	37 860	2.961	241 320	7.847	167 180
27	4.357	38 440	5.535	62 420	2.998	245 160	7.767	263 780
28	4.634	37 760	3.724	37 880	4.318	120 940	7.666	177 400
29	3.455	57 260	4.750	4180	2.850	241 360	7.769	263 800
30	4.766	16 080	4.170	40 720	2.953	245 740	8.188	223 240
31	5.563	39 460	5.243	15 900	-	-	7.829	263 840
32	5.420	3300	3.886	44 140	-	-	7.828	263 140
33	-	-	-	-	-	-	7.944	296 840
34	-	-	-	-	-	-	7.746	178 400
35	-	-	-	-	-	-	7.604	263 820
36	-	-	-	-	-	-	7.794	202 040
Min	3.455	57 260	2.914	48 220	2.615	241 420	7.139	18 340

Table 34: SMD full results, all simulations had the same duration 2 ns.

Force	1L2Y		1YRF		1GB1	
KJ/mol	AARMSD	BBRMSD	AARMSD	BBRMSD	AARMSD	BBRMSD
1	5.931	3.923	7.659	6.914	7.694	6.167
10	2.235	1.650	2.994	1.946	5.033	3.564
20	2.648	1.842	3.212	2.708	3.855	2.240
30	4.556	4.274	3.350	2.720	2.271	1.454
40	1.089	0.606	1.425	0.961	2.005	1.186
50	4.647	4.430	0.819	0.505	1.680	1.156
60	4.754	4.380	1.507	1.217	1.669	1.034
70	1.254	0.760	1.123	0.652	1.710	0.802
80	4.394	4.094	0.629	0.254	1.495	0.942
90	1.044	0.572	0.808	0.396	1.591	0.761

Table 35: GPA* runtime analysis of the metrics' progress. 1L2Y with AMBER force field. 1st run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	15 591	45.52 %	110	66.67 %	7.06
ANGL	10	20 %	7871	22.98 %	51	30.91 %	6.48
ANDH	5	10 %	2817	8.22 %	3	1.82 %	1.06
AND	5	10 %	2760	8.06 %	0	0.00 %	0.00
XOR	10	20 %	5213	15.22 %	1	0.61 %	0.19
Total	50	100 %	34 252	100.00 %	165	100.00 %	4.82×5

Table 36: GPA* runtime analysis of the metrics' progress. 1L2Y with AMBER force field. 2nd run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	15 116	44.26 %	98	60.12 %	6.48
ANGL	10	20 %	8559	25.06 %	61	37.42 %	7.13
ANDH	5	10 %	2685	7.86 %	2	1.23 %	0.74
AND	5	10 %	2640	7.73 %	0	0.00 %	0.00
XOR	10	20 %	5153	15.09 %	2	1.23 %	0.39
Total	50	100 %	34 153	100.00 %	163	100.00 %	4.77×5

Table 37: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with AMBER force field 1st run and 2nd run.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	30 707	44.89 %	208	63.41 %	6.77
ANGL	10	20 %	16 430	24.02 %	112	34.15 %	6.82
ANDH	5	10 %	5502	8.04 %	5	1.52 %	0.91
AND	5	10 %	5400	7.89 %	0	0.00 %	0.00
XOR	10	20 %	10 366	15.15 %	3	0.91 %	0.29
Total	50	100 %	68 405	100.00 %	328	100.00 %	4.79×5

Table 38: GPA* runtime analysis of the metrics' progress. 1L2Y with CHARMM force field. 1st run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	13 352	39.73 %	100	57.47 %	7.49
ANGL	10	20 %	7985	23.76 %	52	29.89 %	6.51
ANDH	5	10 %	3325	9.89 %	11	6.32 %	3.31
AND	5	10 %	3177	9.45 %	3	1.72 %	0.94
XOR	10	20 %	5772	17.17 %	8	4.60 %	1.39
Total	50	100 %	33 611	100.00 %	174	100.00 %	5.18×5

Table 39: GPA* runtime analysis of the metrics' progress. 1L2Y with CHARMM force field. 2nd run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	12 484	41.23 %	94	78.99 %	7.53
ANGL	10	20 %	6175	20.39 %	15	12.61 %	2.43
ANDH	5	10 %	3082	10.18 %	4	3.36 %	1.30
AND	5	10 %	3016	9.96 %	3	2.52 %	0.99
XOR	10	20 %	5521	18.23 %	3	2.52 %	0.54
Total	50	100 %	30 278	100.00 %	119	100.00 %	3.93×5

Table 40: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with CHARMM force field 1st run and 2nd run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	25 836	40.44 %	194	66.21 %	7.51
ANGL	10	20 %	14 160	22.16 %	67	22.87 %	4.73
ANDH	5	10 %	6407	10.03 %	15	5.12 %	2.34
AND	5	10 %	6193	9.69 %	6	2.05 %	0.97
XOR	10	20 %	11 293	17.68 %	11	3.75 %	0.97
Total	50	100 %	63 889	100.00 %	293	100.00 %	4.59×5

Table 41: GPA* runtime analysis of the metrics' progress. 1L2Y with GROMOS force field. 1st run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	14 189	42.25 %	82	57.34 %	5.78
ANGL	10	20 %	7711	22.96 %	47	32.87 %	6.10
ANDH	5	10 %	2907	8.66 %	4	2.80 %	1.38
AND	5	10 %	2957	8.81 %	4	2.80 %	1.35
XOR	10	20 %	5819	17.33 %	6	4.20 %	1.03
Total	50	100 %	33 583	100.00 %	143	100.00 %	4.26×5

Table 42: GPA* runtime analysis of the metrics' progress. 1L2Y with GROMOS force field. 2nd run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	14 810	43.56 %	115	75.66 %	7.77
ANGL	10	20 %	7163	21.07 %	29	19.08 %	4.05
ANDH	5	10 %	3164	9.31 %	3	1.97 %	0.95
AND	5	10 %	3030	8.91 %	1	0.66 %	0.33
XOR	10	20 %	5834	17.16 %	4	2.63 %	0.69
Total	50	100 %	34 001	100.00 %	152	100.00 %	4.47×5

Table 43: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with GROMOS force field 1st run and 2nd run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	28 999	42.91 %	197	66.78 %	6.79
ANGL	10	20 %	14 874	22.01 %	76	25.76 %	5.11
ANDH	5	10 %	6071	8.98 %	7	2.37 %	1.15
AND	5	10 %	5987	8.86 %	5	1.69 %	0.84
XOR	10	20 %	11 653	17.24 %	10	3.39 %	0.86
Total	50	100 %	67 584	100.00 %	295	100.00 %	4.36×5

Table 44: GPA* runtime analysis of the metrics' progress. 1L2Y with OPLS force field. 1st run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	13 372	40.02 %	95	52.78 %	7.10
ANGL	10	20 %	7988	23.91 %	50	27.78 %	6.26
ANDH	5	10 %	3418	10.23 %	19	10.56 %	5.56
AND	5	10 %	2787	8.34 %	2	1.11 %	0.72
XOR	10	20 %	5847	17.50 %	14	7.78 %	2.39
Total	50	100 %	33 412	100.00 %	180	100.00 %	5.39×5

Table 45: GPA* runtime analysis of the metrics' progress. 1L2Y with OPLS force field. 2nd run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	14 074	40.88 %	123	62.76 %	8.74
ANGL	10	20 %	8386	24.36 %	52	26.53 %	6.20
ANDH	5	10 %	3153	9.16 %	6	3.06 %	1.90
AND	5	10 %	3324	9.65 %	8	4.08 %	2.41
XOR	10	20 %	5494	15.96 %	7	3.57 %	1.27
Total	50	100 %	34 431	100.00 %	196	100.00 %	5.69×5

Table 46: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with OPLS force field 1st run and 2nd run

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	27 446	40.46 %	218	57.98 %	7.94
ANGL	10	20 %	16 374	24.14 %	102	27.13 %	6.23
ANDH	5	10 %	6571	9.69 %	25	6.65 %	3.80
AND	5	10 %	6111	9.01 %	10	2.66 %	1.64
XOR	10	20 %	11 341	16.72 %	21	5.59 %	1.85
Total	50	100 %	67 843	100.00 %	376	100.00 %	5.54×5

Table 47: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with AMBER (1st run), CHARMM (1st run), GROMOS (1st run), and OPLS (1st run) force fields.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	56 504	41.90 %	387	58.46 %	6.85
ANGL	10	20 %	31 555	23.40 %	200	30.21 %	6.34
ANDH	5	10 %	12 467	9.24 %	37	5.59 %	2.97
AND	5	10 %	11 681	8.66 %	9	1.36 %	0.77
XOR	10	20 %	22 651	16.80 %	29	4.38 %	1.28
Total	50	100 %	134 858	100.00 %	662	100.00 %	4.91×5

Table 48: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with AMBER (2nd run), CHARMM (2nd run), GROMOS (2nd run), and OPLS (2nd run) force fields.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	56 484	42.51 %	430	68.25 %	7.61
ANGL	10	20 %	30 283	22.79 %	157	24.92 %	5.18
ANDH	5	10 %	12 084	9.10 %	15	2.38 %	1.24
AND	5	10 %	12 010	9.04 %	12	1.90 %	1.00
XOR	10	20 %	22 002	16.56 %	16	2.54 %	0.73
Total	50	100 %	132 863	100.00 %	630	100.00 %	4.74×5

Table 49: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with AMBER (1st and 2nd run), CHARMM (1st and 2nd run), GROMOS (1st and 2nd run), and OPLS (1st and 2nd run) force fields.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	112 988	42.20 %	817	63.24 %	7.23
ANGL	10	20 %	61 838	23.10 %	357	27.63 %	5.77
ANDH	5	10 %	24 551	9.17 %	52	4.02 %	2.12
AND	5	10 %	23 691	8.85 %	21	1.63 %	0.89
XOR	10	20 %	44 653	16.68 %	45	3.48 %	1.01
Total	50	100 %	267 721	100.00 %	1292	100.00 %	4.83×5

Table 50: GPA* runtime analysis of the metrics' progress. 1L2Y with AMBER force field. 1st run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	3898	24.34 %	27.5	48.67 %	7.06
ANGL	5	20 %	3936	24.57 %	25.5	45.13 %	6.48
ANDH	5	20 %	2817	17.59 %	3.0	5.31 %	1.06
AND	5	20 %	2760	17.23 %	0.0	0.00 %	0.00
XOR	5	20 %	2606	16.27 %	0.5	0.88 %	0.19
Total	25	100 %	16 017	100.00 %	56.5	100.00 %	3.53×5

Table 51: GPA* runtime analysis of the metrics' progress. 1L2Y with AMBER force field. 2nd run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	3779	23.68 %	24.5	42.24 %	6.48
ANGL	5	20 %	4280	26.81 %	30.5	52.59 %	7.13
ANDH	5	20 %	2685	16.82 %	2.0	3.45 %	0.74
AND	5	20 %	2640	16.54 %	0.0	0.00 %	0.00
XOR	5	20 %	2576	16.14 %	1.0	1.72 %	0.39
Total	25	100 %	15 960	100.00 %	58.0	100.00 %	3.63×5

Table 52: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with AMBER force field 1st run and 2nd run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	7677	24.01 %	52.0	45.41 %	6.77
ANGL	5	20 %	8215	25.69 %	56.0	48.91 %	6.82
ANDH	5	20 %	5502	17.21 %	5.0	4.37 %	0.91
AND	5	20 %	5400	16.89 %	0.0	0.00 %	0.00
XOR	5	20 %	5183	16.21 %	1.5	1.31 %	0.29
Total	25	100 %	31 977	100.00 %	114.5	100.00 %	3.58×5

Table 53: GPA* runtime analysis of the metrics' progress. 1L2Y with CHARMM force field. 1st run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	3338	19.97 %	25.0	36.23 %	7.49
ANGL	5	20 %	3992	23.88 %	26.0	37.68 %	6.51
ANDH	5	20 %	3325	19.89 %	11.0	15.94 %	3.31
AND	5	20 %	3177	19.00 %	3.0	4.35 %	0.94
XOR	5	20 %	2886	17.26 %	4.0	5.80 %	1.39
Total	25	100 %	16718	100.00 %	69.0	100.00 %	4.13×5

Table 54: GPA* runtime analysis of the metrics' progress. 1L2Y with CHARMM force field. 2nd run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	3121	20.71 %	23.5	59.49 %	7.53
ANGL	5	20 %	3088	20.49 %	7.5	18.99 %	2.43
ANDH	5	20 %	3082	20.46 %	4.0	10.13 %	1.30
AND	5	20 %	3016	20.02 %	3.0	7.59 %	0.99
XOR	5	20 %	2760	18.32 %	1.5	3.80 %	0.54
Total	25	100 %	15067	100.00 %	39.5	100.00 %	2.62×5

Table 55: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with CHARMM force field 1st run and 2nd run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	6459	20.32 %	48.5	44.70 %	7.51
ANGL	5	20 %	7080	22.27 %	33.5	30.88 %	4.73
ANDH	5	20 %	6407	20.16 %	15.0	13.82 %	2.34
AND	5	20 %	6193	19.48 %	6.0	5.53 %	0.97
XOR	5	20 %	5646	17.76 %	5.5	5.07 %	0.97
Total	25	100 %	31 786	100.00 %	108.5	100.00 %	3.41×5

Table 56: GPA* runtime analysis of the metrics' progress. 1L2Y with GROMOS force field. 1st run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	3547	21.93 %	20.5	37.27 %	5.78
ANGL	5	20 %	3856	23.83 %	23.5	42.73 %	6.10
ANDH	5	20 %	2907	17.97 %	4.0	7.27 %	1.38
AND	5	20 %	2957	18.28 %	4.0	7.27 %	1.35
XOR	5	20 %	2910	17.99 %	3.0	5.45 %	1.03
Total	25	100 %	16 176	100.00 %	55.0	100.00 %	3.40×5

Table 57: GPA* runtime analysis of the metrics' progress. 1L2Y with GROMOS force field. 2nd run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	3702	22.58 %	28.75	58.38 %	7.77
ANGL	5	20 %	3582	21.85 %	14.5	29.44 %	4.05
ANDH	5	20 %	3164	19.30 %	3.0	6.09 %	0.95
AND	5	20 %	3030	18.48 %	1.0	2.03 %	0.33
XOR	5	20 %	2917	17.79 %	2.0	4.06 %	0.69
Total	25	100 %	16395	100.00 %	49.25	100.00 %	3.00×5

Table 58: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with GROMOS force field 1st run and 2nd run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	7250	22.26 %	49.25	47.24 %	6.79
ANGL	5	20 %	7437	22.83 %	38.0	36.45 %	5.11
ANDH	5	20 %	6071	18.64 %	7.0	6.71 %	1.15
AND	5	20 %	5987	18.38 %	5.0	4.80 %	0.84
XOR	5	20 %	5826	17.89 %	5.0	4.80 %	0.86
Total	25	100 %	32571	100.00 %	104.25	100.00 %	3.20×5

Table 59: GPA* runtime analysis of the metrics' progress. 1L2Y with OPLS force field. 1st run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	3343	20.30 %	23.75	30.94 %	7.10
ANGL	5	20 %	3994	24.26 %	25.0	32.57 %	6.26
ANDH	5	20 %	3418	20.76 %	19.0	24.76 %	5.56
AND	5	20 %	2787	16.93 %	2.0	2.61 %	0.72
XOR	5	20 %	2924	17.76 %	7.0	9.12 %	2.39
Total	25	100 %	16466	100.00 %	76.75	100.00 %	4.66×5

Table 60: GPA* runtime analysis of the metrics' progress. 1L2Y with OPLS force field. 2nd run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	3518	20.78 %	30.75	41.41 %	8.74
ANGL	5	20 %	4193	24.76 %	26.0	35.02 %	6.20
ANDH	5	20 %	3153	18.62 %	6.0	8.08 %	1.90
AND	5	20 %	3324	19.63 %	8.0	10.77 %	2.41
XOR	5	20 %	2747	16.22 %	3.5	4.71 %	1.27
Total	25	100 %	16936	100 %	74.25	100.00 %	4.38×5

Table 61: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with OPLS force field 1st run and 2nd run. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	6862	20.54 %	54.5	36.09 %	7.94
ANGL	5	20 %	8187	24.51 %	51.0	33.77 %	6.23
ANDH	5	20 %	6571	19.67 %	25.0	16.56 %	3.80
AND	5	20 %	6111	18.30 %	10.0	6.62 %	1.64
XOR	5	20 %	5670	16.98 %	10.5	6.95 %	1.85
Total	25	100 %	33 401	100.00 %	151.0	100.00 %	4.52×5

Table 62: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with AMBER (1st run), CHARMM (1st run), GROMOS (1st run), and OPLS (1st run) force fields. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	14 126	21.61 %	96.75	37.61 %	6.85
ANGL	5	20 %	15 778	24.13 %	100.0	38.87 %	6.34
ANDH	5	20 %	12 467	19.07 %	37.0	14.38 %	2.97
AND	5	20 %	11 681	17.87 %	9.0	3.50 %	0.77
XOR	5	20 %	11 326	17.32 %	14.5	5.64 %	1.28
Total	25	100 %	65 377	100.00 %	257.25	100.00 %	3.93×5

Table 63: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with AMBER (2nd run), CHARMM (2nd run), GROMOS (2nd run), and OPLS (2nd run) force fields. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	14 121	21.94 %	107.5	48.64 %	7.61
ANGL	5	20 %	15 142	23.53 %	78.5	35.52 %	5.18
ANDH	5	20 %	12 084	18.78 %	15.0	6.79 %	1.24
AND	5	20 %	12 010	18.66 %	12.0	5.43 %	1.00
XOR	5	20 %	11 001	17.09 %	8.0	3.62 %	0.73
Total	25	100 %	64 358	100.00 %	221.0	100.00 %	3.43×5

Table 64: GPA* runtime analysis of the metrics' progress. Summary of 1L2Y with AMBER (1st and 2nd run), CHARMM (1st and 2nd run), GROMOS (1st and 2nd run), and OPLS (1st and 2nd run) force fields. Normalized.

Metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	28 247	21.77 %	204.25	42.71 %	7.23
ANGL	5	20 %	30 919	23.83 %	178.5	37.32 %	5.77
ANDH	5	20 %	24 551	18.92 %	52.0	10.87 %	2.12
AND	5	20 %	23 691	18.26 %	21.0	4.39 %	0.89
XOR	5	20 %	22 326	17.21 %	22.5	4.70 %	1.01
Total	25	100 %	129 734	100.00 %	478.25	100.00 %	3.69×5

Table 65: GPA* runtime analysis of the metrics' progress. 1YRF with AMBER force field.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	65 479	38.39 %	173	39.59 %	2.64
ANGL	10	20 %	36 191	21.22 %	144	32.95 %	3.98
ANDH	5	10 %	17 951	10.53 %	60	13.73 %	3.34
AND	5	10 %	17 492	10.26 %	23	5.26 %	1.31
XOR	10	20 %	33 440	19.61 %	37	8.47 %	1.11
Total	50	100 %	170 553	100.00 %	437	100.00 %	2.56×5

Table 66: GPA* runtime analysis of the metrics' progress. 1YRF with CHARMM force field.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	53 902	37.79 %	129	40.82 %	2.39
ANGL	10	20 %	31 053	21.77 %	119	37.66 %	3.83
ANDH	5	10 %	15 217	10.67 %	34	10.76 %	2.23
AND	5	10 %	14 774	10.36 %	13	4.11 %	0.88
XOR	10	20 %	27 677	19.41 %	21	6.65 %	0.76
Total	50	100 %	142 623	100.00 %	316	100.00 %	2.22×5

Table 67: GPA* runtime analysis of the metrics' progress. 1YRF with GROMOS force field.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	66 346	38.19 %	120	40.96 %	1.81
ANGL	10	20 %	37 786	21.75 %	139	47.44 %	3.68
ANDH	5	10 %	17 469	10.06 %	2	0.68 %	0.11
AND	5	10 %	17 747	10.22 %	6	2.05 %	0.34
XOR	10	20 %	34 371	19.79 %	26	8.87 %	0.76
Total	50	100 %	173 719	100.00 %	293	100.00 %	1.69×5

Table 68: GPA* runtime analysis of the metrics' progress. 1YRF with OPLS force field.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	68 343	41.13 %	230	55.69 %	3.37
ANGL	10	20 %	33 303	20.04 %	95	23.00 %	2.85
ANDH	5	10 %	17 174	10.34 %	38	9.20 %	2.21
AND	5	10 %	15 866	9.55 %	9	2.18 %	0.57
XOR	10	20 %	31 460	18.94 %	41	9.93 %	1.30
Total	50	100 %	166 146	100.00 %	413	100.00 %	2.49×5

Table 69: GPA* runtime analysis of the metrics' progress. Summary of 1YRF with AMBER, CHARMM, GROMOS, and OPLS force fields.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	254 070	38.91 %	652	44.69 %	2.57
ANGL	10	20 %	138 333	21.18 %	497	34.06 %	3.59
ANDH	5	10 %	67 811	10.38 %	134	9.18 %	1.98
AND	5	10 %	65 879	10.09 %	51	3.50 %	0.77
XOR	10	20 %	126 948	19.44 %	125	8.57 %	0.98
Total	50	100 %	653 041	100.00 %	1459	100.00 %	2.23×5

Table 70: GPA* runtime analysis of the metrics' progress. 1YRF with AMBER force field. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	16 370	18.90 %	43.25	19.95 %	2.64
ANGL	5	20 %	18 096	20.89 %	72.0	33.22 %	3.98
ANDH	5	20 %	17 951	20.72 %	60.0	27.68 %	3.34
AND	5	20 %	17 492	20.19 %	23.0	10.61 %	1.31
XOR	5	20 %	16 720	19.30 %	18.5	8.54 %	1.11
Total	25	100 %	86 628	100.00 %	216.75	100.00 %	2.50×5

Table 71: GPA* runtime analysis of the metrics' progress. 1YRF with CHARMM force field. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	13 476	18.50 %	32.25	21.61 %	2.39
ANGL	5	20 %	15 526	21.32 %	59.5	39.87 %	3.83
ANDH	5	20 %	15 217	20.89 %	34.0	22.78 %	2.23
AND	5	20 %	14 774	20.29 %	13.0	8.71 %	0.88
XOR	5	20 %	13 838	19.00 %	10.5	7.04 %	0.76
Total	25	100 %	72 832	100.00 %	149.25	100.00 %	2.05×5

Table 72: GPA* runtime analysis of the metrics' progress. 1YRF with GROMOS force field. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	16 586	18.87 %	30.0	24.90 %	1.81
ANGL	5	20 %	18 893	21.50 %	69.5	57.68 %	3.68
ANDH	5	20 %	17 469	19.88 %	2.0	1.66 %	0.11
AND	5	20 %	17 747	20.19 %	6.0	4.98 %	0.34
XOR	5	20 %	17 186	19.56 %	13.0	10.79 %	0.76
Total	25	100 %	87 881	100.00 %	120.5	100.00 %	1.37×5

Table 73: GPA* runtime analysis of the metrics' progress. 1YRF with OPLS force field. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	17 086	20.71 %	57.5	33.33 %	3.37
ANGL	5	20 %	16 652	20.18 %	47.5	27.54 %	2.85
ANDH	5	20 %	17 174	20.82 %	38.0	22.03 %	2.21
AND	5	20 %	15 866	19.23 %	9.0	5.22 %	0.57
XOR	5	20 %	15 730	19.06 %	20.5	11.88 %	1.30
Total	25	100 %	82 507	100.00 %	172.5	100.00 %	2.09×5

Table 74: GPA* runtime analysis of the metrics' progress. Summary of 1YRF with AMBER, CHARMM, GROMOS, and OPLS force fields. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	63 518	19.26 %	163.0	24.73 %	2.57
ANGL	5	20 %	69 166	20.97 %	248.5	37.71 %	3.59
ANDH	5	20 %	67 811	20.56 %	134.0	20.33 %	1.98
AND	5	20 %	65 879	19.97 %	51.0	7.74 %	0.77
XOR	5	20 %	63 474	19.24 %	62.5	9.48 %	0.98
Total	25	100 %	329 848	100.00 %	659.0	100.00 %	2.00×5

Table 75: GPA* runtime analysis of the metrics' progress. 1GB1 with AMBER force field.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	79 182	39.80 %	296	35.75 %	3.74
ANGL	10	20 %	44 115	22.17 %	291	35.14 %	6.60
ANDH	5	10 %	21 099	10.60 %	137	16.55 %	6.49
AND	5	10 %	19 319	9.71 %	48	5.80 %	2.48
XOR	10	20 %	35 251	17.72 %	56	6.76 %	1.59
Total	50	100 %	198 966	100.00 %	828	100.00 %	4.16

Table 76: GPA* runtime analysis of the metrics' progress. 1GB1 with CHARMM force field.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	115 520	38.58 %	222	28.14 %	1.92
ANGL	10	20 %	65 134	21.75 %	292	37.01 %	4.48
ANDH	5	10 %	32 953	11.00 %	178	22.56 %	5.40
AND	5	10 %	30 522	10.19 %	47	5.96 %	1.54
XOR	10	20 %	55 318	18.47 %	50	6.34 %	0.90
Total	50	100 %	299 447	100.00 %	789	100.00 %	2.63

Table 77: GPA* runtime analysis of the metrics' progress. 1GB1 with GROMOS force field.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	99 730	39.69 %	238	57.35 %	2.39
ANGL	10	20 %	55 377	22.04 %	177	42.65 %	3.20
ANDH	5	10 %	25 080	9.98 %	0	0.00 %	0.00
AND	5	10 %	25 080	9.98 %	0	0.00 %	0.00
XOR	10	20 %	45 980	18.30 %	0	0.00 %	0.00
Total	50	100 %	251 247	100.00 %	415	100.00 %	1.65

Table 78: GPA* runtime analysis of the metrics' progress. 1GB1 with OPLS force field.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	83 992	39.95 %	176	51.01 %	2.10
ANGL	10	20 %	45 220	21.51 %	161	46.67 %	3.56
ANDH	5	10 %	21 269	10.12 %	5	1.45 %	0.24
AND	5	10 %	21 000	9.99 %	0	0.00 %	0.00
XOR	10	20 %	38 773	18.44 %	3	0.87 %	0.08
Total	50	100 %	210 254	100.00 %	345	100.00 %	1.64

Table 79: GPA* runtime analysis of the metrics' progress. Summary of 1GB1 with AMBER, CHARMM, GROMOS, and OPLS force fields.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	20	40 %	378 424	39.42 %	932	39.21 %	2.46
ANGL	10	20 %	209 846	21.86 %	921	38.75 %	4.39
ANDH	5	10 %	100 401	10.46 %	320	13.46 %	3.19
AND	5	10 %	95 921	9.99 %	95	4.00 %	0.99
XOR	10	20 %	175 322	18.26 %	109	4.59 %	0.62
Total	50	100 %	959 914	100.00 %	2377	100.00 %	2.48

Table 80: GPA* runtime analysis of the metrics' progress. 1GB1 with AMBER force field. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	19 796	19.82 %	74.0	17.11 %	3.74
ANGL	5	20 %	22 058	22.08 %	145.5	33.64 %	6.60
ANDH	5	20 %	21 099	21.12 %	137.0	31.68 %	6.49
AND	5	20 %	19 319	19.34 %	48.0	11.10 %	2.48
XOR	5	20 %	17 626	17.64 %	28.0	6.47 %	1.59
Total	25	100 %	99 896	100 %	432.5	100 %	4.33

Table 81: GPA* runtime analysis of the metrics' progress. 1GB1 with CHARMM force field. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	28 880	18.93 %	55.5	12.29 %	1.92
ANGL	5	20 %	32 567	21.34 %	146.0	32.34 %	4.48
ANDH	5	20 %	32 953	21.60 %	178.0	39.42 %	5.40
AND	5	20 %	30 522	20.00 %	47.0	10.41 %	1.54
XOR	5	20 %	27 659	18.13 %	25.0	5.54 %	0.90
Total	25	100 %	152 581	100 %	451.5	100 %	2.96

Table 82: GPA* runtime analysis of the metrics' progress. 1GB1 with GROMOS force field. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	24 932	19.82 %	59.5	40.20 %	2.39
ANGL	5	20 %	27 688	22.02 %	88.5	59.80 %	3.20
ANDH	5	20 %	25 080	19.94 %	0.0	0.00 %	0.00
AND	5	20 %	25 080	19.94 %	0.0	0.00 %	0.00
XOR	5	20 %	22 990	18.28 %	0.0	0.00 %	0.00
Total	25	100 %	125 771	100 %	148.0	100 %	1.18

Table 83: GPA* runtime analysis of the metrics' progress. 1GB1 with OPLS force field. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	20 998	19.95 %	44.0	33.59 %	2.10
ANGL	5	20 %	22 610	21.48 %	80.5	61.45 %	3.56
ANDH	5	20 %	21 269	20.21 %	5.0	3.82 %	0.24
AND	5	20 %	21 000	19.95 %	0.0	0.00 %	0.00
XOR	5	20 %	19 386	18.42 %	1.5	1.15 %	0.08
Total	25	100 %	105 264	100 %	131.0	100 %	1.24

Table 84: GPA* runtime analysis of the metrics' progress. Summary of 1GB1 with AMBER, CHARMM, GROMOS, and OPLS force fields. Normalized.

metric	Allowed fails	Percent allowed	Metric total steps	Percent steps	Promotions per metric	Percent of promotions	Promotions per 1000 steps
RMSD	5	20 %	94 606	19.57 %	233.0	20.03 %	2.46
ANGL	5	20 %	104 923	21.70 %	460.5	39.60 %	4.39
ANDH	5	20 %	100 401	20.76 %	320.0	27.52 %	3.19
AND	5	20 %	95 921	19.84 %	95.0	8.17 %	0.99
XOR	5	20 %	87 661	18.13 %	54.5	4.69 %	0.62
Total	25	100 %	483 512	100 %	1163.0	100 %	2.41

Table 85: Correlation coefficients among metrics and potential energy for the first simulation of 1L2Y protein with AMBER force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.81	0.79	0.80	0.70	0.70
ANGL	0.79	1.00	0.87	0.90	0.84	0.72
ANDH	0.78	0.86	1.00	0.99	0.94	0.67
AND	0.79	0.89	0.99	1.00	0.95	0.69
XOR	0.70	0.82	0.94	0.95	1.00	0.55

Table 86: Correlation coefficients among metrics and potential energy for the second simulation of 1L2Y protein with AMBER force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.79	0.78	0.80	0.74	0.74
ANGL	0.75	1.00	0.91	0.93	0.88	0.60
ANDH	0.75	0.87	1.00	0.99	0.96	0.65
AND	0.77	0.90	0.99	1.00	0.97	0.68
XOR	0.72	0.86	0.96	0.97	1.00	0.60

Table 87: Correlation coefficients among metrics and potential energy for the first simulation of 1L2Y protein with CHARMM force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.55	0.50	0.52	0.38	0.61
ANGL	0.49	1.00	0.91	0.94	0.88	0.82
ANDH	0.19	0.90	1.00	1.00	0.98	0.69
AND	0.20	0.92	1.00	1.00	0.98	0.71
XOR	0.15	0.91	0.98	0.98	1.00	0.66

Table 88: Correlation coefficients among metrics and potential energy for the second simulation of 1L2Y protein with CHARMM force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.85	0.78	0.79	0.75	0.77
ANGL	0.78	1.00	0.92	0.93	0.90	0.92
ANDH	0.70	0.92	1.00	1.00	0.98	0.85
AND	0.71	0.93	1.00	1.00	0.98	0.86
XOR	0.67	0.90	0.98	0.98	1.00	0.79

Table 89: Correlation coefficients among metrics and potential energy for the first simulation of 1L2Y protein with GROMOS force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.78	0.69	0.74	0.66	0.66
ANGL	0.74	1.00	0.91	0.93	0.89	0.52
ANDH	0.66	0.89	1.00	0.97	0.89	0.63
AND	0.70	0.93	0.97	1.00	0.94	0.60
XOR	0.66	0.89	0.91	0.94	1.00	0.45

Table 90: Correlation coefficients among metrics and potential energy for the second simulation of 1L2Y protein with GROMOS force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.85	0.80	0.80	0.65	0.69
ANGL	0.77	1.00	0.93	0.94	0.85	0.73
ANDH	0.73	0.92	1.00	0.98	0.89	0.69
AND	0.72	0.93	0.98	1.00	0.91	0.76
XOR	0.61	0.83	0.88	0.90	1.00	0.54

Table 91: Correlation coefficients among metrics and potential energy for the first simulation of 1L2Y protein with OPLS force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.49	0.18	0.18	−0.27	0.57
ANGL	−0.06	1.00	0.29	0.47	0.24	0.42
ANDH	0.26	0.58	1.00	0.98	0.90	0.39
AND	0.14	0.62	0.98	1.00	0.89	0.37
XOR	0.20	0.55	0.90	0.90	1.00	0.22

Table 92: Correlation coefficients among metrics and potential energy for the second simulation of 1L2Y protein with OPLS force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.87	0.82	0.84	0.80	0.64
ANGL	0.84	1.00	0.93	0.95	0.92	0.41
ANDH	0.80	0.93	1.00	1.00	0.98	0.37
AND	0.82	0.95	1.00	1.00	0.98	0.39
XOR	0.76	0.92	0.98	0.98	1.00	0.28

Table 93: Correlation coefficients among metrics and potential energy for simulation of 1YRF protein with AMBER force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.94	0.89	0.90	0.85	0.80
ANGL	0.57	1.00	0.66	0.72	0.60	0.81
ANDH	0.01	0.95	1.00	1.00	0.99	0.82
AND	0.01	0.95	1.00	1.00	0.99	0.88
XOR	0.04	0.94	0.99	0.99	1.00	0.82

Table 94: Correlation coefficients among metrics and potential energy for simulation of 1YRF protein with CHARMM force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.84	0.83	0.84	0.74	0.83
ANGL	0.10	1.00	0.89	0.92	0.90	0.71
ANDH	0.29	0.93	1.00	1.00	0.97	0.86
AND	0.27	0.94	1.00	1.00	0.98	0.86
XOR	0.34	0.93	0.97	0.98	1.00	0.82

Table 95: Correlation coefficients among metrics and potential energy for simulation of 1YRF protein with GROMOS force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.40	0.35	0.49	0.04	0.56
ANGL	0.68	1.00	0.87	0.89	0.84	0.82
ANDH	0.75	0.86	1.00	0.97	0.90	0.75
AND	0.79	0.88	0.98	1.00	0.93	0.75
XOR	0.62	0.81	0.90	0.93	1.00	0.56

Table 96: Correlation coefficients among metrics and potential energy for simulation of 1YRF protein with OPLS force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.86	0.85	0.86	0.82	0.70
ANGL	0.85	1.00	0.95	0.96	0.95	0.66
ANDH	−0.35	0.93	1.00	1.00	0.98	0.79
AND	0.86	0.96	1.00	1.00	0.98	0.68
XOR	−0.54	0.94	0.99	0.99	1.00	0.71

Table 97: Correlation coefficients among metrics and potential energy for simulation of 1GB1 protein with AMBER force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.68	0.42	0.53	−0.16	0.75
ANGL	0.37	1.00	0.88	0.91	0.73	0.76
ANDH	0.50	0.69	1.00	1.00	0.98	0.77
AND	0.53	0.72	1.00	1.00	0.98	0.79
XOR	0.43	0.66	0.98	0.97	1.00	0.72

Table 98: Correlation coefficients among metrics and potential energy for simulation of 1GB1 protein with CHARMM force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.37	0.12	0.17	−0.52	0.66
ANGL	−0.27	1.00	0.80	0.87	0.80	0.42
ANDH	0.26	0.67	1.00	0.99	0.96	0.50
AND	0.14	0.71	0.99	1.00	0.97	0.63
XOR	0.15	0.70	0.96	0.97	1.00	0.54

Table 99: Correlation coefficients among metrics and potential energy for simulation of 1GB1 protein with GROMOS force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.83	0.77	0.77	0.54	−0.22
ANGL	0.33	1.00	0.94	0.94	0.86	−0.62
ANDH	0.30	0.94	1.00	0.97	0.86	−0.53
AND	0.34	0.94	0.97	1.00	0.87	−0.53
XOR	0.55	0.77	0.83	0.83	1.00	0.12

Table 100: Correlation coefficients among metrics and potential energy for simulation of 1GB1 protein with OPLS force field. Rows simultaneously represent the best trajectory according to the listed metric and correlation between this metric and other metrics and potential energy.

	RMSD	ANGL	ANDH	AND	XOR	Potential energy
RMSD	1.00	0.75	0.72	0.77	0.12	0.77
ANGL	−0.16	1.00	0.91	0.94	0.92	0.43
ANDH	0.01	0.90	1.00	0.99	0.92	0.50
AND	−0.01	0.94	0.99	1.00	0.93	0.51
XOR	−0.21	0.91	0.92	0.94	1.00	0.29

Table 101: Determination coefficients among metrics and potential energy for the first simulation of 1L2Y protein with AMBER force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−2.40	−0.51	−2.97	−1.14	−2.85	−0.89	−4.41	−2.34	−0.63	−1.46
ANGL	−0.82	−3.14	1.00	1.00	0.74	0.66	0.79	0.75	0.68	0.60	0.12	0.43
ANDH	−4.04	−5.18	−0.02	0.45	1.00	1.00	0.98	0.98	0.88	0.89	−0.41	−0.73
AND	−3.44	−4.97	0.22	0.54	0.98	0.98	1.00	1.00	0.89	0.89	−0.31	−0.46
XOR	−5.90	−6.61	−0.69	0.18	0.75	0.80	0.67	0.76	1.00	1.00	−0.99	−1.72

Table 102: Determination coefficients among metrics and potential energy for the second simulation of 1L2Y protein with AMBER force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−1.51	0.25	−3.16	−1.16	−2.92	−0.75	−3.62	−1.19	−0.29	−1.13
ANGL	−0.12	−3.44	1.00	1.00	0.75	0.55	0.77	0.63	0.65	0.39	−0.41	0.34
ANDH	−2.98	−5.00	−0.68	0.28	1.00	1.00	0.98	0.98	0.92	0.93	−0.36	−0.53
AND	−2.64	−5.00	−0.44	0.33	0.98	0.98	1.00	1.00	0.93	0.93	−0.34	−0.37
XOR	−2.81	−5.36	−0.59	0.25	0.93	0.92	0.93	0.93	1.00	1.00	−0.57	−0.59

Table 103: Determination coefficients among metrics and potential energy for the first simulation of 1L2Y protein with CHARMM force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−2.57	−0.14	−3.36	−1.63	−3.15	−0.87	−3.48	−1.03	−1.41	−2.81
ANGL	−0.39	−4.19	1.00	1.00	0.75	0.56	0.80	0.69	0.67	0.45	0.24	0.65
ANDH	−1.93	−6.17	0.57	0.65	1.00	1.00	0.99	0.99	0.94	0.94	−0.13	0.34
AND	−1.73	−6.27	0.66	0.70	0.99	0.99	1.00	1.00	0.96	0.96	−0.14	0.39
XOR	−1.67	−6.09	0.56	0.64	0.95	0.95	0.95	0.96	1.00	1.00	−0.34	0.17

Table 104: Determination coefficients among metrics and potential energy for the second simulation of 1L2Y protein with CHARMM force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−1.60	0.08	−2.34	−1.72	−2.53	−1.61	−2.21	−1.99	−1.15	−2.08
ANGL	−0.74	−3.10	1.00	1.00	0.83	0.72	0.85	0.79	0.79	0.63	0.74	0.81
ANDH	−1.84	−3.35	0.60	0.78	1.00	1.00	0.99	0.99	0.95	0.94	0.61	0.55
AND	−2.23	−3.74	0.61	0.78	0.99	0.99	1.00	1.00	0.92	0.91	0.60	0.52
XOR	−2.03	−3.03	0.50	0.75	0.95	0.96	0.93	0.95	1.00	1.00	0.57	0.44

Table 105: Determination coefficients among metrics and potential energy for the first simulation of 1L2Y protein with GROMOS force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−0.36	−0.18	−2.34	−2.36	−2.30	−2.02	−1.81	−1.56	−2.47	−1.81
ANGL	−1.27	−1.70	1.00	1.00	0.76	0.81	0.79	0.85	0.67	0.64	−0.23	0.14
ANDH	−4.97	−4.16	−0.21	0.16	1.00	1.00	0.85	0.89	0.13	0.43	−0.17	−0.24
AND	−2.63	−3.39	0.43	0.47	0.94	0.93	1.00	1.00	0.84	0.83	−0.18	0.19
XOR	−3.10	−3.21	0.45	0.55	0.81	0.81	0.85	0.88	1.00	1.00	−0.34	−0.08

Table 106: Determination coefficients among metrics and potential energy for the second simulation of 1L2Y protein with GROMOS force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−1.63	−0.66	−2.44	−1.88	−2.49	−2.04	−1.06	−1.37	−2.24	−2.63
ANGL	−1.52	−2.83	1.00	1.00	0.86	0.84	0.85	0.84	0.61	0.31	0.34	0.35
ANDH	−4.53	−4.42	0.29	0.62	1.00	1.00	0.89	0.91	0.16	0.12	0.01	−0.49
AND	−4.34	−4.32	0.26	0.61	0.95	0.95	1.00	1.00	0.20	0.11	0.32	0.10
XOR	−6.57	−3.38	−0.23	0.54	0.56	0.71	0.64	0.79	1.00	1.00	0.20	−0.62

Table 107: Determination coefficients among metrics and potential energy for the first simulation of 1L2Y protein with OPLS force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−6.02	−3.04	−8.25	−6.29	−7.80	−4.99	−7.54	−4.07	−4.10	−8.38
ANGL	−0.53	−2.58	1.00	1.00	−0.36	−1.18	0.05	−0.78	−0.40	−1.37	−1.45	−0.68
ANDH	−0.90	−1.28	−0.48	−0.11	1.00	1.00	0.95	0.95	0.81	0.79	−0.19	−0.31
AND	−0.80	−1.05	−0.55	0.18	0.94	0.95	1.00	1.00	0.68	0.74	−0.06	−0.63
XOR	−1.58	−1.43	−1.49	−0.43	0.25	0.53	0.30	0.49	1.00	1.00	−0.45	−0.97

Table 108: Determination coefficients among metrics and potential energy for the second simulation of 1L2Y protein with OPLS force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−2.01	−0.11	−4.83	−1.65	−3.91	−1.23	−3.84	−1.19	−0.43	−1.40
ANGL	−1.11	−3.42	1.00	1.00	0.82	0.84	0.88	0.89	0.83	0.84	−0.65	−0.32
ANDH	−2.87	−5.86	0.47	0.60	1.00	1.00	0.97	0.97	0.87	0.89	−1.31	−1.24
AND	−2.01	−4.77	0.68	0.75	0.98	0.99	1.00	1.00	0.96	0.95	−1.00	−0.75
XOR	−2.38	−5.10	0.57	0.68	0.94	0.95	0.93	0.94	1.00	1.00	−1.24	−1.09

Table 109: Determination coefficients among metrics and potential energy for simulation of 1YRF protein with AMBER force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−0.12	0.49	−1.21	−0.03	−1.11	0.08	−1.65	−0.32	0.07	0.09
ANGL	0.11	0.02	1.00	1.00	0.34	−0.31	0.47	0.05	0.26	−0.29	0.34	0.48
ANDH	−1.10	−1.15	0.80	0.81	1.00	1.00	0.99	0.99	0.97	0.98	0.51	0.65
AND	−1.10	−0.99	0.81	0.81	0.99	0.99	1.00	1.00	0.97	0.97	0.71	0.77
XOR	−1.02	−0.87	0.67	0.70	0.89	0.91	0.92	0.93	1.00	1.00	0.58	0.65

Table 110: Determination coefficients among metrics and potential energy for simulation of 1YRF protein with CHARMM force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−1.00	0.05	−2.26	−1.12	−2.29	−0.98	−1.31	−0.80	−0.49	−2.15
ANGL	−0.78	−4.39	1.00	1.00	0.79	0.69	0.85	0.81	0.80	0.69	−0.12	0.39
ANDH	−0.47	−2.12	0.86	0.84	1.00	1.00	0.99	0.99	0.95	0.94	0.72	0.65
AND	−0.66	−2.26	0.82	0.80	0.98	0.98	1.00	1.00	0.95	0.93	0.74	0.66
XOR	−0.82	−1.89	0.77	0.79	0.89	0.92	0.93	0.95	1.00	1.00	0.66	0.46

Table 111: Determination coefficients among metrics and potential energy for simulation of 1YRF protein with GROMOS force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−2.18	−1.65	−3.68	−3.11	−2.14	−1.93	−4.12	−3.70	−1.05	−0.71
ANGL	0.43	−0.08	1.00	1.00	0.52	0.61	0.49	0.60	0.33	0.37	0.35	0.35
ANDH	−1.04	−2.00	−0.67	−0.35	1.00	1.00	0.83	0.88	0.62	0.72	0.31	0.14
AND	−0.32	−1.41	−0.04	−0.04	0.75	0.79	1.00	1.00	0.84	0.84	0.38	0.39
XOR	−1.27	−2.86	−0.71	−0.54	0.46	0.59	0.72	0.79	1.00	1.00	−0.06	−0.18

Table 112: Determination coefficients among metrics and potential energy for simulation of 1YRF protein with OPLS force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−3.17	−0.11	−1.38	0.04	−1.40	0.07	−2.94	−0.56	−0.11	−0.33
ANGL	−0.54	−3.91	1.00	1.00	0.77	0.71	0.82	0.77	0.89	0.86	−1.48	0.05
ANDH	−1.27	−2.72	0.71	0.83	1.00	1.00	0.99	0.99	0.96	0.96	0.35	0.58
AND	−1.14	−2.82	0.67	0.84	0.99	0.99	1.00	1.00	0.93	0.94	−0.68	−0.11
XOR	−1.98	−2.77	0.76	0.86	0.93	0.93	0.92	0.93	1.00	1.00	0.24	0.50

Table 113: Determination coefficients among metrics and potential energy for simulation of 1GB1 protein with AMBER force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−2.86	−1.82	−4.57	−4.19	−3.15	−2.80	−3.65	−3.40	−0.04	0.10
ANGL	−0.57	−2.14	1.00	1.00	0.65	0.62	0.71	0.64	−0.28	−0.68	0.25	0.34
ANDH	−1.30	−3.00	0.02	−0.49	1.00	1.00	0.97	0.97	0.96	0.95	0.51	0.56
AND	−0.92	−2.42	0.24	−0.19	0.97	0.97	1.00	1.00	0.94	0.94	0.55	0.60
XOR	−2.01	−3.41	−0.32	−0.67	0.90	0.92	0.86	0.88	1.00	1.00	0.42	0.39

Table 114: Determination coefficients among metrics and potential energy for simulation of 1GB1 protein with CHARMM force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−3.34	−3.20	−5.53	−4.96	−5.14	−4.26	−9.95	−6.55	−1.06	−2.44
ANGL	−5.84	−2.85	1.00	1.00	−0.03	0.29	0.13	0.45	0.38	0.56	−1.85	−2.36
ANDH	−0.96	−0.21	0.31	−0.58	1.00	1.00	0.96	0.97	0.87	0.88	−0.16	0.10
AND	−1.05	−0.50	0.41	−0.13	0.99	0.99	1.00	1.00	0.87	0.86	0.12	0.26
XOR	−1.11	−0.46	0.13	−0.55	0.84	0.85	0.86	0.87	1.00	1.00	0.02	0.12

Table 115: Determination coefficients among metrics and potential energy for simulation of 1GB1 protein with GROMOS force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−1.39	−0.62	−1.92	−1.31	−2.27	−1.68	−2.53	−2.52	−2.75	−2.95
ANGL	−1.08	−8.59	1.00	1.00	0.86	0.80	0.84	0.73	0.72	0.54	−4.44	−1.20
ANDH	−3.80	−10	0.47	0.73	1.00	1.00	0.95	0.95	0.61	0.62	−2.61	−1.88
AND	−3.85	−9.92	0.45	0.73	0.95	0.95	1.00	1.00	0.62	0.65	−2.53	−1.95
XOR	−4.67	−3.46	−0.16	0.39	0.30	0.57	0.42	0.63	1.00	1.00	−1.04	−1.69

Table 116: Determination coefficients among metrics and potential energy for simulation of 1GB1 protein with OPLS force field. Rows simultaneously represent the best trajectory according to the listed metric and determination between this metric and other metrics and potential energy.

	RMSD		ANGL		ANDH		AND		XOR		Potential energy	
	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}	r^2_{xy}	r^2_{yx}
RMSD	1.00	1.00	−2.98	−1.24	−3.20	−1.71	−2.70	−1.30	−4.62	−3.74	−0.95	−1.20
ANGL	−1.30	−8.81	1.00	1.00	0.80	0.66	0.86	0.79	0.82	0.73	−2.38	−0.02
ANDH	−4.18	−8.47	0.19	0.66	1.00	1.00	0.97	0.97	0.81	0.80	−1.89	−1.00
AND	−3.36	−8.24	0.51	0.76	0.97	0.97	1.00	1.00	0.83	0.81	−1.76	−0.65
XOR	−5.65	−10	−0.13	0.55	0.80	0.81	0.75	0.80	1.00	1.00	−3.03	−1.96

APPENDIX B

Extra Tables for Chapter III

Table 117: A1. Relationship between protein types and affinities types. Graph width analysis. Raw values.

			NFP				IDP				Total				
			N	M	W	C	N	M	W	C	N	M	W	C	
Entropic	DN	KNN	6	5	7	3	6	4	8	5	12	9	15	8	5
		Perp	10	5	3	3	9	5	4	5	19	10	7	8	6
	SP	KNN	16	2	0	1	9	6	3	2	25	8	3	3	7
		Perp	13	5	0	3	13	5	0	3	26	10	0	6	8
	SS	KNN	13	2	3	0	9	1	8	3	22	3	11	3	9
		Perp	15	3	0	2	13	4	1	4	28	7	1	6	10
Plain	DN	KNN	0	0	0	0	0	0	0	0	0	0	0	0	11
		Perp	0	0	0	0	0	0	0	0	0	0	0	0	12
	SP	KNN	3	2	13	3	3	0	15	4	6	2	28	7	13
		Perp	5	4	9	1	0	4	14	2	5	8	23	3	14
	SS	KNN	10	4	4	3	5	4	9	6	15	8	13	9	15
		Perp	0	3	15	0	0	0	18	1	0	3	33	1	16
Total	DN	KNN	6	5	7	3	6	4	8	5	12	9	15	8	17
		Perp	10	5	3	3	9	5	4	5	19	10	7	8	18
	SP	KNN	19	4	13	4	12	6	18	6	31	10	31	1 0	19
		Perp	18	9	9	4	13	9	14	5	31	18	23	9	20
	SS	KNN	23	6	7	3	14	5	17	9	37	11	24	12	21
		Perp	15	6	15	2	13	4	19	5	28	10	34	7	22
Total	En	KNN	35	9	10	4	24	11	19	10	59	20	29	14	23
		Perp	38	13	3	8	35	14	5	12	73	27	8	2 0	24
	Pl	KNN	13	6	17	6	8	4	24	10	21	10	41	16	25
		Perp	5	7	24	1	0	4	32	3	5	11	56	4	26
Total		KNN	48	15	27	10	32	15	43	20	80	30	70	3 0	27
		Perp	43	20	27	9	35	18	37	15	78	38	64	24	28
			E	F	G	H	I	J	K	L	M	N	O	P	

Table 118: A2. Relationship the data sparsity and graphs' width. Clustering algorithms analysis. Raw values.

			N			M			W			C			Total				
			SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	N	M	W	C	
Entropic	DN	KNN	1	7	4	6	3	0	5	2	8	2	2	4	12	9	15	8	33
		Perp	11	6	2	1	5	4	0	1	6	1	3	4	19	10	7	8	34
	SP	KNN	5	12	8	4	0	4	3	0	0	2	0	1	25	8	3	3	35
		Perp	10	9	7	2	3	5	0	0	0	2	3	1	26	10	0	6	36
	SS	KNN	1	10	11	2	1	0	9	1	1	1	1	1	22	3	11	3	37
		Perp	8	9	11	4	2	1	0	1	0	3	2	1	28	7	1	6	38
Plain	DN	KNN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39
		Perp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40
	SP	KNN	2	0	4	0	2	0	10	10	8	3	1	3	6	2	28	7	41
		Perp	2	0	3	5	3	0	5	9	9	2	1	0	5	8	23	3	42
	SS	KNN	5	2	8	3	3	2	4	7	2	4	3	2	15	8	13	9	43
		Perp	0	0	0	0	3	0	12	9	12	1	0	0	0	3	33	1	44
Total	DN	KNN	1	7	4	6	3	0	5	2	8	2	2	4	12	9	15	8	45
		Perp	11	6	2	1	5	4	0	1	6	1	3	4	19	10	7	8	46
	SP	KNN	7	12	12	4	2	4	13	10	8	5	1	4	31	10	31	10	47
		Perp	12	9	10	7	6	5	5	9	9	4	4	1	31	18	23	9	48
	SS	KNN	6	12	19	5	4	2	13	8	3	5	4	3	37	11	24	12	49
		Perp	8	9	11	4	5	1	12	10	12	4	2	1	28	10	34	7	50
Total	En	KNN	7	29	23	12	4	4	17	3	9	5	3	6	59	20	29	14	51
		Perp	29	24	20	7	10	10	0	2	6	6	8	6	73	27	8	20	52
	Pl	KNN	7	2	12	3	5	2	14	17	10	7	4	5	21	10	41	16	53
		Perp	2	0	3	5	6	0	17	18	21	3	1	0	5	11	56	4	54
			E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	

Table 119: A3. Relationship between graphs' width and protein types. Clustering algorithms analysis. Raw values.

			N			M			W			C			Total				
			SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	N	M	W	C	
NFP	DN	KNN	0	4	2	3	2	0	3	0	4	0	1	2	6	5	7	3	59
		Perp	6	2	2	0	4	1	0	0	3	0	1	2	10	5	3	3	60
	SP	KNN	5	6	8	1	2	1	6	4	3	1	1	2	19	4	13	4	61
		Perp	8	5	5	1	4	4	3	3	3	1	2	1	18	9	9	4	62
	SS	KNN	3	8	12	4	2	0	5	2	0	2	1	0	23	6	7	3	63
		Perp	4	5	6	2	4	0	6	3	6	1	1	0	15	6	15	2	64
IDP	DN	KNN	1	3	2	3	1	0	2	2	4	2	1	2	6	4	8	5	65
		Perp	5	4	0	1	1	3	0	1	3	1	2	2	9	5	4	5	66
	SP	KNN	2	6	4	3	0	3	7	6	5	4	0	2	12	6	18	6	67
		Perp	4	4	5	6	2	1	2	6	6	3	2	0	13	9	14	5	68
	SS	KNN	3	4	7	1	2	2	8	6	3	3	3	3	14	5	17	9	69
		Perp	4	4	5	2	1	1	6	7	6	3	1	1	13	4	19	5	70
Total	DN	KNN	1	7	4	6	3	0	5	2	8	2	2	4	12	9	15	8	71
		Perp	11	6	2	1	5	4	0	1	6	1	3	4	19	10	7	8	72
	SP	KNN	7	12	12	4	2	4	13	10	8	5	1	4	31	10	31	10	73
		Perp	12	9	10	7	6	5	5	9	9	4	4	1	31	18	23	9	74
	SS	KNN	6	12	19	5	4	2	13	8	3	5	4	3	37	11	24	12	75
		Perp	8	9	11	4	5	1	12	10	12	4	2	1	28	10	34	7	76
Total	NFP	KNN	8	18	22	8	6	1	14	6	7	3	3	4	48	15	27	10	77
		Perp	18	12	13	3	12	5	9	6	12	2	4	3	43	20	27	9	78
	IDP	KNN	6	13	13	7	3	5	17	14	12	9	4	7	32	15	43	20	79
		Perp	13	12	10	9	4	5	8	14	15	7	5	3	35	18	37	15	80
Total		KNN	14	31	35	15	9	6	31	20	19	12	7	11	80	30	70	30	81
		Perp	31	24	23	12	16	10	17	20	27	9	9	6	78	38	64	24	82
			E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	

Table 120: A4. Relationship between affinity types and protein types. Graph width analysis. Raw values.

			Entropic				Plain				Total				
			N	M	W	C	N	M	W	C	N	M	W	C	
NFP	SPC	KNN	6	6	6	0	2	2	8	3	8	8	14	3	87
		Perp	16	2	0	1	2	1	9	1	18	3	9	2	88
	SDS	KNN	16	2	0	1	2	4	6	2	18	6	6	3	89
		Perp	12	6	0	4	0	6	6	0	12	12	6	4	90
	SES	KNN	13	1	4	3	9	0	3	1	22	1	7	4	91
		Perp	10	5	3	3	3	0	9	0	13	5	12	3	92
IDP	SPC	KNN	1	6	11	5	5	1	6	4	6	7	17	9	93
		Perp	13	5	0	5	0	4	8	2	13	9	8	7	94
	SDS	KNN	13	2	3	2	0	1	11	2	13	3	14	4	95
		Perp	12	4	2	4	0	0	12	1	12	4	14	5	96
	SES	KNN	10	3	5	3	3	2	7	4	13	5	12	7	97
		Perp	10	5	3	3	0	0	12	0	10	5	15	3	98
Total	SPC	KNN	7	12	17	5	7	3	14	7	14	15	31	12	99
		Perp	29	7	0	6	2	5	17	3	31	12	17	9	100
	SDS	KNN	29	4	3	3	2	5	17	4	31	9	20	7	101
		Perp	24	10	2	8	0	6	18	1	24	16	20	9	102
	SES	KNN	23	4	9	6	12	2	10	5	35	6	19	11	103
		Perp	20	10	6	6	3	0	21	0	23	10	27	6	104
Total	NFP	KNN	35	9	10	4	13	6	17	6	48	15	27	10	105
		Perp	38	13	3	8	5	7	24	1	43	20	27	9	106
	IDP	KNN	24	11	19	10	8	4	24	10	32	15	43	20	107
		Perp	35	14	5	12	0	4	32	3	35	18	37	15	108
Total		KNN	59	20	29	14	21	10	41	16	80	30	70	30	109
		Perp	73	27	8	20	5	11	56	4	78	38	64	24	110
			E	F	G	H	I	J	K	L	M	N	O	P	

Table 121: A4.1. Relationship between the data sparsity and protein types. Graph width analysis. Raw values.

			DN				SP				SS				Total				
			N	M	W	C	N	M	W	C	N	M	W	C	N	M	W	C	
NFP	SPC	KNN	0	3	3	0	5	1	6	1	3	4	5	2	8	8	14	3	115
		Perp	6	0	0	0	8	1	3	1	4	2	6	1	18	3	9	2	116
	SDS	KNN	4	2	0	1	6	2	4	1	8	2	2	1	18	6	6	3	117
		Perp	2	4	0	1	5	4	3	2	5	4	3	1	12	12	6	4	118
	SES	KNN	2	0	4	2	8	1	3	2	12	0	0	0	22	1	7	4	119
		Perp	2	1	3	2	5	4	3	1	6	0	6	0	13	5	12	3	120
IDP	SPC	KNN	1	3	2	2	2	3	7	4	3	1	8	3	6	7	17	9	121
		Perp	5	1	0	1	4	6	2	3	4	2	6	3	13	9	8	7	122
	SDS	KNN	3	1	2	1	6	0	6	0	4	2	6	3	13	3	14	4	123
		Perp	4	1	1	2	4	2	6	2	4	1	7	1	12	4	14	5	124
	SES	KNN	2	0	4	2	4	3	5	2	7	2	3	3	13	5	12	7	125
		Perp	0	3	3	2	5	1	6	0	5	1	6	1	10	5	15	3	126
Total	SPC	KNN	1	6	5	2	7	4	13	5	8	10	18	7	15	14	31	12	127
		Perp	11	1	0	1	12	7	5	4	23	8	5	5	35	15	10	9	128
	SDS	KNN	7	3	2	2	12	2	10	1	19	5	12	3	31	7	22	4	129
		Perp	6	5	1	3	9	6	9	4	15	11	10	7	24	17	19	11	130
	SES	KNN	4	0	8	4	12	4	8	4	16	4	16	8	28	8	24	12	131
		Perp	2	4	6	4	10	5	9	1	12	9	15	5	22	14	24	6	132
Total	NFP	KNN	6	5	7	3	19	4	13	4	25	9	20	7	44	13	33	11	133
		Perp	10	5	3	3	18	9	9	4	28	14	12	7	46	23	21	11	134
	IDP	KNN	6	4	8	5	12	6	18	6	18	10	26	11	30	16	44	17	135
		Perp	9	5	4	5	13	9	14	5	22	14	18	10	35	23	32	15	136
Total		KNN	12	9	15	8	31	10	31	10	43	19	46	18	74	29	77	28	137
		Perp	19	10	7	8	31	18	23	9	50	28	30	17	81	46	53	26	138
			E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	

Table 122: A5. Relationship between protein types and affinity types. Trend line direction analysis. Raw values.

			NFP					IDP					Total					
			/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	
Entropic	DN	KNN	5	10	3	3	1	4	12	2	3	2	9	22	5	6	3	145
		Perp	2	13	3	1	0	2	7	9	1	2	4	20	12	2	2	146
	SP	KNN	2	16	0	0	0	3	12	3	2	0	5	28	3	2	0	147
		Perp	2	13	3	2	0	1	14	3	1	1	3	27	6	3	1	148
	SS	KNN	0	18	0	0	0	2	15	1	0	1	2	33	1	0	1	149
		Perp	1	17	0	1	0	4	12	2	2	0	5	29	2	3	0	150
Plain	DN	KNN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
		Perp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152
	SP	KNN	4	11	3	3	2	5	9	4	5	2	9	20	7	8	4	153
		Perp	0	17	1	0	1	0	16	2	2	2	0	33	3	2	3	154
	SS	KNN	7	8	3	3	5	7	6	5	5	4	14	14	8	8	9	155
		Perp	0	17	1	0	0	0	16	2	2	2	0	33	3	2	2	156
Total	DN	KNN	5	10	3	3	1	4	12	2	3	2	9	22	5	6	3	157
		Perp	2	13	3	1	0	2	7	9	1	2	4	20	12	2	2	158
	SP	KNN	6	27	3	3	2	8	21	7	7	2	14	48	10	10	4	159
		Perp	2	30	4	2	1	1	30	5	3	3	3	60	9	5	4	160
	SS	KNN	7	26	3	3	5	9	21	6	5	5	16	47	9	8	10	161
		Perp	1	34	1	1	0	4	28	4	4	2	5	62	5	5	2	162
Total	En	KNN	7	44	3	3	1	9	39	6	5	3	16	83	9	8	4	163
		Perp	5	43	6	4	0	7	33	14	4	3	12	76	20	8	3	164
	Pl	KNN	11	19	6	6	7	12	15	9	10	6	23	34	15	16	13	165
		Perp	0	34	2	0	1	0	32	4	4	4	0	66	6	4	5	166
			E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	

Table 123: A6. Relationship between graph trend direction and affinity types. Clustering algorithms analysis. Raw values.

			/			-			\			-			S			Total					
			SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	/	-	\	C	S	
Entropic	DN	KNN	1	2	6	10	9	3	1	1	3	0	2	4	0	0	3	9	22	5	6	3	171
		Perp	1	2	1	6	9	5	5	1	6	0	1	1	1	0	1	4	20	12	2	2	172
	SP	KNN	0	1	4	11	10	7	1	1	1	1	1	0	0	0	0	5	28	3	2	0	173
		Perp	0	1	2	8	10	9	4	1	1	1	1	1	1	0	0	3	27	6	3	1	174
	SS	KNN	0	0	2	12	11	10	0	1	0	0	0	0	0	1	0	2	33	1	0	1	175
		Perp	3	0	2	8	11	10	1	1	0	3	0	0	0	0	0	5	29	2	3	0	176
Plain	DN	KNN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	177
		Perp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	178
	SP	KNN	1	1	7	10	5	5	1	6	0	3	4	1	0	2	2	9	20	7	8	4	179
		Perp	0	0	0	12	9	12	0	3	0	1	1	0	0	3	0	0	33	3	2	3	180
	SS	KNN	3	0	11	7	6	1	2	6	0	4	3	1	2	3	4	14	14	8	8	9	181
		Perp	0	0	0	12	9	12	0	3	0	1	1	0	0	2	0	0	33	3	2	2	182
Total	DN	KNN	1	2	6	10	9	3	1	1	3	0	2	4	0	0	3	9	22	5	6	3	183
		Perp	1	2	1	6	9	5	5	1	6	0	1	1	1	0	1	4	20	12	2	2	184
	SP	KNN	1	2	11	21	15	12	2	7	1	4	5	1	0	2	2	14	48	10	10	4	185
		Perp	0	1	2	20	19	21	4	4	1	2	2	1	1	3	0	3	60	9	5	4	186
	SS	KNN	3	0	13	19	17	11	2	7	0	4	3	1	2	4	4	16	47	9	8	10	187
		Perp	3	0	2	20	20	22	1	4	0	4	1	0	0	2	0	5	62	5	5	2	188
Total	En	KNN	1	3	12	33	30	20	2	3	4	1	3	4	0	1	3	16	83	9	8	4	189
		Perp	4	3	5	22	30	24	10	3	7	4	2	2	2	0	1	12	76	20	8	3	190
	Pl	KNN	4	1	18	17	11	6	3	12	0	7	7	2	2	5	6	23	34	15	16	13	191
		Perp	0	0	0	24	18	24	0	6	0	2	2	0	0	5	0	0	66	6	4	5	192
			E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	

Table 124: A7. Relationship between graph trend direction and protein types. Clustering algorithms analysis. Raw values.

			/			-			\			C			S			Total					
			SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	/	-	\	C	S	
NFP	DN	KNN	1	1	3	5	4	1	0	1	2	0	1	2	0	0	1	5	10	3	3	1	197
		Perp	1	0	1	4	6	3	1	0	2	0	0	1	0	0	0	2	13	3	1	0	198
	SP	KNN	0	1	5	12	8	7	0	3	0	1	2	0	0	1	1	6	27	3	3	2	199
		Perp	0	1	1	11	9	10	1	2	1	0	1	1	0	1	0	2	30	4	2	1	200
	SS	KNN	2	0	5	9	10	7	1	2	0	2	1	0	2	1	2	7	26	3	3	5	201
		Perp	1	0	0	11	11	12	0	1	0	1	0	0	0	0	0	1	34	1	1	0	202
IDP	DN	KNN	0	1	3	5	5	2	1	0	1	0	1	2	0	0	2	4	12	2	3	2	203
		Perp	0	2	0	2	3	2	4	1	4	0	1	0	1	0	1	2	7	9	1	2	204
	SP	KNN	1	1	6	9	7	5	2	4	1	3	3	1	0	1	1	8	21	7	7	2	205
		Perp	0	0	1	9	10	11	3	2	0	2	1	0	1	2	0	1	30	5	3	3	206
	SS	KNN	1	0	8	10	7	4	1	5	0	2	2	1	0	3	2	9	21	6	5	5	207
		Perp	2	0	2	9	9	10	1	3	0	3	1	0	0	2	0	4	28	4	4	2	208
Total	DN	KNN	1	2	6	10	9	3	1	1	3	0	2	4	0	0	3	9	22	5	6	3	209
		Perp	1	2	1	6	9	5	5	1	6	0	1	1	1	0	1	4	20	12	2	2	210
	SP	KNN	1	2	11	21	15	12	2	7	1	4	5	1	0	2	2	14	48	10	10	4	211
		Perp	0	1	2	20	19	21	4	4	1	2	2	1	1	3	0	3	60	9	5	4	212
	SS	KNN	3	0	13	19	17	11	2	7	0	4	3	1	2	4	4	16	47	9	8	10	213
		Perp	3	0	2	20	20	22	1	4	0	4	1	0	0	2	0	5	62	5	5	2	214
Total	NFP	KNN	3	2	13	26	22	15	1	6	2	3	4	2	2	2	4	18	63	9	9	8	215
		Perp	2	1	2	26	26	25	2	3	3	1	1	2	0	1	0	5	77	8	4	1	216
	IDP	KNN	2	2	17	24	19	11	4	9	2	5	6	4	0	4	5	21	54	15	15	9	217
		Perp	2	2	3	20	22	23	8	6	4	5	3	0	2	4	1	7	65	18	8	7	218
			E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	

Table 125: A8. Relationship between clustering algorithms and protein types. Trend line direction analysis. Raw values.

			SPC					SDS					SES					Total				
			/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	/	-	\	C	S
NFP	DN	KNN	1	5	0	0	0	1	4	1	1	0	3	1	2	2	1	5	10	3	3	1
		Perp	1	4	1	0	0	0	6	0	0	0	1	3	2	1	0	2	13	3	1	0
	SP	KNN	0	12	0	1	0	1	8	3	2	1	5	7	0	0	1	6	27	3	3	2
		Perp	0	11	1	0	0	1	9	2	1	1	1	10	1	1	0	2	30	4	2	1
	SS	KNN	2	9	1	2	2	0	10	2	1	1	5	7	0	0	2	7	26	3	3	5
		Perp	1	11	0	1	0	0	11	1	0	0	0	12	0	0	0	1	34	1	1	0
IDP	DN	KNN	0	5	1	0	0	1	5	0	1	0	3	2	1	2	2	4	12	2	3	2
		Perp	0	2	4	0	1	2	3	1	1	0	0	2	4	0	1	2	7	9	1	2
	SP	KNN	1	9	2	3	0	1	7	4	3	1	6	5	1	1	1	8	21	7	7	2
		Perp	0	9	3	2	1	0	10	2	1	2	1	11	0	0	0	1	30	5	3	3
	SS	KNN	1	10	1	2	0	0	7	5	2	3	8	4	0	1	2	9	21	6	5	5
		Perp	2	9	1	3	0	0	9	3	1	2	2	10	0	0	0	4	28	4	4	2
Total	DN	KNN	1	10	1	0	0	2	9	1	2	0	6	3	3	4	3	9	22	5	6	3
		Perp	1	6	5	0	1	2	9	1	1	0	1	5	6	1	1	4	20	12	2	2
	SP	KNN	1	21	2	4	0	2	15	7	5	2	11	12	1	1	2	14	48	10	10	4
		Perp	0	20	4	2	1	1	19	4	2	3	2	21	1	1	0	3	60	9	5	4
	SS	KNN	3	19	2	4	2	0	17	7	3	4	13	11	0	1	4	16	47	9	8	10
		Perp	3	20	1	4	0	0	20	4	1	2	2	22	0	0	0	5	62	5	5	2
Total	NFP	KNN	3	26	1	3	2	2	22	6	4	2	13	15	2	2	4	18	63	9	9	8
		Perp	2	26	2	1	0	1	26	3	1	1	2	25	3	2	0	5	77	8	4	1
	IDP	KNN	2	24	4	5	0	2	19	9	6	4	17	11	2	4	5	21	54	15	15	9
		Perp	2	20	8	5	2	2	22	6	3	4	3	23	4	0	1	7	65	18	8	7
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV

Table 126: A9. Relationship between affinity types and protein types. Trend line direction analysis. Raw values.

			Entropic					Plain					Total				
			/	-	\	C	S	/	-	\	C	S	/	-	\	C	S
NFP	SPC	KNN	1	17	0	0	0	2	9	1	3	2	3	26	1	3	2
		Perp	2	14	2	1	0	0	12	0	0	0	2	26	2	1	0
	SDS	KNN	1	16	1	1	0	1	6	5	3	2	2	22	6	4	2
		Perp	1	16	1	1	0	0	10	2	0	1	1	26	3	1	1
	SES	KNN	5	11	2	2	1	8	4	0	0	3	13	15	2	2	4
		Perp	2	13	3	2	0	0	12	0	0	0	2	25	3	2	0
IDP	SPC	KNN	0	16	2	1	0	2	8	2	4	0	2	24	4	5	0
		Perp	2	8	8	3	2	0	12	0	2	0	2	20	8	5	2
	SDS	KNN	2	14	2	2	1	0	5	7	4	3	2	19	9	6	4
		Perp	2	14	2	1	0	0	8	4	2	4	2	22	6	3	4
	SES	KNN	7	9	2	2	2	10	2	0	2	3	17	11	2	4	5
		Perp	3	11	4	0	1	0	12	0	0	0	3	23	4	0	1
Total	SPC	KNN	1	33	2	1	0	4	17	3	7	2	5	50	5	8	2
		Perp	4	22	10	4	2	0	24	0	2	0	4	46	10	6	2
	SDS	KNN	3	30	3	3	1	1	11	12	7	5	4	41	15	10	6
		Perp	3	30	3	2	0	0	18	6	2	5	3	48	9	4	5
	SES	KNN	12	20	4	4	3	18	6	0	2	6	30	26	4	6	9
		Perp	5	24	7	2	1	0	24	0	0	0	5	48	7	2	1
Total	NFP	KNN	7	44	3	3	1	11	19	6	6	7	18	63	9	9	8
		Perp	5	43	6	4	0	0	34	2	0	1	5	77	8	4	1
	IDP	KNN	9	39	6	5	3	12	15	9	10	6	21	54	15	15	9
		Perp	7	33	14	4	3	0	32	4	4	4	7	65	18	8	7
Total		KNN	16	83	9	8	4	23	34	15	16	13	39	117	24	24	17
		Perp	12	76	20	8	3	0	66	6	4	5	12	142	26	12	8
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ

Table 127: A10. Relationship between data sparsity and protein types. Trend line direction analysis. Raw values.

			DN					SP					SS					Total				
			/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	/	-	\	C	S
NFP	SPC	KNN	1	5	0	0	0	0	12	0	1	0	2	9	1	2	2	3	26	1	3	2
		Perp	1	4	1	0	0	0	11	1	0	0	1	11	0	1	0	2	26	2	1	0
	SDS	KNN	1	4	1	1	0	1	8	3	2	1	0	10	2	1	1	2	22	6	4	2
		Perp	0	6	0	0	0	1	9	2	1	1	0	11	1	0	0	1	26	3	1	1
	SES	KNN	3	1	2	2	1	5	7	0	0	1	5	7	0	0	2	13	15	2	2	4
		Perp	1	3	2	1	0	1	10	1	1	0	0	12	0	0	0	2	25	3	2	0
IDP	SPC	KNN	0	5	1	0	0	1	9	2	3	0	1	10	1	2	0	2	24	4	5	0
		Perp	0	2	4	0	1	0	9	3	2	1	2	9	1	3	0	2	20	8	5	2
	SDS	KNN	1	5	0	1	0	1	7	4	3	1	0	7	5	2	3	2	19	9	6	4
		Perp	2	3	1	1	0	0	10	2	1	2	0	9	3	1	2	2	22	6	3	4
	SES	KNN	3	2	1	2	2	6	5	1	1	1	8	4	0	1	2	17	11	2	4	5
		Perp	0	2	4	0	1	1	11	0	0	0	2	10	0	0	0	3	23	4	0	1
Total	SPC	KNN	1	10	1	0	0	1	21	2	4	0	3	19	2	4	2	5	50	5	8	2
		Perp	1	6	5	0	1	0	20	4	2	1	3	20	1	4	0	4	46	10	6	2
	SDS	KNN	2	9	1	2	0	2	15	7	5	2	0	17	7	3	4	4	41	15	10	6
		Perp	2	9	1	1	0	1	19	4	2	3	0	20	4	1	2	3	48	9	4	5
	SES	KNN	6	3	3	4	3	11	12	1	1	2	13	11	0	1	4	30	26	4	6	9
		Perp	1	5	6	1	1	2	21	1	1	0	2	22	0	0	0	5	48	7	2	1
Total	NFP	KNN	5	10	3	3	1	6	27	3	3	2	7	26	3	3	5	18	63	9	9	8
		Perp	2	13	3	1	0	2	30	4	2	1	1	34	1	1	0	5	77	8	4	1
	IDP	KNN	4	12	2	3	2	8	21	7	7	2	9	21	6	5	5	21	54	15	15	9
		Perp	2	7	9	1	2	1	30	5	3	3	4	28	4	4	2	7	65	18	8	7
Total		KNN	9	22	5	6	3	14	48	10	10	4	16	47	9	8	10	39	117	24	24	17
		Perp	4	20	12	2	2	3	60	9	5	4	5	62	5	5	2	12	142	26	12	8
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV

Table 128: B1. Relationship between protein types and affinities types. Graph width analysis. Percentage.

			NFP				IDP				Total				
			N	M	W	C	N	M	W	C	N	M	W	C	
Entropic	DN	KNN	33%	28%	39%	50%	33%	22%	44%	83%	33%	25%	42%	67%	5
		Perp	56%	28%	17%	50%	50%	28%	22%	83%	53%	28%	19%	67%	6
	SP	KNN	89%	11%	0%	17%	50%	33%	17%	33%	69%	22%	8%	25%	7
		Perp	72%	28%	0%	50%	72%	28%	0%	50%	72%	28%	0%	50%	8
	SS	KNN	72%	11%	17%	0%	50%	6%	44%	50%	61%	8%	31%	25%	9
		Perp	83%	17%	0%	33%	72%	22%	6%	67%	78%	19%	3%	50%	10
Plain	DN	KNN	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	11
		Perp	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	12
	SP	KNN	17%	11%	72%	50%	17%	0%	83%	67%	17%	6%	78%	58%	13
		Perp	28%	22%	50%	17%	0%	22%	78%	33%	14%	22%	64%	25%	14
	SS	KNN	56%	22%	22%	50%	28%	22%	50%	100%	42%	22%	36%	75%	15
		Perp	0%	17%	83%	0%	0%	0%	100%	17%	0%	8%	92%	8%	16
Total	DN	KNN	33%	28%	39%	50%	33%	22%	44%	83%	33%	25%	42%	67%	17
		Perp	56%	28%	17%	50%	50%	28%	22%	83%	53%	28%	19%	67%	18
	SP	KNN	53%	11%	36%	33%	33%	17%	50%	50%	43%	14%	43%	42%	19
		Perp	50%	25%	25%	33%	36%	25%	39%	42%	43%	25%	32%	38%	20
	SS	KNN	64%	17%	19%	25%	39%	14%	47%	75%	51%	15%	33%	50%	21
		Perp	42%	17%	42%	17%	36%	11%	53%	42%	39%	14%	47%	29%	22
Total	En	KNN	65%	17%	19%	22%	44%	20%	35%	56%	55%	19%	27%	39%	23
		Perp	70%	24%	6%	44%	65%	26%	9%	67%	68%	25%	7%	56%	24
	Pl	KNN	36%	17%	47%	50%	22%	11%	67%	83%	29%	14%	57%	67%	25
		Perp	14%	19%	67%	8%	0%	11%	89%	25%	7%	15%	78%	17%	26
Total		KNN	53%	17%	30%	33%	36%	17%	48%	67%	44%	17%	39%	50%	27
		Perp	48%	22%	30%	30%	39%	20%	41%	50%	43%	21%	36%	40%	28
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	

Table 129: B2. Relationship the data sparsity and graphs' width. Clustering algorithms analysis. Percentage.

			N			M			W			C			Total				
			SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	N	M	W	C	
Entropic	DN	KNN	8%	58%	33%	50%	25%	0%	42%	17%	67%	50%	50%	100%	33%	25%	42%	67%	33
		Perp	92%	50%	17%	8%	42%	33%	0%	8%	50%	25%	75%	100%	53%	28%	19%	67%	34
	SP	KNN	42%	100%	67%	33%	0%	33%	25%	0%	0%	50%	0%	25%	69%	22%	8%	25%	35
		Perp	83%	75%	58%	17%	25%	42%	0%	0%	0%	50%	75%	25%	72%	28%	0%	50%	36
	SS	KNN	8%	83%	92%	17%	8%	0%	75%	8%	8%	25%	25%	25%	61%	8%	31%	25%	37
		Perp	67%	75%	92%	33%	17%	8%	0%	8%	0%	75%	50%	25%	78%	19%	3%	50%	38
Plain	DN	KNN	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	39
		Perp	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	40
	SP	KNN	17%	0%	33%	0%	17%	0%	83%	83%	67%	75%	25%	75%	17%	6%	78%	58%	41
		Perp	17%	0%	25%	42%	25%	0%	42%	75%	75%	50%	25%	0%	14%	22%	64%	25%	42
	SS	KNN	42%	17%	67%	25%	25%	17%	33%	58%	17%	100%	75%	50%	42%	22%	36%	75%	43
		Perp	0%	0%	0%	0%	25%	0%	100%	75%	100%	25%	0%	0%	0%	8%	92%	8%	44
Total	DN	KNN	8%	58%	33%	50%	25%	0%	42%	17%	67%	50%	50%	100%	33%	25%	42%	67%	45
		Perp	92%	50%	17%	8%	42%	33%	0%	8%	50%	25%	75%	100%	53%	28%	19%	67%	46
	SP	KNN	29%	50%	50%	17%	8%	17%	54%	42%	33%	63%	13%	50%	43%	14%	43%	42%	47
		Perp	50%	38%	42%	29%	25%	21%	21%	38%	38%	50%	50%	13%	43%	25%	32%	38%	48
	SS	KNN	25%	50%	79%	21%	17%	8%	54%	33%	13%	63%	50%	38%	51%	15%	33%	50%	49
		Perp	33%	38%	46%	17%	21%	4%	50%	42%	50%	50%	25%	13%	39%	14%	47%	29%	50
Total	En	KNN	19%	81%	64%	33%	11%	11%	47%	8%	25%	42%	25%	50%	55%	19%	27%	39%	51
		Perp	81%	67%	56%	19%	28%	28%	0%	6%	17%	50%	67%	50%	68%	25%	7%	56%	52
	Pl	KNN	29%	8%	50%	13%	21%	8%	58%	71%	42%	88%	50%	63%	29%	14%	57%	67%	53
		Perp	8%	0%	13%	21%	25%	0%	71%	75%	88%	38%	13%	0%	7%	15%	78%	17%	54
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	

Table 130: B3. Relationship between graphs' width and protein types. Clustering algorithms analysis. Percentage.

			N			M			W			C			Total				
			SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	N	M	W	C	
NFP	DN	KNN	0%	67%	33%	50%	33%	0%	50%	0%	67%	0%	50%	100%	33%	28%	39%	50%	59
		Perp	100%	33%	33%	0%	67%	17%	0%	0%	50%	0%	50%	100%	56%	28%	17%	50%	60
	SP	KNN	42%	50%	67%	8%	17%	8%	50%	33%	25%	25%	25%	50%	53%	11%	36%	33%	61
		Perp	67%	42%	42%	8%	33%	33%	25%	25%	25%	25%	50%	25%	50%	25%	25%	33%	62
	SS	KNN	25%	67%	100%	33%	17%	0%	42%	17%	0%	50%	25%	0%	64%	17%	19%	25%	63
		Perp	33%	42%	50%	17%	33%	0%	50%	25%	50%	25%	25%	0%	42%	17%	42%	17%	64
IDP	DN	KNN	17%	50%	33%	50%	17%	0%	33%	33%	67%	100%	50%	100%	33%	22%	44%	83%	65
		Perp	83%	67%	0%	17%	17%	50%	0%	17%	50%	50%	100%	100%	50%	28%	22%	83%	66
	SP	KNN	17%	50%	33%	25%	0%	25%	58%	50%	42%	100%	0%	50%	33%	17%	50%	50%	67
		Perp	33%	33%	42%	50%	17%	8%	17%	50%	50%	75%	50%	0%	36%	25%	39%	42%	68
	SS	KNN	25%	33%	58%	8%	17%	17%	67%	50%	25%	75%	75%	75%	39%	14%	47%	75%	69
		Perp	33%	33%	42%	17%	8%	8%	50%	58%	50%	75%	25%	25%	36%	11%	53%	42%	70
Total	DN	KNN	8%	58%	33%	50%	25%	0%	42%	17%	67%	50%	50%	100%	33%	25%	42%	67%	71
		Perp	92%	50%	17%	8%	42%	33%	0%	8%	50%	25%	75%	100%	53%	28%	19%	67%	72
	SP	KNN	29%	50%	50%	17%	8%	17%	54%	42%	33%	63%	13%	50%	43%	14%	43%	42%	73
		Perp	50%	38%	42%	29%	25%	21%	21%	38%	38%	50%	50%	13%	43%	25%	32%	38%	74
	SS	KNN	25%	50%	79%	21%	17%	8%	54%	33%	13%	63%	50%	38%	51%	15%	33%	50%	75
		Perp	33%	38%	46%	17%	21%	4%	50%	42%	50%	50%	25%	13%	39%	14%	47%	29%	76
Total	NFP	KNN	27%	60%	73%	27%	20%	3%	47%	20%	23%	30%	30%	40%	53%	17%	30%	33%	77
		Perp	60%	40%	43%	10%	40%	17%	30%	20%	40%	20%	40%	30%	48%	22%	30%	30%	78
	IDP	KNN	20%	43%	43%	23%	10%	17%	57%	47%	40%	90%	40%	70%	36%	17%	48%	67%	79
		Perp	43%	40%	33%	30%	13%	17%	27%	47%	50%	70%	50%	30%	39%	20%	41%	50%	80
Total		KNN	23%	52%	58%	25%	15%	10%	52%	33%	32%	60%	35%	55%	44%	17%	39%	50%	81
		Perp	52%	40%	38%	20%	27%	17%	28%	33%	45%	45%	45%	30%	43%	21%	36%	40%	82
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	

Table 131: B4. Relationship between affinity types and protein types. Graph width analysis. Percentage.

			Entropic				Plain				Total				
			N	M	W	C	N	M	W	C	N	M	W	C	
NFP	SPC	KNN	33%	33%	33%	0%	17%	17%	67%	75%	27%	27%	47%	30%	87
		Perp	89%	11%	0%	17%	17%	8%	75%	25%	60%	10%	30%	20%	88
	SDS	KNN	89%	11%	0%	17%	17%	33%	50%	50%	60%	20%	20%	30%	89
		Perp	67%	33%	0%	67%	0%	50%	50%	0%	40%	40%	20%	40%	90
	SES	KNN	72%	6%	22%	50%	75%	0%	25%	25%	73%	3%	23%	40%	91
		Perp	56%	28%	17%	50%	25%	0%	75%	0%	43%	17%	40%	30%	92
IDP	SPC	KNN	6%	33%	61%	83%	42%	8%	50%	100%	20%	23%	57%	90%	93
		Perp	72%	28%	0%	83%	0%	33%	67%	50%	43%	30%	27%	70%	94
	SDS	KNN	72%	11%	17%	33%	0%	8%	92%	50%	43%	10%	47%	40%	95
		Perp	67%	22%	11%	67%	0%	0%	100%	25%	40%	13%	47%	50%	96
	SES	KNN	56%	17%	28%	50%	25%	17%	58%	100%	43%	17%	40%	70%	97
		Perp	56%	28%	17%	50%	0%	0%	100%	0%	33%	17%	50%	30%	98
Total	SPC	KNN	19%	33%	47%	42%	29%	13%	58%	88%	23%	25%	52%	60%	99
		Perp	81%	19%	0%	50%	8%	21%	71%	38%	52%	20%	28%	45%	100
	SDS	KNN	81%	11%	8%	25%	8%	21%	71%	50%	52%	15%	33%	35%	101
		Perp	67%	28%	6%	67%	0%	25%	75%	13%	40%	27%	33%	45%	102
	SES	KNN	64%	11%	25%	50%	50%	8%	42%	63%	58%	10%	32%	55%	103
		Perp	56%	28%	17%	50%	13%	0%	88%	0%	38%	17%	45%	30%	104
Total	NFP	KNN	65%	17%	19%	22%	36%	17%	47%	50%	53%	17%	30%	33%	105
		Perp	70%	24%	6%	44%	14%	19%	67%	8%	48%	22%	30%	30%	106
	IDP	KNN	44%	20%	35%	56%	22%	11%	67%	83%	36%	17%	48%	67%	107
		Perp	65%	26%	9%	67%	0%	11%	89%	25%	39%	20%	41%	50%	108
Total		KNN	55%	19%	27%	39%	29%	14%	57%	67%	44%	17%	39%	50%	109
		Perp	68%	25%	7%	56%	7%	15%	78%	17%	43%	21%	36%	40%	110
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	

Table 132: B4.1. Relationship between the data sparsity and protein types. Graph width analysis. Percentage.

			DN				SP				SS				Total				
			N	M	W	C	N	M	W	C	N	M	W	C	N	M	W	C	
NFP	SPC	KNN	0%	50%	50%	0%	42%	8%	50%	25%	25%	33%	42%	50%	27%	27%	47%	30%	115
		Perp	100%	0%	0%	0%	67%	8%	25%	25%	33%	17%	50%	25%	60%	10%	30%	20%	116
	SDS	KNN	67%	33%	0%	50%	50%	17%	33%	25%	67%	17%	17%	25%	60%	20%	20%	30%	117
		Perp	33%	67%	0%	50%	42%	33%	25%	50%	42%	33%	25%	25%	40%	40%	20%	40%	118
	SES	KNN	33%	0%	67%	100%	67%	8%	25%	50%	100%	0%	0%	0%	73%	3%	23%	40%	119
		Perp	33%	17%	50%	100%	42%	33%	25%	25%	50%	0%	50%	0%	43%	17%	40%	30%	120
IDP	SPC	KNN	17%	50%	33%	100%	17%	25%	58%	100%	25%	8%	67%	75%	20%	23%	57%	90%	121
		Perp	83%	17%	0%	50%	33%	50%	17%	75%	33%	17%	50%	75%	43%	30%	27%	70%	122
	SDS	KNN	50%	17%	33%	50%	50%	0%	50%	0%	33%	17%	50%	75%	43%	10%	47%	40%	123
		Perp	67%	17%	17%	100%	33%	17%	50%	50%	33%	8%	58%	25%	40%	13%	47%	50%	124
	SES	KNN	33%	0%	67%	100%	33%	25%	42%	50%	58%	17%	25%	75%	43%	17%	40%	70%	125
		Perp	0%	50%	50%	100%	42%	8%	50%	0%	42%	8%	50%	25%	33%	17%	50%	30%	126
Total	SPC	KNN	8%	50%	42%	50%	29%	17%	54%	63%	22%	28%	50%	58%	25%	23%	52%	60%	127
		Perp	92%	8%	0%	25%	50%	29%	21%	50%	64%	22%	14%	42%	58%	25%	17%	45%	128
	SDS	KNN	58%	25%	17%	50%	50%	8%	42%	13%	53%	14%	33%	25%	52%	12%	37%	20%	129
		Perp	50%	42%	8%	75%	38%	25%	38%	50%	42%	31%	28%	58%	40%	28%	32%	55%	130
	SES	KNN	33%	0%	67%	100%	50%	17%	33%	50%	44%	11%	44%	67%	47%	13%	40%	60%	131
		Perp	17%	33%	50%	100%	42%	21%	38%	13%	33%	25%	42%	42%	37%	23%	40%	30%	132
Total	NFP	KNN	33%	28%	39%	50%	53%	11%	36%	33%	46%	17%	37%	39%	49%	14%	37%	37%	133
		Perp	56%	28%	17%	50%	50%	25%	25%	33%	52%	26%	22%	39%	51%	26%	23%	37%	134
	IDP	KNN	33%	22%	44%	83%	33%	17%	50%	50%	33%	19%	48%	61%	33%	18%	49%	57%	135
		Perp	50%	28%	22%	83%	36%	25%	39%	42%	41%	26%	33%	56%	39%	26%	36%	50%	136
Total		KNN	33%	25%	42%	67%	43%	14%	43%	42%	40%	18%	43%	50%	41%	16%	43%	47%	137
		Perp	53%	28%	19%	67%	43%	25%	32%	38%	46%	26%	28%	47%	45%	26%	29%	43%	138
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	

Table 133: B5. Relationship between protein types and affinity types. Trend line direction analysis. Percentage.

			NFP					IDP					Total					
			/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	
Entropic	DN	KNN	28%	56%	17%	50%	17%	22%	67%	11%	50%	33%	25%	61%	14%	50%	25%	145
		Perp	11%	72%	17%	17%	0%	11%	39%	50%	17%	33%	11%	56%	33%	17%	17%	146
	SP	KNN	11%	89%	0%	0%	0%	17%	67%	17%	33%	0%	14%	78%	8%	17%	0%	147
		Perp	11%	72%	17%	33%	0%	6%	78%	17%	17%	17%	8%	75%	17%	25%	8%	148
	SS	KNN	0%	100%	0%	0%	0%	11%	83%	6%	0%	17%	6%	92%	3%	0%	8%	149
		Perp	6%	94%	0%	17%	0%	22%	67%	11%	33%	0%	14%	81%	6%	25%	0%	150
Plain	DN	KNN	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	151
		Perp	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	152
	SP	KNN	22%	61%	17%	50%	33%	28%	50%	22%	83%	33%	25%	56%	19%	67%	33%	153
		Perp	0%	94%	6%	0%	17%	0%	89%	11%	33%	33%	0%	92%	8%	17%	25%	154
	SS	KNN	39%	44%	17%	50%	83%	39%	33%	28%	83%	67%	39%	39%	22%	67%	75%	155
		Perp	0%	94%	6%	0%	0%	0%	89%	11%	33%	33%	0%	92%	8%	17%	17%	156
Total	DN	KNN	28%	56%	17%	50%	17%	22%	67%	11%	50%	33%	25%	61%	14%	50%	25%	157
		Perp	11%	72%	17%	17%	0%	11%	39%	50%	17%	33%	11%	56%	33%	17%	17%	158
	SP	KNN	17%	75%	8%	25%	17%	22%	58%	19%	58%	17%	19%	67%	14%	42%	17%	159
		Perp	6%	83%	11%	17%	8%	3%	83%	14%	25%	25%	4%	83%	13%	21%	17%	160
	SS	KNN	19%	72%	8%	25%	42%	25%	58%	17%	42%	42%	22%	65%	13%	33%	42%	161
		Perp	3%	94%	3%	8%	0%	11%	78%	11%	33%	17%	7%	86%	7%	21%	8%	162
Total	En	KNN	13%	81%	6%	17%	6%	17%	72%	11%	28%	17%	15%	77%	8%	22%	11%	163
		Perp	9%	80%	11%	22%	0%	13%	61%	26%	22%	17%	11%	70%	19%	22%	8%	164
	Pl	KNN	31%	53%	17%	50%	58%	33%	42%	25%	83%	50%	32%	47%	21%	67%	54%	165
		Perp	0%	94%	6%	0%	8%	0%	89%	11%	33%	33%	0%	92%	8%	17%	21%	166
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	

Table 134: B6. Relationship between graph trend direction and affinity types. Clustering algorithms analysis. Percentage.

			/			-			\			C			S			Toal					
			SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	/	-	\	C	S	
Entropic	DN	KNN	8%	17%	50%	83%	75%	25%	8%	8%	25%	0%	50%	100%	0%	0%	75%	25%	61%	14%	50%	25%	171
		Perp	8%	17%	8%	50%	75%	42%	42%	8%	50%	0%	25%	25%	25%	0%	25%	11%	56%	33%	17%	17%	172
	SP	KNN	0%	8%	33%	92%	83%	58%	8%	8%	8%	25%	25%	0%	0%	0%	0%	14%	78%	8%	17%	0%	173
		Perp	0%	8%	17%	67%	83%	75%	33%	8%	8%	25%	25%	25%	25%	0%	0%	8%	75%	17%	25%	8%	174
	SS	KNN	0%	0%	17%	100%	92%	83%	0%	8%	0%	0%	0%	0%	0%	25%	0%	6%	92%	3%	0%	8%	175
		Perp	25%	0%	17%	67%	92%	83%	8%	8%	0%	75%	0%	0%	0%	0%	0%	14%	81%	6%	25%	0%	176
Plain	DN	KNN	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	177
		Perp	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	178
	SP	KNN	8%	8%	58%	83%	42%	42%	8%	50%	0%	75%	100%	25%	0%	50%	50%	25%	56%	19%	67%	33%	179
		Perp	0%	0%	0%	100%	75%	100%	0%	25%	0%	25%	25%	0%	0%	75%	0%	0%	92%	8%	17%	25%	180
	SS	KNN	25%	0%	92%	58%	50%	8%	17%	50%	0%	100%	75%	25%	50%	75%	100%	39%	39%	22%	67%	75%	181
		Perp	0%	0%	0%	100%	75%	100%	0%	25%	0%	25%	25%	0%	0%	50%	0%	0%	92%	8%	17%	17%	182
Total	DN	KNN	8%	17%	50%	83%	75%	25%	8%	8%	25%	0%	50%	100%	0%	0%	75%	25%	61%	14%	50%	25%	183
		Perp	8%	17%	8%	50%	75%	42%	42%	8%	50%	0%	25%	25%	25%	0%	25%	11%	56%	33%	17%	17%	184
	SP	KNN	4%	8%	46%	88%	63%	50%	8%	29%	4%	50%	63%	13%	0%	25%	25%	19%	67%	14%	42%	17%	185
		Perp	0%	4%	8%	83%	79%	88%	17%	17%	4%	25%	25%	13%	13%	38%	0%	4%	83%	13%	21%	17%	186
	SS	KNN	13%	0%	54%	79%	71%	46%	8%	29%	0%	50%	38%	13%	25%	50%	50%	22%	65%	13%	33%	42%	187
		Perp	13%	0%	8%	83%	83%	92%	4%	17%	0%	50%	13%	0%	0%	25%	0%	7%	86%	7%	21%	8%	188
Total	En	KNN	3%	8%	33%	92%	83%	56%	6%	8%	11%	8%	25%	33%	0%	8%	25%	15%	77%	8%	22%	11%	189
		Perp	11%	8%	14%	61%	83%	67%	28%	8%	19%	33%	17%	17%	17%	0%	8%	11%	70%	19%	22%	8%	190
	Pl	KNN	17%	4%	75%	71%	46%	25%	13%	50%	0%	88%	88%	25%	25%	63%	75%	32%	47%	21%	67%	54%	191
		Perp	0%	0%	0%	100%	75%	100%	0%	25%	0%	25%	25%	0%	0%	63%	0%	0%	92%	8%	17%	21%	192
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	

Table 135: B7. Relationship between graph trend direction and protein types. Clustering algorithms analysis. Percentage.

			/			-			\			C			S			Total					
			SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	SPC	SDS	SES	/	-	\	C	S	
NFP	DN	KNN	17%	17%	50%	83%	67%	17%	0%	17%	33%	0%	50%	100%	0%	0%	50%	28%	56%	17%	50%	17%	197
		Perp	17%	0%	17%	67%	100%	50%	17%	0%	33%	0%	0%	50%	0%	0%	0%	11%	72%	17%	17%	0%	198
	SP	KNN	0%	8%	42%	100%	67%	58%	0%	25%	0%	25%	50%	0%	0%	25%	25%	17%	75%	8%	25%	17%	199
		Perp	0%	8%	8%	92%	75%	83%	8%	17%	8%	0%	25%	25%	0%	25%	0%	6%	83%	11%	17%	8%	200
	SS	KNN	17%	0%	42%	75%	83%	58%	8%	17%	0%	50%	25%	0%	50%	25%	50%	19%	72%	8%	25%	42%	201
		Perp	8%	0%	0%	92%	92%	100%	0%	8%	0%	25%	0%	0%	0%	0%	0%	3%	94%	3%	8%	0%	202
IDP	DN	KNN	0%	17%	50%	83%	83%	33%	17%	0%	17%	0%	50%	100%	0%	0%	100%	22%	67%	11%	50%	33%	203
		Perp	0%	33%	0%	33%	50%	33%	67%	17%	67%	0%	50%	0%	50%	0%	50%	11%	39%	50%	17%	33%	204
	SP	KNN	8%	8%	50%	75%	58%	42%	17%	33%	8%	75%	75%	25%	0%	25%	25%	22%	58%	19%	58%	17%	205
		Perp	0%	0%	8%	75%	83%	92%	25%	17%	0%	50%	25%	0%	25%	50%	0%	3%	83%	14%	25%	25%	206
	SS	KNN	8%	0%	67%	83%	58%	33%	8%	42%	0%	50%	50%	25%	0%	75%	50%	25%	58%	17%	42%	42%	207
		Perp	17%	0%	17%	75%	75%	83%	8%	25%	0%	75%	25%	0%	0%	50%	0%	11%	78%	11%	33%	17%	208
Total	DN	KNN	8%	17%	50%	83%	75%	25%	8%	8%	25%	0%	50%	100%	0%	0%	75%	25%	61%	14%	50%	25%	209
		Perp	8%	17%	8%	50%	75%	42%	42%	8%	50%	0%	25%	25%	25%	0%	25%	11%	56%	33%	17%	17%	210
	SP	KNN	4%	8%	46%	88%	63%	50%	8%	29%	4%	50%	63%	13%	0%	25%	25%	19%	67%	14%	42%	17%	211
		Perp	0%	4%	8%	83%	79%	88%	17%	17%	4%	25%	25%	13%	13%	38%	0%	4%	83%	13%	21%	17%	212
	SS	KNN	13%	0%	54%	79%	71%	46%	8%	29%	0%	50%	38%	13%	25%	50%	50%	22%	65%	13%	33%	42%	213
		Perp	13%	0%	8%	83%	83%	92%	4%	17%	0%	50%	13%	0%	0%	25%	0%	7%	86%	7%	21%	8%	214
Total	NFP	KNN	10%	7%	43%	87%	73%	50%	3%	20%	7%	30%	40%	20%	20%	20%	40%	20%	70%	10%	30%	27%	215
		Perp	7%	3%	7%	87%	87%	83%	7%	10%	10%	10%	10%	20%	0%	10%	0%	6%	86%	9%	13%	3%	216
	IDP	KNN	7%	7%	57%	80%	63%	37%	13%	30%	7%	50%	60%	40%	0%	40%	50%	23%	60%	17%	50%	30%	217
		Perp	7%	7%	10%	67%	73%	77%	27%	20%	13%	50%	30%	0%	20%	40%	10%	8%	72%	20%	27%	23%	218
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	

Table 136: B8. Relationship between clustering algorithms and protein types. Trend line direction analysis. Percentage.

			SPC					SDS					SES					Total					
			/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	
NFP	DN	KNN	17%	83%	0%	0%	0%	17%	67%	17%	50%	0%	50%	17%	33%	100%	50%	28%	56%	17%	50%	17%	223
		Perp	17%	67%	17%	0%	0%	0%	100%	0%	0%	0%	17%	50%	33%	50%	0%	11%	72%	17%	17%	0%	224
	SP	KNN	0%	100%	0%	25%	0%	8%	67%	25%	50%	25%	42%	58%	0%	0%	25%	17%	75%	8%	25%	17%	225
		Perp	0%	92%	8%	0%	0%	8%	75%	17%	25%	25%	8%	83%	8%	25%	0%	6%	83%	11%	17%	8%	226
	SS	KNN	17%	75%	8%	50%	50%	0%	83%	17%	25%	25%	42%	58%	0%	0%	50%	19%	72%	8%	25%	42%	227
		Perp	8%	92%	0%	25%	0%	0%	92%	8%	0%	0%	0%	100%	0%	0%	0%	3%	94%	3%	8%	0%	228
IDP	DN	KNN	0%	83%	17%	0%	0%	17%	83%	0%	50%	0%	50%	33%	17%	100%	100%	22%	67%	11%	50%	33%	229
		Perp	0%	33%	67%	0%	50%	33%	50%	17%	50%	0%	0%	33%	67%	0%	50%	11%	39%	50%	17%	33%	230
	SP	KNN	8%	75%	17%	75%	0%	8%	58%	33%	75%	25%	50%	42%	8%	25%	25%	22%	58%	19%	58%	17%	231
		Perp	0%	75%	25%	50%	25%	0%	83%	17%	25%	50%	8%	92%	0%	0%	0%	3%	83%	14%	25%	25%	232
	SS	KNN	8%	83%	8%	50%	0%	0%	58%	42%	50%	75%	67%	33%	0%	25%	50%	25%	58%	17%	42%	42%	233
		Perp	17%	75%	8%	75%	0%	0%	75%	25%	25%	50%	17%	83%	0%	0%	0%	11%	78%	11%	33%	17%	234
Total	DN	KNN	8%	83%	8%	0%	0%	17%	75%	8%	50%	0%	50%	25%	25%	100%	75%	25%	61%	14%	50%	25%	235
		Perp	8%	50%	42%	0%	25%	17%	75%	8%	25%	0%	8%	42%	50%	25%	25%	11%	56%	33%	17%	17%	236
	SP	KNN	4%	88%	8%	50%	0%	8%	63%	29%	63%	25%	46%	50%	4%	13%	25%	19%	67%	14%	42%	17%	237
		Perp	0%	83%	17%	25%	13%	4%	79%	17%	25%	38%	8%	88%	4%	13%	0%	4%	83%	13%	21%	17%	238
	SS	KNN	13%	79%	8%	50%	25%	0%	71%	29%	38%	50%	54%	46%	0%	13%	50%	22%	65%	13%	33%	42%	239
		Perp	13%	83%	4%	50%	0%	0%	83%	17%	13%	25%	8%	92%	0%	0%	0%	7%	86%	7%	21%	8%	240
Total	NFP	KNN	10%	87%	3%	30%	20%	7%	73%	20%	40%	20%	43%	50%	7%	20%	40%	20%	70%	10%	30%	27%	241
		Perp	7%	87%	7%	10%	0%	3%	87%	10%	10%	10%	7%	83%	10%	20%	0%	6%	86%	9%	13%	3%	242
	IDP	KNN	7%	80%	13%	50%	0%	7%	63%	30%	60%	40%	57%	37%	7%	40%	50%	23%	60%	17%	50%	30%	243
		Perp	7%	67%	27%	50%	20%	7%	73%	20%	30%	40%	10%	77%	13%	0%	10%	8%	72%	20%	27%	23%	244
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	

Table 137: B9. Relationship between affinity types and protein types. Trend line direction analysis. Percentage.

			Entropic					Plain					Total					
			/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	
NFP	SPC	KNN	6%	94%	0%	0%	0%	17%	75%	8%	75%	50%	10%	87%	3%	30%	20%	249
		Perp	11%	78%	11%	17%	0%	0%	100%	0%	0%	0%	7%	87%	7%	10%	0%	250
	SDS	KNN	6%	89%	6%	17%	0%	8%	50%	42%	75%	50%	7%	73%	20%	40%	20%	251
		Perp	6%	89%	6%	17%	0%	0%	83%	17%	0%	25%	3%	87%	10%	10%	10%	252
	SES	KNN	28%	61%	11%	33%	17%	67%	33%	0%	0%	75%	43%	50%	7%	20%	40%	253
		Perp	11%	72%	17%	33%	0%	0%	100%	0%	0%	0%	7%	83%	10%	20%	0%	254
IDP	SPC	KNN	0%	89%	11%	17%	0%	17%	67%	17%	100%	0%	7%	80%	13%	50%	0%	255
		Perp	11%	44%	44%	50%	33%	0%	100%	0%	50%	0%	7%	67%	27%	50%	20%	256
	SDS	KNN	11%	78%	11%	33%	17%	0%	42%	58%	100%	75%	7%	63%	30%	60%	40%	257
		Perp	11%	78%	11%	17%	0%	0%	67%	33%	50%	100%	7%	73%	20%	30%	40%	258
	SES	KNN	39%	50%	11%	33%	33%	83%	17%	0%	50%	75%	57%	37%	7%	40%	50%	259
		Perp	17%	61%	22%	0%	17%	0%	100%	0%	0%	0%	10%	77%	13%	0%	10%	260
Total	SPC	KNN	3%	92%	6%	8%	0%	17%	71%	13%	88%	25%	8%	83%	8%	40%	10%	261
		Perp	11%	61%	28%	33%	17%	0%	100%	0%	25%	0%	7%	77%	17%	30%	10%	262
	SDS	KNN	8%	83%	8%	25%	8%	4%	46%	50%	88%	63%	7%	68%	25%	50%	30%	263
		Perp	8%	83%	8%	17%	0%	0%	75%	25%	25%	63%	5%	80%	15%	20%	25%	264
	SES	KNN	33%	56%	11%	33%	25%	75%	25%	0%	25%	75%	50%	43%	7%	30%	45%	265
		Perp	14%	67%	19%	17%	8%	0%	100%	0%	0%	0%	8%	80%	12%	10%	5%	266
Total	NFP	KNN	13%	81%	6%	17%	6%	31%	53%	17%	50%	58%	20%	70%	10%	30%	27%	267
		Perp	9%	80%	11%	22%	0%	0%	94%	6%	0%	8%	6%	86%	9%	13%	3%	268
	IDP	KNN	17%	72%	11%	28%	17%	33%	42%	25%	83%	50%	23%	60%	17%	50%	30%	269
		Perp	13%	61%	26%	22%	17%	0%	89%	11%	33%	33%	8%	72%	20%	27%	23%	270
Total		KNN	15%	77%	8%	22%	11%	32%	47%	21%	67%	54%	22%	65%	13%	40%	28%	271
		Perp	11%	70%	19%	22%	8%	0%	92%	8%	17%	21%	7%	79%	14%	20%	13%	272
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	

Table 138: B10. Relationship between data sparsity and protein types. Trend line direction analysis. Percentage.

			DN					SP					SS					Total					
			/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	/	-	\	C	S	
NFP	SPC	KNN	17%	83%	0%	0%	0%	0%	100%	0%	25%	0%	17%	75%	8%	50%	50%	10%	87%	3%	30%	20%	277
		Perp	17%	67%	17%	0%	0%	0%	92%	8%	0%	0%	8%	92%	0%	25%	0%	7%	87%	7%	10%	0%	278
	SDS	KNN	17%	67%	17%	50%	0%	8%	67%	25%	50%	25%	0%	83%	17%	25%	25%	7%	73%	20%	40%	20%	279
		Perp	0%	100%	0%	0%	0%	8%	75%	17%	25%	25%	0%	92%	8%	0%	0%	3%	87%	10%	10%	10%	280
	SES	KNN	50%	17%	33%	100%	50%	42%	58%	0%	0%	25%	42%	58%	0%	0%	50%	43%	50%	7%	20%	40%	281
		Perp	17%	50%	33%	50%	0%	8%	83%	8%	25%	0%	0%	100%	0%	0%	0%	7%	83%	10%	20%	0%	282
IDP	SPC	KNN	0%	83%	17%	0%	0%	8%	75%	17%	75%	0%	8%	83%	8%	50%	0%	7%	80%	13%	50%	0%	283
		Perp	0%	33%	67%	0%	50%	0%	75%	25%	50%	25%	17%	75%	8%	75%	0%	7%	67%	27%	50%	20%	284
	SDS	KNN	17%	83%	0%	50%	0%	8%	58%	33%	75%	25%	0%	58%	42%	50%	75%	7%	63%	30%	60%	40%	285
		Perp	33%	50%	17%	50%	0%	0%	83%	17%	25%	50%	0%	75%	25%	25%	50%	7%	73%	20%	30%	40%	286
	SES	KNN	50%	33%	17%	100%	100%	50%	42%	8%	25%	25%	67%	33%	0%	25%	50%	57%	37%	7%	40%	50%	287
		Perp	0%	33%	67%	0%	50%	8%	92%	0%	0%	0%	17%	83%	0%	0%	0%	10%	77%	13%	0%	10%	288
Total	SPC	KNN	8%	83%	8%	0%	0%	4%	88%	8%	50%	0%	13%	79%	8%	50%	25%	8%	83%	8%	40%	10%	289
		Perp	8%	50%	42%	0%	25%	0%	83%	17%	25%	13%	13%	83%	4%	50%	0%	7%	77%	17%	30%	10%	290
	SDS	KNN	17%	75%	8%	50%	0%	8%	63%	29%	63%	25%	0%	71%	29%	38%	50%	7%	68%	25%	50%	30%	291
		Perp	17%	75%	8%	25%	0%	4%	79%	17%	25%	38%	0%	83%	17%	13%	25%	5%	80%	15%	20%	25%	292
	SES	KNN	50%	25%	25%	100%	75%	46%	50%	4%	13%	25%	54%	46%	0%	13%	50%	50%	43%	7%	30%	45%	293
		Perp	8%	42%	50%	25%	25%	8%	88%	4%	13%	0%	8%	92%	0%	0%	0%	8%	80%	12%	10%	5%	294
Total	NFP	KNN	28%	56%	17%	50%	17%	17%	75%	8%	25%	17%	19%	72%	8%	25%	42%	20%	70%	10%	30%	27%	295
		Perp	11%	72%	17%	17%	0%	6%	83%	11%	17%	8%	3%	94%	3%	8%	0%	6%	86%	9%	13%	3%	296
	IDP	KNN	22%	67%	11%	50%	33%	22%	58%	19%	58%	17%	25%	58%	17%	42%	42%	23%	60%	17%	50%	30%	297
		Perp	11%	39%	50%	17%	33%	3%	83%	14%	25%	25%	11%	78%	11%	33%	17%	8%	72%	20%	27%	23%	298
Total		KNN	25%	61%	14%	50%	25%	19%	67%	14%	42%	17%	22%	65%	13%	33%	42%	22%	65%	13%	40%	28%	299
		Perp	11%	56%	33%	17%	17%	4%	83%	13%	21%	17%	7%	86%	7%	21%	8%	7%	79%	14%	20%	13%	300
			AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	

APPENDIX C

Detailed Results for Chapter III

C.I **Protein type**

Affinities

Width

EN

For KNN NFP showed 65% [AC23] of narrow shapes, 19% [AE23] of wide shapes, and 22% [AF23] of changes. IDP showed 44% [AG23] of narrow shapes, 35% [AI23] of wide shapes, and 56% [AJ23] of changes.

For pp NFP showed 70% [AC24] of narrow shapes, 6% [AE24] of wide shapes, and 44% [AF24] of changes. IDP showed 65% [AG24] of narrow shapes, 9% [AI24] of wide shapes, and 67% [AJ24] of changes.

PL

For KNN NFP showed 36% [AC25] of narrow shapes, 47% [AE25] of wide shapes, and 50% [AF25] of changes. IDP showed 22% [AG25] of narrow shapes, 67% [AI25] of

wide shapes, and 83% [AJ25] of changes.

For sigma NFP showed 14% [AC26] of narrow shapes, 67% [AE26] of wide shapes, and 8% [AF26] of changes. IDP showed 0% [AG26] of narrow shapes, 89% [AI26] of wide shapes, and 25% [AJ26] of changes.

Total

For KNN NFP showed 53% [AC27] of narrow shapes, 30% [AE27] of wide shapes, and 33% [AF27] of changes. IDP showed 36% [AG27] of narrow shapes, 48% [AI27] of wide shapes, and 67% [AJ27] of changes.

For pp/sigma NFP showed 48% [AC28] of narrow shapes, 30% [AE28] of wide shapes, and 30% [AF28] of changes. IDP showed 39% [AG28] of narrow shapes, 41% [AI28] of wide shapes, and 50% [AJ28] of changes.

Shape

EN

For KNN NFP showed 13% [AC163] of rising parts, 6% [AE163] of falling parts, 6% [AG163]of them were strong, and 17% [AF163] of V and A shapes. IDP showed 17% [AC163] of rising parts, 11% [AE163] of falling parts, 17% [AG163]of them were strong, and 28% [AF163] of V and A shapes.

For pp NFP showed 9% [AC164] of rising parts, 11% [AE164] of falling parts, 0% [AG164]of them were strong, and 22% [AF164] of V and A shapes. IDP showed 13% [AC164] of rising parts, 26% [AE164] of falling parts, 17% [AG164]of them were strong, and 22% [AF164] of V and A shapes.

PL

For KNN NFP showed 31% [AC163] of rising parts, 17% [AE163] of falling parts, 58% [AG163]of them were strong, and 50% [AF163] of V and A shapes. IDP showed 33% [AC163] of rising parts, 25% [AE163] of falling parts, 50% [AG163]of them were strong, and 83% [AF163] of V and A shapes.

For sigma NFP showed 0% [AC164] of rising parts, 6% [AE164] of falling parts, 8% [AG164]of them were strong, and 0% [AF164] of V and A shapes. IDP showed 0% [AC164] of rising parts, 11% [AE164] of falling parts, 33% [AG164]of them were strong, and 33% [AF164] of V and A shapes.

Total

For KNN NFP showed 31% [AC163] of rising parts, 17% [AE163] of falling parts, 58% [AG163]of them were strong, and 50% [AF163] of V and A shapes. IDP showed 33% [AC163] of rising parts, 25% [AE163] of falling parts, 50% [AG163]of them were strong, and 83% [AF163] of V and A shapes.

For pp/sigma NFP showed 0% [AC164] of rising parts, 6% [AE164] of falling parts, 8% [AG164]of them were strong, and 0% [AF164] of V and A shapes. IDP showed 0%

[AC164] of rising parts, 11% [AE164] of falling parts, 33% [AG164]of them were strong, and 33% [AF164] of V and A shapes.

Algorithms

Width

SPC

For KNN NFP showed 27% [AK87] of narrow shapes, 47% [AM87] of wide shapes, and 30% [AF91] of changes. IDP showed 20% [AK93] of narrow shapes, 57% [AM93] of wide shapes, and 90% [AN93] of changes.

For pp/sigma NFP showed 60% [AK88] of narrow shapes, 30% [AM88] of wide shapes, and 20% [AN88] of changes. IDP showed 43% [AK94] of narrow shapes, 27% [AM94] of wide shapes, and 70% [AN94] of changes.

SDS

For KNN NFP showed 60% [AK89] of narrow shapes, 20% [AK89] of wide shapes, and 30% [AN89] of changes. IDP showed 43% [AC95] of narrow shapes, 47% [AM95] of wide shapes, and 40% [AN95] of changes.

For pp/sigma NFP showed 40% [AK90] of narrow shapes, 20% [AM90] of wide shapes, and 40% [AN90] of changes. IDP showed 40% [AK96] of narrow shapes, 47% [AM96] of wide shapes, and 50% [AN96] of changes.

SES

For KNN NFP showed 73% [AK91] of narrow shapes, 23% [AM91] of wide shapes, and 40% [AN91] of changes. IDP showed 43% [AK97] of narrow shapes, 40% [AM97] of wide shapes, and 70% [AN97] of changes.

For pp/sigma NFP showed 43% [AK92] of narrow shapes, 40% [AM92] of wide shapes, and 30% [AN92] of changes. IDP showed 33% [AK98] of narrow shapes, 50% [AM98] of wide shapes, and 30% [AN98] of changes.

Total

For KNN NFP showed 53% [AK105] of narrow shapes, 30% [AM105] of wide shapes, and 33% [AN105] of changes. IDP showed 36% [AK107] of narrow shapes, 48% [AM107] of wide shapes, and 67% [AN107] of changes.

For pp/sigma NFP showed 48% [AK106] of narrow shapes, 30% [AM106] of wide shapes, and 30% [AN106] of changes. IDP showed 39% [AK108] of narrow shapes, 41% [AM108] of wide shapes, and 50% [AN108] of changes.

Shape

SPC

For KNN NFP showed 10% [AC241] of rising parts, 3% [AE241] of falling parts, 20% [AG241]of them were strong, and 30% [AF241] of V and A shapes. IDP showed 7% [AC243] of rising parts, 13% [AE243] of falling parts, 0% [AG243]of them were strong,

and 50% [AF243] of V and A shapes.

For pp/sigma NFP showed 7% [AC242] of rising parts, 7% [AE242] of falling parts, 0% [AG242]of them were strong, and 10% [AF242] of V and A shapes. IDP showed 7% [AC244] of rising parts, 27% [AE244] of falling parts, 20% [AG244]of them were strong, and 50% [AF244] of V and A shapes.

SDS

For KNN NFP showed 7% [AH241] of rising parts, 20% [AJ241] of falling parts, 20% [AL241]of them were strong, and 40% [AK241] of V and A shapes. IDP showed 7% [AH243] of rising parts, 30% [AJ243] of falling parts, 40% [AL243]of them were strong, and 60% [AK243] of V and A shapes.

For pp/sigma NFP showed 3% [AH242] of rising parts, 10% [AJ242] of falling parts, 10% [AL242]of them were strong, and 10% [AK242] of V and A shapes. IDP showed 7% [AH244] of rising parts, 20% [AJ244] of falling parts, 40% [AL244]of them were strong, and 30% [AK244] of V and A shapes.

SES

For KNN NFP showed 43% [AM241] of rising parts, 7% [AO241] of falling parts, 40% [AQ241]of them were strong, and 20% [AP241] of V and A shapes. IDP showed 57% [AM243] of rising parts, 7% [AO243] of falling parts, 50% [AQ243]of them were strong, and 40% [AP243] of V and A shapes.

For pp/sigma NFP showed 7% [AM242] of rising parts, 10% [AO242] of falling parts, 0% [AQ242]of them were strong, and 20% [AP242] of V and A shapes. IDP showed 10% [AM244] of rising parts, 13% [AO244] of falling parts, 10% [AQ244]of them were strong, and 0% [AP244] of V and A shapes.

Total

For KNN NFP showed 20% [AR241] of rising parts, 10% [AT241] of falling parts, 27% [AV241]of them were strong, and 30% [AU241] of V and A shapes. IDP showed 57% [AR243] of rising parts, 7% [AT243] of falling parts, 50% [AV243]of them were strong, and 40% [AU243] of V and A shapes.

For pp/sigma NFP showed 6% [AR242] of rising parts, 9% [AT242] of falling parts, 3% [AV242]of them were strong, and 13% [AU242] of V and A shapes. IDP showed 8% [AR244] of rising parts, 20% [AT244] of falling parts, 23% [AV244]of them were strong, and 27% [AU244] of V and A shapes.

Sparsity

Width

DN

For KNN NFP showed 33% [AO59] of narrow shapes, 39% [AQ59] of wide shapes, and 50% [AR59] of changes. IDP showed 33% [AO65] of narrow shapes, 44% [AQ65] of wide shapes, and 83% [AR65] of changes.

For pp NFP showed 56% [AO60] of narrow shapes, 17% [AQ60] of wide shapes, and 50% [AR60] of changes. IDP showed 50% [AO66] of narrow shapes, 22% [AQ66] of wide shapes, and 83% [AR66] of changes.

SP

For KNN NFP showed 53% [AO61] of narrow shapes, 36% [AQ61] of wide shapes, and 33% [AR61] of changes. IDP showed 33% [AO67] of narrow shapes, 50% [AQ67] of wide shapes, and 50% [AR67] of changes.

For pp/sigma NFP showed 50% [AO62] of narrow shapes, 25% [AQ62] of wide shapes, and 33% [AR62] of changes. IDP showed 36% [AO68] of narrow shapes, 39% [AQ68] of wide shapes, and 42% [AR68] of changes.

SS

For KNN NFP showed 64% [AO63] of narrow shapes, 19% [AQ63] of wide shapes, and 25% [AR63] of changes. IDP showed 39% [AO69] of narrow shapes, 47% [AQ69] of wide shapes, and 75% [AR69] of changes.

For pp/sigma NFP showed 42% [AO64] of narrow shapes, 42% [AQ64] of wide shapes, and 17% [AR64] of changes. IDP showed 36% [AO70] of narrow shapes, 53% [AQ70] of wide shapes, and 42% [AR70] of changes.

Total

For KNN NFP showed 53% [AO77] of narrow shapes, 30% [AQ77] of wide shapes, and 33% [AR77] of changes. IDP showed 36% [AO79] of narrow shapes, 48% [AQ79] of wide shapes, and 67% [AR79] of changes.

For pp/sigma NFP showed 48% [AO78] of narrow shapes, 30% [AQ78] of wide shapes, and 30% [AR78] of changes. IDP showed 39% [AO80] of narrow shapes, 41% [AQ80] of wide shapes, and 50% [AR80] of changes.

Shape

DN

For KNN NFP showed 28% [AR223] of rising parts, 17% [AT223] of falling parts, 17% [AV223]of them were strong, and 50% [AU223] of V and A shapes. IDP showed 22% [AR229] of rising parts, 11% [AT229] of falling parts, 33% [AV229]of them were strong, and 50% [AU229] of V and A shapes.

For pp NFP showed 11% [AR224] of rising parts, 17% [AT224] of falling parts, 0% [AV224]of them were strong, and 17% [AU224] of V and A shapes. IDP showed 11% [AR230] of rising parts, 50% [AT230] of falling parts, 33% [AV230]of them were strong, and 17% [AU230] of V and A shapes.

SP

For KNN NFP showed 17% [AR225] of rising parts, 8% [AT225] of falling parts, 17% [AV225]of them were strong, and 25% [AU225] of V and A shapes. IDP showed 22% [AR231] of rising parts, 19% [AT231] of falling parts, 17% [AV231]of them were

strong, and 58% [AU231] of V and A shapes.

For pp/sigma NFP showed 6% [AR226] of rising parts, 11% [AT226] of falling parts, 17% [AV226]of them were strong, and 17% [AU226] of V and A shapes. IDP showed 3% [AR232] of rising parts, 14% [AT232] of falling parts, 25% [AV232]of them were strong, and 25% [AU232] of V and A shapes.

SS

For KNN NFP showed 19% [AR227] of rising parts, 8% [AT227] of falling parts, 42% [AV227]of them were strong, and 25% [AU227] of V and A shapes. IDP showed 25% [AR233] of rising parts, 17% [AT233] of falling parts, 42% [AV233]of them were strong, and 42% [AU233] of V and A shapes.

For pp/sigma NFP showed 3% [AR228] of rising parts, 3% [AT228] of falling parts, 0% [AV228]of them were strong, and 8% [AU228] of V and A shapes. IDP showed 11% [AR234] of rising parts, 11% [AT234] of falling parts, 17% [AV234]of them were strong, and 33% [AU234] of V and A shapes.

Total

For KNN NFP showed 20% [AR241] of rising parts, 10% [AT241] of falling parts, 27% [AV241]of them were strong, and 30% [AU241] of V and A shapes. IDP showed 23% [AR243] of rising parts, 17% [AT243] of falling parts, 30% [AV243]of them were strong, and 50% [AU243] of V and A shapes.

For pp/sigma NFP showed 6% [AR242] of rising parts, 9% [AT242] of falling parts, 3% [AV242]of them were strong, and 13% [AU242] of V and A shapes. IDP showed 8% [AR244] of rising parts, 20% [AT244] of falling parts, 23% [AV244]of them were strong, and 27% [AU244] of V and A shapes.

C.II **Affinity type**

Protein type

Width

NFP

For KNN Entropic affinities showed 65% [AC23] of narrow shapes, 19% [AE23] of wide shapes, and 22% [AF23] of changes. Plain affinities showed 36% [AC25] of narrow shapes, 47% [AE25] of wide shapes, and 50% [AF25] of changes.

For pp/sigma Entropic affinities showed 70% [AC24] of narrow shapes, 6% [AE24] of wide shapes, and 44% [AF24] of changes. Plain affinities showed 14% [AC26] of narrow shapes, 67% [AE26] of wide shapes, and 8% [AF26] of changes.

IDP

For KNN Entropic affinities showed 44% [AG23] of narrow shapes, 35% [AI23] of wide shapes, and 56% [AJ23] of changes. Plain affinities showed 22% [AG25] of narrow shapes, 67% [AI25] of wide shapes, and 83% [AJ25] of changes.

For pp/sigma Entropic affinities showed 65% [AG24] of narrow shapes, 9% [AI24] of wide shapes, and 67% [AJ24] of changes. Plain affinities showed 0% [AG26] of narrow shapes, 89% [AI26] of wide shapes, and 25% [AJ26] of changes.

Total

For KNN Entropic affinities showed 55% [AK23] of narrow shapes, 27% [AM23] of wide shapes, and 39% [AN23] of changes. Plain affinities showed 29% [AK25] of narrow shapes, 57% [AM25] of wide shapes, and 67% [AN25] of changes.

For pp/sigma Entropic affinities showed 68% [AK24] of narrow shapes, 7% [AM24] of wide shapes, and 56% [AN24] of changes. Plain affinities showed 7% [AK26] of narrow shapes, 78% [AM26] of wide shapes, and 17% [AN26] of changes.

Shape

NFP

For KNN Entropic affinities showed 13% [AC267] of rising parts, 6% [AE267] of falling parts, 6% [AG267]of them were strong, and 17% [AF267] of V and A shapes. Plain affinities showed 31% [AH267] of rising parts, 17% [AJ267] of falling parts, 58% [AL267]of them were strong, and 50% [AK267] of V and A shapes.

For pp/sigma Entropic affinities showed 9% [AC268] of rising parts, 11% [AE268] of falling parts, 0% [AG268]of them were strong, and 22% [AF268] of V and A shapes. Plain affinities showed 0% [AH268] of rising parts, 6% [AJ268] of falling parts, 8% [AL268]of them were strong, and 0% [AK268] of V and A shapes.

IDP

For KNN Entropic affinities showed 17% [AC269] of rising parts, 11% [AE269] of falling parts, 17% [AG269]of them were strong, and 28% [AF269] of V and A shapes. Plain affinities showed 33% [AH269] of rising parts, 25% [AJ269] of falling parts, 50% [AL269]of them were strong, and 83% [AK269] of V and A shapes.

For pp/sigma Entropic affinities showed 13% [AC270] of rising parts, 26% [AE270] of falling parts, 17% [AG270]of them were strong, and 22% [AF270] of V and A shapes. Plain affinities showed 0% [AH270] of rising parts, 11% [AJ270] of falling parts, 33% [AL270]of them were strong, and 33% [AK270] of V and A shapes.

Total

For KNN Entropic affinities showed 15% [AC271] of rising parts, 8% [AE271] of falling parts, 11% [AG271]of them were strong, and 22% [AF271] of V and A shapes. Plain affinities showed 32% [AH271] of rising parts, 21% [AJ271] of falling parts, 54% [AL271]of them were strong, and 67% [AK271] of V and A shapes.

For pp/sigma Entropic affinities showed 11% [AC272] of rising parts, 19% [AE272] of falling parts, 8% [AG272]of them were strong, and 22% [AF272] of V and A shapes. Plain affinities showed 0% [AH272] of rising parts, 8% [AJ272] of falling parts, 21% [AL272]of them were strong, and 17% [AK272] of V and A shapes.

Algorithms

Width

SPC

For KNN Entropic affinities showed 19% [AC99] of narrow shapes, 47% [AE99] of wide shapes, and 42% [AF99] of changes. Plain affinities showed 29% [AF99] of narrow shapes, 58% [AI99] of wide shapes, and 88% [AJ99] of changes.

For pp/sigma Entropic affinities showed 81% [AC100] of narrow shapes, 0% [AE100] of wide shapes, and 50% [AF100] of changes. Plain affinities showed 8% [AG100] of narrow shapes, 71% [AI100] of wide shapes, and 38% [AJ100] of changes.

SDS

For KNN Entropic affinities showed 81% [AC101] of narrow shapes, 8% [AE101] of wide shapes, and 25% [AF101] of changes. Plain affinities showed 8% [AF101] of narrow shapes, 71% [AI101] of wide shapes, and 50% [AJ101] of changes.

For pp/sigma Entropic affinities showed 67% [AC102] of narrow shapes, 6% [AE101] of wide shapes, and 67% [AF102] of changes. Plain affinities showed 0% [AG102] of narrow shapes, 75% [AI102] of wide shapes, and 13% [AJ102] of changes.

SES

For KNN Entropic affinities showed 64% [AC103] of narrow shapes, 25% [AE103] of wide shapes, and 50% [AF103] of changes. Plain affinities showed 50% [AF103] of

narrow shapes, 42% [AI103] of wide shapes, and 63% [AJ103] of changes.

For pp/sigma Entropic affinities showed 56% [AC104] of narrow shapes, 17% [AE104] of wide shapes, and 50% [AF104] of changes. Plain affinities showed 12.5% [AG104] of narrow shapes, 87.5% [AI100] of wide shapes, and 0% [AJ100] of changes.

Total

For KNN Entropic affinities showed 55% [AC109] of narrow shapes, 27% [AE109] of wide shapes, and 39% [AF109] of changes. Plain affinities showed 29% [AF109] of narrow shapes, 57% [AI109] of wide shapes, and 67% [AJ109] of changes.

For pp/sigma Entropic affinities showed 68% [AC109] of narrow shapes, 7% [AE109] of wide shapes, and 56% [AF109] of changes. Plain affinities showed 7% [AG109] of narrow shapes, 78% [AI109] of wide shapes, and 17% [AJ109] of changes.

Shape

SPC

For KNN Entropic affinities showed 3% [AC261] of rising parts, 6% [AE261] of falling parts, 0% [AG261]of them were strong, and 8% [AF261] of V and A shapes. Plain affinities showed 17% [AH261] of rising parts, 13% [AJ261] of falling parts, 25% [AL261]of them were strong, and 88% [AK261] of V and A shapes.

For pp/sigma Entropic affinities showed 11% [AC262] of rising parts, 28% [AE262] of falling parts, 17% [AG262]of them were strong, and 33% [AF262] of V and A shapes. Plain affinities showed 0% [AH262] of rising parts, 0% [AJ262] of falling parts, 0% [AL262]of them were strong, and 25% [AK262] of V and A shapes.

SDS

For KNN Entropic affinities showed 8% [AC263] of rising parts, 8% [AE263] of falling parts, 8% [AG263]of them were strong, and 25% [AF263] of V and A shapes. Plain affinities showed 4% [AH263] of rising parts, 50% [AJ263] of falling parts, 63% [AL263]of them were strong, and 88% [AK263] of V and A shapes.

For pp/sigma Entropic affinities showed 8% [AC264] of rising parts, 8% [AE264] of falling parts, 0% [AG264]of them were strong, and 17% [AF264] of V and A shapes. Plain affinities showed 0% [AH264] of rising parts, 25% [AJ264] of falling parts, 63% [AL264]of them were strong, and 25% [AK264] of V and A shapes.

SES

For KNN Entropic affinities showed 33% [AC265] of rising parts, 11% [AE265] of falling parts, 25% [AG265]of them were strong, and 33% [AF265] of V and A shapes. Plain affinities showed 75% [AH265] of rising parts, 0% [AJ265] of falling parts, 75% [AL265]of them were strong, and 25% [AK265] of V and A shapes.

For pp/sigma Entropic affinities showed 14% [AC266] of rising parts, 19% [AE266] of falling parts, 8% [AG266]of them were strong, and 17% [AF266] of V and A shapes.

Plain affinities showed 0% [AH266] of rising parts, 0% [AJ266] of falling parts, 0% [AL266]of them were strong, and 0% [AK266] of V and A shapes.

Total

For KNN Entropic affinities showed 15% [AC271] of rising parts, 8% [AE271] of falling parts, 11% [AG271]of them were strong, and 22% [AF271] of V and A shapes. Plain affinities showed 32% [AH271] of rising parts, 21% [AJ271] of falling parts, 54% [AL271]of them were strong, and 67% [AK271] of V and A shapes.

For pp/sigma Entropic affinities showed 11% [AC272] of rising parts, 19% [AE272] of falling parts, 8% [AG272]of them were strong, and 22% [AF272] of V and A shapes. Plain affinities showed 0% [AH272] of rising parts, 8% [AJ272] of falling parts, 21% [AL272]of them were strong, and 17% [AK272] of V and A shapes.

Data sparsity

Width

DN

For KNN Entropic affinities showed 33% [AK5] of narrow shapes, 42% [AM5] of wide shapes, and 67% [AN5] of changes.

For pp Entropic affinities showed 53% [AK6] of narrow shapes, 19% [AM6] of wide shapes, and 67% [AN6] of changes.

SP

For KNN Entropic affinities showed 69% [AK7] of narrow shapes, 8% [AM7] of wide shapes, and 25% [AN7] of changes. Plain affinities showed 17% [AK13] of narrow shapes, 78% [AM13] of wide shapes, and 58% [AN13] of changes.

For pp/sigma Entropic affinities showed 72% [AK8] of narrow shapes, 0% [AM8] of wide shapes, and 50% [AN8] of changes. Plain affinities showed 14% [AK14] of narrow shapes, 64% [AM14] of wide shapes, and 25% [AN14] of changes.

SS

For KNN Entropic affinities showed 61% [AK9] of narrow shapes, 31% [AM9] of wide shapes, and 25% [AN9] of changes. Plain affinities showed 42% [AK15] of narrow shapes, 36% [AM15] of wide shapes, and 75% [AN15] of changes.

For pp/sigma Entropic affinities showed 78% [AK10] of narrow shapes, 3% [AM10] of wide shapes, and 50% [AN10] of changes. Plain affinities showed 0% [AK16] of narrow shapes, 92% [AM16] of wide shapes, and 8% [AN16] of changes.

Total

For KNN Entropic affinities showed 55% [AK23] of narrow shapes, 27% [AM23] of wide shapes, and 39% [AN23] of changes. Plain affinities showed 29% [AK25] of narrow shapes, 57% [AM25] of wide shapes, and 67% [AN25] of changes.

For pp/sigma Entropic affinities showed 68% [AK24] of narrow shapes, 7% [AM24] of wide shapes, and 56% [AN24] of changes. Plain affinities showed 7% [AK26] of narrow shapes, 78% [AM26] of wide shapes, and 17% [AN26] of changes.

Shape

DN

For KNN Entropic affinities showed 25% [AM145] of rising parts, 14% [AO145] of falling parts, 25% [AQ145]of them were strong, and 50% [AP145] of V and A shapes.

For pp Entropic affinities showed 11% [AM146] of rising parts, 33% [AO146] of falling parts, 17% [AQ146]of them were strong, and 17% [AP146] of V and A shapes.

SP

For KNN Entropic affinities showed 14% [AM147] of rising parts, 8% [AO147] of falling parts, 0% [AQ147]of them were strong, and 17% [AP147] of V and A shapes. Plain affinities showed 25% [AM153] of rising parts, 19% [AO153] of falling parts, 33% [AQ153]of them were strong, and 67% [AP153] of V and A shapes.

For pp/sigma Entropic affinities showed 8% [AM148] of rising parts, 17% [AO148] of falling parts, 8% [AQ148]of them were strong, and 25% [AP148] of V and A shapes. Plain affinities showed 0% [AM154] of rising parts, 8% [AO154] of falling parts, 25% [AQ154]of them were strong, and 17% [AP154] of V and A shapes.

SS

For KNN Entropic affinities showed 6% [AM149] of rising parts, 3% [AO149] of falling parts, 8% [AQ149]of them were strong, and 0% [AP149] of V and A shapes. Plain affinities showed 39% [AM155] of rising parts, 22% [AO155] of falling parts, 75% [AQ155]of them were strong, and 67% [AP155] of V and A shapes.

For pp/sigma Entropic affinities showed 14% [AM150] of rising parts, 6% [AO150] of falling parts, 0% [AQ150]of them were strong, and 25% [AP150] of V and A shapes. Plain affinities showed 0% [AM156] of rising parts, 8% [AO156] of falling parts, 17% [AQ156]of them were strong, and 17% [AP156] of V and A shapes.

Total

For KNN Entropic affinities showed 15% [AM149] of rising parts, 8% [AO149] of falling parts, 11% [AQ149]of them were strong, and 22% [AP149] of V and A shapes. Plain affinities showed 32% [AM155] of rising parts, 21% [AO155] of falling parts, 54% [AQ155]of them were strong, and 67% [AP155] of V and A shapes.

For pp/sigma Entropic affinities showed 11% [AM150] of rising parts, 19% [AO150] of falling parts, 8% [AQ150]of them were strong, and 22% [AP150] of V and A shapes. Plain affinities showed 0% [AM156] of rising parts, 8% [AO156] of falling parts, 21% [AQ156]of them were strong, and 17% [AP156] of V and A shapes.

C.III **Algorithm type**

For KNN SDS and SES algorithms behave similarly having 50% of graphs with narrow shape. SPC algorithm has 52% of graphs described as wide and 22% of graphs

described as medium or narrow.

For pp/sigma we can observe different behaviour. For SPC algorithm 50% of graphs are described as narrow, while 20% and 26% of graphs are medium and wide.

Protein

Width

NFP

For KNN SPC algorithm showed 27% [AK87] of narrow shapes, 47% [AM87] of wide shapes, and 30% [AN87] of changes. SDS algorithm showed 60% [AK89] of narrow shapes, 20% [AM89] of wide shapes, and 30% [AN89] of changes. SES algorithm showed 73% [AK91] of narrow shapes, 23% [AM91] of wide shapes, and 40% [AN91] of changes.

For pp/sigma SPC algorithm showed 60% [AK88] of narrow shapes, 30% [AM88] of wide shapes, and 20% [AN88] of changes.
SDS algorithm showed 40% [AK90] of narrow shapes, 20% [AM90] of wide shapes, and 40% [AN90] of changes. SES algorithm showed 43% [AK92] of narrow shapes, 40% [AM92] of wide shapes, and 30% [AN92] of changes.

IDP

For KNN SPC algorithm showed 20% [AK93] of narrow shapes, 57% [AM93] of wide shapes, and 90% [AN93] of changes. SDS algorithm showed 43% [AK95] of narrow shapes, 47% [AM95] of wide shapes, and 40% [AN95] of changes. SES algorithm

showed 43% [AK97] of narrow shapes, 40% [AM97] of wide shapes, and 70% [AN97] of changes.

For pp/sigma SPC algorithm showed 43% [AK94] of narrow shapes, 27% [AM94] of wide shapes, and 70% [AN94] of changes. SDS algorithm showed 40% [AK96] of narrow shapes, 47% [AM96] of wide shapes, and 50% [AN96] of changes. SES algorithm showed 33% [AK98] of narrow shapes, 50% [AM98] of wide shapes, and 30% [AN98] of changes.

Total

For KNN SPC algorithm showed 23% [AK99] of narrow shapes, 52% [AM99] of wide shapes, and 60% [AN99] of changes. SDS algorithm showed 52% [AK101] of narrow shapes, 33% [AM101] of wide shapes, and 35% [AN101] of changes. SES algorithm showed 58% [AK103] of narrow shapes, 32% [AM103] of wide shapes, and 55% [AN103] of changes.

For pp/sigma SPC algorithm showed 52% [AK100] of narrow shapes, 28% [AM100] of wide shapes, and 45% [AN100] of changes. SDS algorithm showed 40% [AK102] of narrow shapes, 33% [AM102] of wide shapes, and 45% [AN102] of changes. SES algorithm showed 38% [AK104] of narrow shapes, 45% [AM104] of wide shapes, and 30% [AN104] of changes.

Shape

NFP

For KNN SPC algorithm showed 17% [AH249] of rising parts, 8% [AJ249] of falling parts, 50% [AL249] of them were strong, and 75% [AK249] of V and A shapes. SDS algorithm showed 8% [AH251] of rising parts, 42% [AJ251] of falling parts, 50% [AL251] of them were strong, and 75% [AK251] of V and A shapes. SES algorithm showed 67% [AH253] of rising parts, 0% [AJ253] of falling parts, 75% [AL253] of them were strong, and 0% [AK253] of V and A shapes.

For pp/sigma SPC algorithm showed 0% [AH250] of rising parts, 0% [AJ250] of falling parts, 0% [AL250] of them were strong, and 0% [AK250] of V and A shapes. SDS algorithm showed 0% [AH252] of rising parts, 17% [AJ252] of falling parts, 25% [AL252] of them were strong, and 0% [AK252] of V and A shapes. SES algorithm showed 0% [AH254] of rising parts, 0% [AJ254] of falling parts, 0% [AL254] of them were strong, and 0% [AK254] of V and A shapes.

IDP

For KNN SPC algorithm showed 17% [AH255] of rising parts, 17% [AJ255] of falling parts, 0% [AL255] of them were strong, and 100% [AK255] of V and A shapes. SDS algorithm showed 0% [AH257] of rising parts, 58% [AJ257] of falling parts, 75% [AL257] of them were strong, and 100% [AK257] of V and A shapes. SES algorithm showed 83% [AH259] of rising parts, 0% [AJ259] of falling parts, 75% [AL259] of them were strong, and 50% [AK259] of V and A shapes.

For pp/sigma SPC algorithm showed 0% [AH256] of rising parts, 0% [AJ256] of falling parts, 0% [AL256] of them were strong, and 50% [AK256] of V and A shapes. SDS algorithm showed 0% [AH258] of rising parts, 33% [AJ258] of falling parts, 100%

[AL258]of them were strong, and 50% [AK258] of V and A shapes. SES algorithm showed 0% [AH260] of rising parts, 0% [AJ260] of falling parts, 0% [AL260] of them were strong, and 0% [AK260] of V and A shapes.

Total

For KNN SPC algorithm showed 17% [AH261] of rising parts, 13% [AJ261] of falling parts, 25% [AL261] of them were strong, and 88% [AK261] of V and A shapes. SDS algorithm showed 4% [AH263] of rising parts, 50% [AJ263] of falling parts, 63% [AL263] of them were strong, and 88% [AK263] of V and A shapes. SES algorithm showed 75% [AH265] of rising parts, 0% [AJ265] of falling parts, 75% [AL265] of them were strong, and 25% [AK265] of V and A shapes.

For pp/sigma SPC algorithm showed 0% [AH262] of rising parts, 0% [AJ262] of falling parts, 0% [AL262] of them were strong, and 25% [AK262] of V and A shapes. SDS algorithm showed 0% [AH264] of rising parts, 25% [AJ264] of falling parts, 63% [AL264] of them were strong, and 25% [AK264] of V and A shapes. SES algorithm showed 0% [AH266] of rising parts, 0% [AJ266] of falling parts, 0% [AL266] of them were strong, and 0% [AK266] of V and A shapes.

Affinity

Width

EN

For KNN SPC algorithm showed 19% [AC99] of narrow shapes, 47% [AE99] of wide shapes, and 42% [AF99] of changes. SDS algorithm showed 81% [AC101] of narrow shapes, 8% [AE101] of wide shapes, and 25% [AF101] of changes. SES algorithm showed 64% [AC103] of narrow shapes, 25% [AE103] of wide shapes, and 50% [AF103] of changes.

For pp SPC algorithm showed 81% [AC100] of narrow shapes, 0% [AE100] of wide shapes, and 50% [AF100] of changes. SDS algorithm showed 67% [AC102] of narrow shapes, 6% [AE102] of wide shapes, and 67% [AF102] of changes. SES algorithm showed 56% [AC104] of narrow shapes, 17% [AE104] of wide shapes, and 50% [AF104] of changes.

PL

For KNN SPC algorithm showed 29% [AG99] of narrow shapes, 58% [AI99] of wide shapes, and 88% [AJ99] of changes. SDS algorithm showed 8% [AG101] of narrow shapes, 71% [AI101] of wide shapes, and 50% [AJ101] of changes. SES algorithm showed 50% [AG103] of narrow shapes, 42% [AI103] of wide shapes, and 63% [AJ103] of changes.

For sigma SPC algorithm showed 8% [AG100] of narrow shapes, 71% [AI100] of wide shapes, and 38% [AJ100] of changes. SDS algorithm showed 0% [AG102] of narrow shapes, 75% [AI102] of wide shapes, and 13% [AJ102] of changes. SES algorithm showed 12.5% [AG104] of narrow shapes, 87.5% [AI104] of wide shapes, and 0% [AJ104] of changes.

Total

For KNN SPC algorithm showed 23% [AK99] of narrow shapes, 52% [AM99] of wide shapes, and 60% [AN99] of changes. SDS algorithm showed 52% [AK101] of narrow shapes, 33% [AM101] of wide shapes, and 35% [AN101] of changes. SES algorithm showed 58% [AK103] of narrow shapes, 32% [AM103] of wide shapes, and 55% [AN103] of changes.

For pp/sigma SPC algorithm showed 52% [AK100] of narrow shapes, 28% [AM100] of wide shapes, and 45% [AN100] of changes. SDS algorithm showed 40% [AK102] of narrow shapes, 33% [AM102] of wide shapes, and 45% [AN102] of changes. SES algorithm showed 38% [AK104] of narrow shapes, 45% [AM104] of wide shapes, and 30% [AN104] of changes.

Shape

EN

For KNN SPC algorithm showed 3% [AC261] of rising parts, 6% [AE261] of falling parts, 0% [AG261] of them were strong, and 8% [AF261] of V and A shapes. SDS algorithm showed 8% [AC263] of rising parts, 8% [AE263] of falling parts, 8% [AG263] of them were strong, and 25% [AF263] of V and A shapes. SES algorithm showed 33% [AC265] of rising parts, 11% [AE265] of falling parts, 25% [AG265] of them were strong, and 33% [AF265] of V and A shapes.

For pp SPC algorithm showed 11% [AC262] of rising parts, 28% [AE262] of falling parts, 17% [AG262] of them were strong, and 33% [AF262] of V and A shapes. SDS algorithm showed 8% [AC264] of rising parts, 8% [AE264] of falling parts, 0% [AG264] of them were strong, and 17% [AF264] of V and A shapes. SES algorithm showed 14% [AC266] of rising parts, 19% [AE266] of falling parts, 8% [AG266] of them were strong, and 17% [AF266] of V and A shapes.

PL

For KNN SPC algorithm showed 17% [AH261] of rising parts, 13% [AJ261] of falling parts, 25% [AL261] of them were strong, and 88% [AK261] of V and A shapes. SDS algorithm showed 4% [AH263] of rising parts, 50% [AJ263] of falling parts, 63% [AL263] of them were strong, and 88% [AK263] of V and A shapes. SES algorithm showed 75% [AH265] of rising parts, 0% [AJ265] of falling parts, 75% [AL265] of them were strong, and 25% [AK265] of V and A shapes.

For sigma SPC algorithm showed 0% [AH262] of rising parts, 0% [AJ262] of falling parts, 0% [AL262] of them were strong, and 25% [AK262] of V and A shapes. SDS algorithm showed 0% [AH264] of rising parts, 25% [AJ264] of falling parts, 63% [AL264] of them were strong, and 25% [AK264] of V and A shapes. SES algorithm showed 0% [AH266] of rising parts, 0% [AJ266] of falling parts, 0% [AL266] of them were strong, and 0% [AK266] of V and A

Total

For KNN SPC algorithm showed 8% [AM261] of rising parts, 8% [AO261] of falling parts, 10% [AQ261] of them were strong, and 40% [AP261] of V and A shapes.

SDS algorithm showed 7% [AM263] of rising parts, 25% [AO263] of falling parts, 30% [AQ263] of them were strong, and 50% [AP263] of V and A shapes. SES algorithm showed 50% [AM265] of rising parts, 7% [AO265] of falling parts, 45% [AQ265] of them were strong, and 30% [AP265] of V and A shapes.

For pp/sigma SPC algorithm showed 7% [AM262] of rising parts, 17% [AO262] of falling parts, 10% [AQ262] of them were strong, and 30% [AP262] of V and A shapes. SDS algorithm showed 5% [AM264] of rising parts, 15% [AO264] of falling parts, 25% [AQ264] of them were strong, and 20% [AP264] of V and A shapes. SES algorithm showed 8% [AM266] of rising parts, 12% [AO266] of falling parts, 5% [AQ266] of them were strong, and 10% [AP266] of V and A

Sparsity

Width

DN

For KNN SPC algorithm showed 8% [AC127] of narrow shapes, 42% [AE127] of wide shapes, and 50% [AF127] of changes. SDS algorithm showed 58% [AC129] of narrow shapes, 17% [AE129] of wide shapes, and 50% [AF129] of changes. SES algorithm showed 33% [AC131] of narrow shapes, 67% [AE131] of wide shapes, and 100% [AF131] of changes.

For pp SPC algorithm showed 92% [AC128] of narrow shapes, 0% [AE128] of wide shapes, and 25% [AF128] of changes. SDS algorithm showed 50% [AC130] of

narrow shapes, 8% [AE130] of wide shapes, and 75% [AF130] of changes. SES algorithm showed 17% [AC132] of narrow shapes, 50% [AE132] of wide shapes, and 100% [AF132] of changes.

SP

For KNN SPC algorithm showed 29% [AG127] of narrow shapes, 54% [AI127] of wide shapes, and 63% [AJ127] of changes. SDS algorithm showed 50% [AG129] of narrow shapes, 42% [AI129] of wide shapes, and 13% [AJ129] of changes. SES algorithm showed 50% [AG131] of narrow shapes, 33% [AI131] of wide shapes, and 50% [AJ131] of changes.

For pp/sigma SPC algorithm showed 50% [AG128] of narrow shapes, 21% [AI128] of wide shapes, and 50% [AJ128] of changes. SDS algorithm showed 38% [AG130] of narrow shapes, 38% [AI129] of wide shapes, and 50% [AJ130] of changes. SES algorithm showed 42% [AG132] of narrow shapes, 38% [AI132] of wide shapes, and 13% [AJ132] of changes.

SS

For KNN SPC algorithm showed 22% [AK127] of narrow shapes, 50% [AM127] of wide shapes, and 58% [AN127] of changes. SDS algorithm showed 53% [AK129] of narrow shapes, 33% [AM129] of wide shapes, and 42% [AN129] of changes. SES algorithm showed 44% [AK131] of narrow shapes, 44% [AM131] of wide shapes, and 67% [AN131] of changes.

For pp/sigma SPC algorithm showed 64% [AK128] of narrow shapes, 14% [AM128] of wide shapes, and 42% [AN128] of changes. SDS algorithm showed 42% [AK130] of narrow shapes, 28% [AM129] of wide shapes, and 67% [AN130] of changes. SES algorithm showed 33% [AK132] of narrow shapes, 42% [AM132] of wide shapes, and 42% [AN132] of changes.

Total

For KNN SPC algorithm showed 25% [AO127] of narrow shapes, 52% [AQ127] of wide shapes, and 60% [AR127] of changes. SDS algorithm showed 52% [AO129] of narrow shapes, 37% [AQ129] of wide shapes, and 30% [AR129] of changes. SES algorithm showed 47% [AO131] of narrow shapes, 40% [AQ131] of wide shapes, and 60% [AR131] of changes.

For pp/sigma SPC algorithm showed 58% [AO128] of narrow shapes, 17% [AQ128] of wide shapes, and 45% [AR128] of changes. SDS algorithm showed 40% [AO130] of narrow shapes, 32% [AQ129] of wide shapes, and 60% [AR130] of changes. SES algorithm showed 37% [AO132] of narrow shapes, 40% [AQ132] of wide shapes, and 30% [AR132] of changes.

Shape

DN

For KNN SPC algorithm showed 8% [AC289] of rising parts, 8% [AE289] of falling parts, 0% [AG289] of them were strong, and 0% [AF289] of V and A shapes. SDS algorithm showed 17% [AC291] of rising parts, 8% [AE291] of falling parts, 0%

[AG291] of them were strong, and 50% [AF291] of V and A shapes. SES algorithm showed 50% [AC293] of rising parts, 25% [AE293] of falling parts, 75% [AG293] of them were strong, and 100% [AF293] of V and A shapes.

For pp SPC algorithm showed 8% [AC290] of rising parts, 42% [AE290] of falling parts, 25% [AG290] of them were strong, and 0% [AF290] of V and A shapes. SDS algorithm showed 17% [AC292] of rising parts, 8% [AE292] of falling parts, 0% [AG292] of them were strong, and 25% [AF292] of V and A shapes. SES algorithm showed 8% [AC294] of rising parts, 50% [AE294] of falling parts, 25% [AG294] of them were strong, and 25% [AF294] of V and A shapes.

SP

For KNN SPC algorithm showed 4% [AH289] of rising parts, 8% [AJ289] of falling parts, 0% [AL289] of them were strong, and 50% [AK289] of V and A shapes. SDS algorithm showed 8% [AH291] of rising parts, 29% [AJ291] of falling parts, 25% [AL291] of them were strong, and 63% [AK291] of V and A shapes. SES algorithm showed 46% [AH293] of rising parts, 4% [AJ293] of falling parts, 25% [AL293] of them were strong, and 13% [AK293] of V and A shapes.

For pp/sigma SPC algorithm showed 0% [AH290] of rising parts, 17% [AJ290] of falling parts, 13% [AL290] of them were strong, and 25% [AK290] of V and A shapes. SDS algorithm showed 4% [AH292] of rising parts, 17% [AJ292] of falling parts, 38% [AL292] of them were strong, and 25% [AK292] of V and A shapes. SES algorithm showed 8% [AH294] of rising parts, 4% [AJ294] of falling parts, 0% [AL294] of them were strong, and 13% [AK294] of V and A shapes.

SS

For KNN SPC algorithm showed 13% [AM289] of rising parts, 8% [AO289] of falling parts, 25% [AQ289] of them were strong, and 50% [AP289] of V and A shapes. SDS algorithm showed 0% [AM291] of rising parts, 29% [AO291] of falling parts, 50% [AQ291] of them were strong, and 38% [AP291] of V and A shapes. SES algorithm showed 54% [AM293] of rising parts, 0% [AO293] of falling parts, 50% [AQ293] of them were strong, and 13% [AP293] of V and A shapes.

For pp/sigma SPC algorithm showed 13% [AM290] of rising parts, 4% [AO290] of falling parts, 0% [AQ290] of them were strong, and 50% [AP290] of V and A shapes. SDS algorithm showed 0% [AM292] of rising parts, 17% [AO292] of falling parts, 25% [AQ292] of them were strong, and 13% [AP292] of V and A shapes. SES algorithm showed 8% [AM294] of rising parts, 0% [AO294] of falling parts, 0% [AQ294] of them were strong, and 0% [AP294] of V and A shapes.

Total

For KNN SPC algorithm showed 8% [AR289] of rising parts, 8% [AT289] of falling parts, 10% [AV289] of them were strong, and 40% [AU289] of V and A shapes. SDS algorithm showed 7% [AR291] of rising parts, 25% [AT291] of falling parts, 30% [AV291] of them were strong, and 50% [AU291] of V and A shapes. SES algorithm showed 50% [AR293] of rising parts, 7% [AT293] of falling parts, 45% [AV293] of them were strong, and 30% [AU293] of V and A shapes.

For pp/sigma SPC algorithm showed 7% [AR290] of rising parts, 17% [AT290] of falling parts, 10% [AV290] of them were strong, and 30% [AU290] of V and A shapes. SDS algorithm showed 5% [AR292] of rising parts, 15% [AT292] of falling parts, 25% [AV292] of them were strong, and 20% [AU292] of V and A shapes. SES algorithm showed 8% [AR294] of rising parts, 12% [AT294] of falling parts, 5% [AV294] of them were strong, and 10% [AU294] of V and A shapes.

C.IV **Sparsity type**

Protein type

Width

NFP

For KNN dense data showed 33% [AC133] of narrow shapes, 39% [AE133] of wide shapes, and 50% [AF133] of changes. sparse data showed 53% [AG133] of narrow shapes, 36% [AI133] of wide shapes, and 33% [AJ133] of changes. supersparse data showed 46% [AK133] of narrow shapes, 37% [AM133] of wide shapes, and 44% [AN133] of changes.

For pp/sigma dense data showed 56% [AC134] of narrow shapes, 17% [AE134] of wide shapes, and 50% [AF134] of changes. sparse data showed 50% [AG134] of narrow shapes, 25% [AI134] of wide shapes, and 33% [AJ134] of changes. supersparse data showed 52% [AK134] of narrow shapes, 22% [AM134] of wide shapes, and 44% [AN134] of changes.

IDP

For KNN dense data showed 33% [AC135] of narrow shapes, 44% [AE135] of wide shapes, and 83% [AF135] of changes. sparse data showed 33% [AG135] of narrow shapes, 50% [AI135] of wide shapes, and 50% [AJ135] of changes. supersparse data showed 33% [AK135] of narrow shapes, 48% [AM135] of wide shapes, and 67% [AN135] of changes.

For pp/sigma dense data showed 50% [AC136] of narrow shapes, 22% [AE136] of wide shapes, and 83% [AF136] of changes. sparse data showed 36% [AG136] of narrow shapes, 39% [AI136] of wide shapes, and 42% [AJ136] of changes. supersparse data showed 41% [AK136] of narrow shapes, 33% [AM136] of wide shapes, and 56% [AN136] of changes.

Total

For KNN dense data showed 33% [AC137] of narrow shapes, 42% [AE137] of wide shapes, and 67% [AF137] of changes. sparse data showed 43% [AG137] of narrow shapes, 43% [AI137] of wide shapes, and 42% [AJ137] of changes. supersparse data showed 40% [AK137] of narrow shapes, 43% [AM137] of wide shapes, and 56% [AN137] of changes.

For pp/sigma dense data showed 53% [AC138] of narrow shapes, 19% [AE138] of wide shapes, and 67% [AF138] of changes. sparse data showed 43% [AG138] of narrow shapes, 32% [AI138] of wide shapes, and 38% [AJ138] of changes. supersparse data showed 46% [AK138] of narrow shapes, 28% [AM138] of wide shapes, and 50% [AN138] of changes.

Shape

NFP

For KNN dense data showed 28% [AC295] of rising parts, 17% [AE295] of falling parts, 17% [AG295] of them were strong, and 50% [AF295] of V and A shapes. sparse data showed 17% [AH295] of rising parts, 8% [AJ295] of falling parts, 17% [AL295] of them were strong, and 25% [AK295] of V and A shapes. supersparse data showed 19% [AM295] of rising parts, 8% [AO295] of falling parts, 42% [AQ295] of them were strong, and 25% [AP295] of V and A shapes.

For pp/sigma dense data showed 11% [AC296] of rising parts, 17% [AE296] of falling parts, 0% [AG296] of them were strong, and 17% [AF296] of V and A shapes. sparse data showed 6% [AH296] of rising parts, 11% [AJ296] of falling parts, 8% [AL296] of them were strong, and 17% [AK296] of V and A shapes. supersparse data showed 3% [AM296] of rising parts, 3% [AO296] of falling parts, 0% [AQ296] of them were strong, and 8% [AP296] of V and A shapes.

IDP

For KNN dense data showed 22% [AC297] of rising parts, 11% [AE297] of falling parts, 33% [AG297] of them were strong, and 50% [AF297] of V and A shapes. sparse data showed 22% [AH297] of rising parts, 19% [AJ297] of falling parts, 17% [AL297] of them were strong, and 58% [AK297] of V and A shapes. supersparse data showed 25% [AM297] of rising parts, 17% [AO297] of falling parts, 42% [AQ297] of them were strong, and 42% [AP297] of V and A shapes.

For pp/sigma dense data showed 11% [AC298] of rising parts, 50% [AE298] of falling parts, 33% [AG298] of them were strong, and 17% [AF298] of V and A shapes. sparse data showed 3% [AH298] of rising parts, 14% [AJ298] of falling parts, 25% [AL298] of them were strong, and 25% [AK298] of V and A shapes. supersparse data showed 11% [AM298] of rising parts, 11% [AO298] of falling parts, 17% [AQ298] of them were strong, and 33% [AP298] of V and A shapes.

Total

For KNN dense data showed 25% [AC299] of rising parts, 14% [AE299] of falling parts, 25% [AG299] of them were strong, and 50% [AF299] of V and A shapes. sparse data showed 19% [AH299] of rising parts, 14% [AJ299] of falling parts, 17% [AL299] of them were strong, and 42% [AK299] of V and A shapes. supersparse data showed 22% [AM299] of rising parts, 13% [AO299] of falling parts, 42% [AQ299] of them were strong, and 33% [AP299] of V and A shapes.

For pp/sigma dense data showed 11% [AC300] of rising parts, 33% [AE300] of falling parts, 17% [AG300] of them were strong, and 17% [AF300] of V and A shapes. sparse data showed 4% [AH300] of rising parts, 13% [AJ300] of falling parts, 17% [AL300] of them were strong, and 21% [AK300] of V and A shapes. supersparse data showed 7% [AM300] of rising parts, 7% [AO300] of falling parts, 8% [AQ300] of them were strong, and 21% [AP300] of V and A shapes.

Affinity type

Width

EN

For KNN dense data showed 33% [AK5] of narrow shapes, 42% [AM5] of wide shapes, and 67% [AN5] of changes. sparse data showed 69% [AK7] of narrow shapes, 8% [AM7] of wide shapes, and 25% [AN7] of changes. supersparse data showed 61% [AK9] of narrow shapes, 31% [AM9] of wide shapes, and 25% [AN9] of changes.

For pp dense data showed 53% [AK6] of narrow shapes, 19% [AM6] of wide shapes, and 67% [AN6] of changes. sparse data showed 72% [AK8] of narrow shapes, 0% [AM8] of wide shapes, and 50% [AN8] of changes. supersparse data showed 78% [AK10] of narrow shapes, 3% [AM10] of wide shapes, and 50% [AN10] of changes.

PL

For KNN sparse data showed 17% [AK13] of narrow shapes, 78% [AM13] of wide shapes, and 58% [AN13] of changes. supersparse data showed 42% [AK15] of narrow shapes, 36% [AM15] of wide shapes, and 75% [AN15] of changes.

For sigma sparse data showed 14% [AK14] of narrow shapes, 64% [AM14] of wide shapes, and 25% [AN14] of changes. supersparse data showed 0% [AK16] of narrow shapes, 92% [AM16] of wide shapes, and 8% [AN16] of changes.

Total

For KNN dense data showed 33% [AK17] of narrow shapes, 42% [AM17] of wide shapes, and 67% [AN17] of changes. sparse data showed 43% [AK19] of narrow shapes, 43% [AM19] of wide shapes, and 42% [AN19] of changes. supersparse data showed 51%

[AK21] of narrow shapes, 33% [AM21] of wide shapes, and 50% [AN21] of changes.

For pp/sigma dense data showed 53% [AK18] of narrow shapes, 19% [AM18] of wide shapes, and 67% [AN18] of changes. sparse data showed 43% [AK20] of narrow shapes, 32% [AM20] of wide shapes, and 38% [AN20] of changes. supersparse data showed 39% [AK22] of narrow shapes, 47% [AM22] of wide shapes, and 29% [AN22] of changes.

Shape

EN

For KNN dense data showed 25% [AM145] of rising parts, 14% [AO145] of falling parts, 25% [AQ145] of them were strong, and 50% [AP145] of V and A shapes. sparse data showed 14% [AM147] of rising parts, 8% [AO147] of falling parts, 0% [AQ147] of them were strong, and 17% [AP147] of V and A shapes. supersparse data showed 6% [AM149] of rising parts, 3% [AO149] of falling parts, 8% [AQ149] of them were strong, and 0% [AP149] of V and A shapes.

For pp dense data showed 11% [AM146] of rising parts, 33% [AO146] of falling parts, 17% [AQ146] of them were strong, and 17% [AP146] of V and A shapes. sparse data showed 8% [AM148] of rising parts, 17% [AO148] of falling parts, 8% [AQ148] of them were strong, and 25% [AP148] of V and A shapes. supersparse data showed 14% [AM150] of rising parts, 6% [AO150] of falling parts, 0% [AQ150] of them were strong, and 25% [AP150] of V and A shapes.

PL

For KNN sparse data showed 25% [AM153] of rising parts, 19% [AO153] of falling parts, 33% [AQ153] of them were strong, and 67% [AP153] of V and A shapes. supersparse data showed 39% [AM155] of rising parts, 22% [AO155] of falling parts, 75% [AQ155] of them were strong, and 67% [AP155] of V and A shapes.

For sigma sparse data showed 0% [AM154] of rising parts, 8% [AO154] of falling parts, 25% [AQ154] of them were strong, and 17% [AP154] of V and A shapes. supersparse data showed 0% [AM156] of rising parts, 8% [AO156] of falling parts, 17% [AQ156] of them were strong, and 17% [AP156] of V and A shapes.

Total

For KNN dense data showed 25% [AM157] of rising parts, 14% [AO157] of falling parts, 25% [AQ157] of them were strong, and 50% [AP157] of V and A shapes. sparse data showed 19% [AM159] of rising parts, 14% [AO159] of falling parts, 17% [AQ159] of them were strong, and 42% [AP159] of V and A shapes. supersparse data showed 22% [AM161] of rising parts, 13% [AO161] of falling parts, 42% [AQ161] of them were strong, and 33% [AP161] of V and A shapes.

For pp/sigma dense data showed 11% [AM158] of rising parts, 33% [AO158] of falling parts, 17% [AQ158] of them were strong, and 17% [AP158] of V and A shapes. sparse data showed 4% [AM160] of rising parts, 13% [AO160] of falling parts, 17% [AQ160] of them were strong, and 21% [AP160] of V and A shapes. supersparse data

showed 7% [AM162] of rising parts, 7% [AO162] of falling parts, 8% [AQ162] of them were strong, and 21% [AP162] of V and A shapes.

Algorithms

Width

SPC

For KNN dense data showed 8% [AC127] of narrow shapes, 42% [AE127] of wide shapes, and 50% [AF127] of changes. sparse data showed 29% [AG127] of narrow shapes, 54% [AI127] of wide shapes, and 63% [AJ127] of changes. supersparse data showed 22% [AK127] of narrow shapes, 50% [AM127] of wide shapes, and 58% [AN127] of changes.

For pp/sigma dense data showed 92% [AC128] of narrow shapes, 0% [AE128] of wide shapes, and 25% [AF128] of changes. sparse data showed 50% [AG128] of narrow shapes, 21% [AI128] of wide shapes, and 50% [AJ128] of changes. supersparse data showed 64% [AK128] of narrow shapes, 14% [AM128] of wide shapes, and 42% [AN128] of changes.

SDS

For KNN dense data showed 58% [AC129] of narrow shapes, 17% [AE129] of wide shapes, and 50% [AF129] of changes. sparse data showed 50% [AG129] of narrow shapes, 42% [AI129] of wide shapes, and 13% [AJ129] of changes. supersparse data showed 53% [AK129] of narrow shapes, 33% [AM129] of wide shapes, and 42% [AN129]

of changes.

For pp/sigma dense data showed 50% [AC130] of narrow shapes, 8% [AE130] of wide shapes, and 75% [AF130] of changes. sparse data showed 38% [AG130] of narrow shapes, 38% [AI130] of wide shapes, and 50% [AJ130] of changes. supersparse data showed 42% [AK130] of narrow shapes, 28% [AM130] of wide shapes, and 67% [AN130] of changes.

SES

For KNN dense data showed 33% [AC131] of narrow shapes, 67% [AE131] of wide shapes, and 100% [AF131] of changes. sparse data showed 50% [AG131] of narrow shapes, 33% [AI131] of wide shapes, and 50% [AJ131] of changes. supersparse data showed 44% [AK131] of narrow shapes, 44% [AM131] of wide shapes, and 67% [AN131] of changes.

For pp/sigma dense data showed 17% [AC132] of narrow shapes, 50% [AE132] of wide shapes, and 100% [AF132] of changes.
sparse data showed 42% [AG132] of narrow shapes, 38% [AI132] of wide shapes, and 13% [AJ132] of changes. supersparse data showed 33% [AK132] of narrow shapes, 42% [AM132] of wide shapes, and 42% [AN132] of changes.

Total

For KNN dense data showed 33% [AC137] of narrow shapes, 42% [AE137] of wide shapes, and 83% [AF137] of changes. sparse data showed 43% [AG137] of narrow shapes, 43% [AI137] of wide shapes, and 42% [AJ137] of changes. supersparse data

showed 40% [AK137] of narrow shapes, 43% [AM137] of wide shapes, and 56% [AN137] of changes.

For pp/sigma dense data showed 53% [AC138] of narrow shapes, 19% [AE138] of wide shapes, and 75% [AF138] of changes. sparse data showed 43% [AG138] of narrow shapes, 32% [AI138] of wide shapes, and 38% [AJ138] of changes. supersparse data showed 46% [AK138] of narrow shapes, 28% [AM138] of wide shapes, and 50% [AN138] of changes.

Shape

SPC

For KNN dense data showed 8% [AC289] of rising parts, 8% [AE289] of falling parts, 0% [AQ289] of them were strong, and 0% [AP289] of V and A shapes. sparse data showed 4% [AH289] of rising parts, 8% [AJ289] of falling parts, 0% [AL289] of them were strong, and 50% [AK289] of V and A shapes. supersparse data showed 13% [AM289] of rising parts, 8% [AO289] of falling parts, 25% [AQ289] of them were strong, and 50% [AP289] of V and A shapes.

For pp/sigma dense data showed 8% [AC290] of rising parts, 42% [AE290] of falling parts, 25% [AQ290] of them were strong, and 0% [AP290] of V and A shapes. sparse data showed 0% [AH290] of rising parts, 17% [AJ290] of falling parts, 13% [AL290] of them were strong, and 25% [AK290] of V and A shapes. supersparse data showed 13% [AM290] of rising parts, 4% [AO290] of falling parts, 0% [AQ290] of them were strong, and 50% [AP290] of V and A shapes.

SDS

For KNN dense data showed 17% [AC291] of rising parts, 8% [AE291] of falling parts, 0% [AQ291] of them were strong, and 50% [AP291] of V and A shapes. sparse data showed 8% [AH291] of rising parts, 29% [AJ291] of falling parts, 25% [AL291] of them were strong, and 63% [AK291] of V and A shapes. supersparse data showed 0% [AM291] of rising parts, 29% [AO291] of falling parts, 50% [AQ291] of them were strong, and 38% [AP291] of V and A shapes.

For pp/sigma dense data showed 17% [AC292] of rising parts, 8% [AE292] of falling parts, 0% [AQ292] of them were strong, and 25% [AP292] of V and A shapes. sparse data showed 4% [AH292] of rising parts, 17% [AJ292] of falling parts, 38% [AL292] of them were strong, and 25% [AK292] of V and A shapes. supersparse data showed 0% [AM292] of rising parts, 17% [AO292] of falling parts, 25% [AQ292] of them were strong, and 13% [AP292] of V and A shapes.

SES

For KNN dense data showed 50% [AC291] of rising parts, 25% [AE291] of falling parts, 75% [AQ291] of them were strong, and 100% [AP291] of V and A shapes. sparse data showed 46% [AH291] of rising parts, 4% [AJ291] of falling parts, 25% [AL291] of them were strong, and 13% [AK291] of V and A shapes. supersparse data showed 54% [AM291] of rising parts, 0% [AO291] of falling parts, 50% [AQ291] of them were strong, and 13% [AP291] of V and A shapes.

For pp/sigma dense data showed 8% [AC292] of rising parts, 50% [AE292] of falling parts, 25% [AQ292] of them were strong, and 25% [AP292] of V and A shapes. sparse data showed 8% [AH292] of rising parts, 4% [AJ292] of falling parts, 0% [AL292] of them were strong, and 13% [AK292] of V and A shapes. supersparse data showed 8% [AM292] of rising parts, 0% [AO292] of falling parts, 0% [AQ292] of them were strong, and 0% [AP292] of V and A shapes.

Total

For KNN dense data showed 25% [AC299] of rising parts, 14% [AE299] of falling parts, 25% [AQ299] of them were strong, and 50% [AP299] of V and A shapes. sparse data showed 19% [AH299] of rising parts, 14% [AJ299] of falling parts, 17% [AL299] of them were strong, and 42% [AK299] of V and A shapes. supersparse data showed 22% [AM299] of rising parts, 13% [AO299] of falling parts, 42% [AQ299] of them were strong, and 33% [AP299] of V and A shapes.

For pp/sigma dense data showed 11% [AC300] of rising parts, 33% [AE300] of falling parts, 17% [AQ300] of them were strong, and 17% [AP300] of V and A shapes. sparse data showed 4% [AH300] of rising parts, 13% [AJ300] of falling parts, 17% [AL300] of them were strong, and 21% [AK300] of V and A shapes. supersparse data showed 7% [AM300] of rising parts, 7% [AO300] of falling parts, 8% [AQ300] of them were strong, and 21% [AP300] of V and A shapes.