Predicting Protein Folding Pathways Using Ensemble Modeling and Sequence Information

by

David Becerra

McGill University
School of Computer Science
Montréal, Québec, Canada

A thesis submitted to McGill University in partial fulfillment of the requirements of the
degree of Doctor of Philosophy

Dedicated to my GRANDMOM. Thanks for everything. You are such a strong woman

Acknowledgements

My doctoral program has been an amazing adventure with ups and downs. The most valuable fact is that I am a different person with respect to the one who started the program five years ago. I am not better or worst, I am just different.

I would like to express my gratitude to all the people who helped me during my time at McGill. This thesis contains only my name as author, but it should have the name of all the people who played an important role in the completion of this thesis (and they are a lot of people). I am certain I would have not reached the end of my thesis without the help, advises and support of all people around me.

First at all, I want to thank my country Colombia. Being abroad, I learnt to value my culture, my roots and my south-american positiveness, happiness and resourcefulness. I am really proud of being Colombian and of showing to the world a bit of what our wonderful land has given to us. I would also like to thank all Colombians, because with your taxes (via Colciencia's funding) I was able to complete this dream. I also like to thank Canada for accepting us in this beautiful land and for embracing our lives and dreams.

I would like to express my deepest gratitude to my wife Luisa and to my son Samuel. You are a constant (and fundamental) source of motivation and encouragement. Your are the real reason why this dream become a reality. I will be always in depth debt with you for walking this journey with me. You have inspired me to be better everyday and to do not give up during the hard times. You know that this thesis really belongs to you two.

I would like to thank my parents for supporting my dreams. They have always encouraged us (me and my brother) to take risks and to work for our dreams. I deeply appreciate their help, support and advices. I still have a lot to learn from you. I want to extend my appreciation to my brother Andres. He is an excellent professional and he has been a long personal and career model to follow. I only hope that all our efforts would inspire Samuel and Maria Paula to pursuit their dreams and happiness.

I would like to express my deepest gratitude to my supervisor Prof. Jérôme Waldispühl for giving me the opportunity to carry out the research presented in this thesis. His guidance, motivation and patience were fundamental for succeeding on this project. Working with Professor Waldispühl has represented for me a once-in-a-life-time opportunity to

learn from a very motivate, innovative and supportive person. Beside my supervisor, I would like to thank Professor Mathieu Blanchette for their generous time and good will through all the (non-appointed) meetings that we had. I feel very honored to have such a distinguished researcher and person as my advisor.

I am also very grateful to all the current and former members of the Waldispüh and Blanchette research groups. I would particularly like to thank Alex and Chris, who helped me in countless ways, from giving me advices, sharing complain lunches, fixing my code and for reading and editing this manuscript. Your friendships have always been invaluable. I also really appreciate Mathieu Lavallé's tips about the Canadian life. Our shared hobbies of watching cycling and soccer sports made my integration to the lab (and country) much easier. Many thanks to Ayrin and Rola for caring so much about Samuel and my family. Your support was fundamental during all this process. I would like to show my gratitude to Mohamed. You are the right model to follow to balance parenthood and graduate studies. I owe sincere and earnest thankfulness to the students who I mentored (Zheng, Shu and Ettienne). I learn so much from you guys. I owe sincere and earnest thankfulness to all the members of the labs with who I shared invaluable moments during the daily life in the group Olivier, Vladimir, Mohammed I and II, Olivia, Antoine, Dilmi, Chrisostomos, Carlos, Roman, James, Jimmy, Faizy, Javier, Navin, Pouriya, Maia.

There are many people to thank beyond the scope of the research environment. Thanks a lot to Juan, Ismenia, Paulina, Felipe, Alejandro, Silvana, Patricia, Pablo, Jessica here in Montreal. Back in Colombia, I would like to thank a los compas Barrantes, Fernando, Goyes, Gomez, Juan Carlos, Lisbeth.

I would like to show my deepest gratitude to my friend Sergio in Netherlands. Even in the long distance, Sergio has been always supporting me and giving me the motivation needed to succeed in this doctoral process. I really value our friendship and I really hope to keep you as my friend for long time.

I am also want to thank Norman for his valuable advices. I will be always in depth debt with all the Computer Science staff. I thank Sheryl, Ann, Diti, Tricia and Professor Kemme.

I apologize for all the persons I have not mentioned. There could be many names that escape my mind right now, but you know that you play an important role in my life. Thanks a lot for all.

Abstract - Abrégé

The protein folding problem aims to predict the complete physical and dynamical process that transforms an unfolded sequence into a functional 3D protein structure. This problem consists of two (open) sub-problems: *i*) the protein structure prediction problem and *ii*) the protein pathway prediction problem. Computational techniques to face these two sub-problems have been based on the theory of evolution and laws of physics. To-date, classical approaches to obtaining detailed information about protein folding rely on time-consuming methods that are primarily limited to relatively small proteins (i.e., $\leq$ 50 amino acids). The overall objective of this thesis is to explore algorithms that conciliate: *i*) the prediction of protein structures and pathways, *ii*) physical-based predictions (i.e., low free-energy models) & evolutionary based predictions (i.e., sequence variation methods), and *iii*) computational costs and granularity level requirements of protein folding simulations. We propose an algorithmic framework for predicting protein folding that offers a better trade-off between resolution and efficiency. This framework computes accurate coarse-grained representations of the conformational landscape for large proteins through the combination of ensemble modeling techniques and evolutionary based sequence information. The resulting conformational energy landscape is then used to predict dominant folding pathways. Given that the proposed framework in this thesis makes use of sequence information, we also explore a crowdsourcing and multi-objective evolutionary strategy to investigate the accuracy of evolutionary information encoded by multiple sequence alignments. Finally, to present our results to the wider biology and computer science communities, we develop an easy-to-use interactive molecular visualizer.

Abstract - Abrégé

Le problème du repliement des protéines a pour objectif de prédire le processus physique et dynamique complet de conversion de la séquence d'une protéine non-repliée en une structure 3D. Ce problème peut être décomposé en deux sous-problèmes (ouverts): *i*) le problème de la prédiction de la structure des protéines et *ii*) le problème de la prédiction de le processus de repliement des protéines. La théorie de l'évolution et les lois de la physique sont les principes sur lesquels les techniques computationnelles se sont basées afin de résoudre ces deux sous-problèmes. À ce jour, les approches classiques qui sont utilisées afin d'obtenir des informations sur le repliement des protéines sont reliées à des méthodes chronophages, en principe limitées à des protéines de petite taille. L'objectif général de cette thèse est d'explorer des algorithmes qui concilient : *i*) la prédiction des structures et celle du processus de repliement des protèines, *ii*) les prédictions basées sur les lois physiques (i.e., des modèles à énergie libre minimale) et celles basées sur la théorie de l'évolution (i.e., des méthodes basées sur les variations de séquences) et *iii*) les coûts computationnels et les niveaux de granularité requis par les simulations. Dans ce but, nous proposons une structure d'algorithme pour le repliement des protéines qui offre un meilleur compromis entre la résolution et l'efficience. Cet algorithm calcule des représentations précises à gros grain du paysage conformationnel des protéines de grande taille, en combinant des techniques de modélisation d'ensemble avec l'informations évolutive des séquences. Ce paysage énergétique conformationnel est ensuite utilisé afin de prédire les voies de repliement les plus probables. Puisque la structure proposée dans le cadre de cette thése utilise des informations de séquences, nous étudions aussi une stratégie participative, et une autre, multi-objectif et évolutive pour examiner la précision de l'information encodée par les alignements multiples de séquences. Finalement, de manière à présenter nos résultats à la communauté de biologistes et de chercheurs en informatique, nous avons développé un visualiseur moléculaire interactif facile d'utilisation.

Declaration of Authorship

This thesis comprises research work done wholly while in candidature for a research degree at McGill University. This work was performed under Dr Jérôme Waldispühl supervision. Collaborative work and published work are listed below in the order they appear in this thesis.

- Section 1.4 was written in collaboration with Dr Mohamed Raef Smaoui.
- Section 2.2.2 and 2.2.3, Figures 2.2, 2.3, 2.4, 2.5 and 2.6, were extracted from the scientific article [1]. I am a first author of this article. The design, preparation, analysis, and exposition of the data and experiments were performed by me under Dr Jérôme Waldispühl supervision. Daniel Kwak designed and implemented the `Open-Phylo` interface and server, Alfred Kam designed and implemented the `Phylo` casual video game, Qikuan Zhou designed and implemented the `Phylo` expert interface, Adam Hops prepared the data, Eleyine Zarour participated in the design and implementation of the `Open-Phylo` interface, Arthur Kam participated in the implementation of the `Phylo` casual video game , Luis Sarmenta contributed to the design of the `Phylo` casual video game. Dr Mathieu Blanchette and Dr Jérôme Waldispühl designed the research.
- Sections 2.3.1, 2.3.2, 2.3.3 and 2.3.4, Figures 2.7, 2.8, 2.9, and 2.10, were extracted from the scientific article [2]. I am a first author of this article. The design of the research, the implementation of the algorithm, the analysis of the data and the writing of the manuscript were performed by me and Wilson Soto.
- I supervised two undergraduate students during the development of this work. Zheng Dai collaborated with me during the methodological definition of `efold`. Shu Haykawa collaborated with me during the development of the online pathway visualizer.
- I got editorial help and advise from Christopher Cameron.

Contents

List of Figures

List of Tables

Abbreviations

3D	**T**hree **D**imensional
AA	**A**mino **A**cid
BPF	**B**oltzmann **P**artition **F**unction
CASP	**C**ritical **A**ssessment of protein **S**tructure **P**rediction
DFS	**D**epth **F**irst **S**earch
DNA	**D**eoxyribo**N**ucleic **A**cid
DP	**D**ynamic **P**rogramming
DSSP	**D**ictionary of **P**rotein **S**econdary **S**tructure
GUI	**G**raphical **U**ser **I**nterface
HMM	**H**idden **M**arkov **M**odel
MD	**M**olecular **D**ynamics
MCM	**M**onte **C**arlo **M**ethods
MOEA	**M**ulti **O**bjective **E**volutionary **A**lgorithm
MOOP	**M**ulti **O**bjective **O**ptimization **P**roblem
NMR	**N**uclear **M**agnetic **R**esonance
ODE	**O**rdinary **D**ifferential **E**quation
PDB	**P**rotein **D**ata **B**ank
PF	**P**rotein **F**olding
PFAM	**P**rotein **FAM**ily data base
PPP	**P**rotein **P**athway **P**rediction
PRM	**P**robabilistic **R**oad **M**aps
PSP	**P**rotein **S**tructure **P**rediction
RNA	**R**ibo**N**ucleic **A**cid

RMSD	**R**oot **M**ean **S**quare **D**eviation
SCOP	**S**tructural **C**lassification **O**f **P**roteins
SS	**S**econdary **S**tructure
TSE	**T**ransition **S**tate **E**nsemble
vdW	**v**an **d**er **W**aals

Physical Constants

Boltzmann constant	k	=	$13.805 \times 10^{-24}\ J\ deg^{-1}$
Angstrom	$\mathring{A}$	=	$0.1\ nm$

Symbols

P strand interaction	Parallel
A strand interaction	Anti-Parallel
N strand interaction	None
β secondary structure	Beta strand
α secondary structure	Alpha helix

Chapter 1

Introduction

1.1 Protein Folding

Of all the molecules found in living organisms, proteins are probably the most studied due to the variety of critical tasks that they perform for cellular metabolism. Specifically, they participate in several vital functions such as: *i*) cellular growth and repair, *ii*) the catalysis of cellular chemical reactions, *iii*) the transport of molecules, *iv*) signal transduction, *v*) segregation of genetic material, *vi*) production and use of energy, and *v*) producing biochemicals such as antibodies, enzymes and hormones. Furthermore, they are important building blocks of bones, muscles, cartilage, skin and blood. Proteins are so important that if their function is impaired, the consequences for the organism can be devastating. Failure to maintain a functional protein may produce a wide range of diseases with different pathological mechanisms and dramatic social impact, for which there are no current treatment.

Proteins, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) are the three different types of molecules in which the biological information of the cell is stored [3]. The instructions needed to construct proteins are encoded in DNA; however, DNA must first be transcribed into RNA before then being translated into amino acids (AA). During transcription, the genetic information encoded within DNA is transcribed into a messenger RNA (mRNA). mRNA molecules are segmented into codons - three consecutive

nucleotides that encode an AA - which are then translated into an AA sequence. However, this translation is based on the ability for codons to form 64 possible encodings (each of the three nucleotides can be one of the four nucleotides adenine, cytosine, guanine or uracil), only 20 standard AA[1] which will be used by cells in protein biosynthesis are represented. An AA is a molecule that contains an amine ($-NH_2$) and a carboxyl ($-COOH$) functional groups, along with a side-chain (*R group*) specific to each AA. Each AA differs in structure by their side-chain substituent and is classified as either hydrophobic (low propensity to be in contact with water), hydrophilic (energetically favourable contact with water) or amphipathic (have both hydrophobic and hydrophilic character) based on their differences in structure, size, electric charge and solubility in water. The molecular makeup allows individual AA to join together in long chains by forming bonds as they fold into a three-dimensional (3D) structure, which then defines the protein function.

The biochemical function of a protein is determined by its 3D structure, and in many cases, by assembling with other proteins into a functional complex (e.g., hemoglobin, DNA polymerase, and ion channels). In addition, many proteins undergo further (post-translation) modifications to improve their functionality. The major steps during the folding process are assisted and facilitated by proteins called *molecular chaperones*. Chaperones do not convey additional information to determine the protein structure; but they help to prevent incorrect folding or aggregation processes (e.g., loss-of-function and gain-of-toxic-function misfolding diseases). The interior of a living cell is an extraordinary complex environment with a complete set of machinery to assist in the folding process. H owever, under certain circumstances a protein can fail in the adoption or maintaineance of its native conformation (i.e., the most stable natural conformation). This malfunctioning folding procedure may result in pathological conditions referred to as *protein misfolding diseases* [4]. Misfolding leads to a wide range of diseases with different pathological mechanisms and dramatic social impact, such as Alzheimer and Parkinson diseases, type 2 diabetes, cystic fibrosis, some forms of emphysema, and many others [5].

In order to better understand protein folding, it is important to consider the four possible levels of folding a protein undergoes. Each successive level of protein folding ultimately

[1]There are twenty two AA that are naturally incorporated into polypeptides; however, of these, only twenty are encoded by the universal genetic code.

contributes to its 3D conformation and therefore its function. *i*) *Primary structure* consists of the AA in sequenced order. *ii*) *Secondary structure* (SS) consists of regular components of folding patterns that contribute to the stabilization of the protein folding process, like α-helices and β-sheets. The α-helix is a right-handed, coiled strand that is remarkably stable due to hydrogen bonds. In contrast, a β-sheet conformation consists of hydrogen-bonding between pairs of strands lying side-by-side. The two strands of a β-sheet may be either parallel or anti-parallel depending on whether the individual strand directions are the same or opposite. *iii*) In *tertiary structure*, elements of the SS are folded forming an almost solid compact structure (i.e., there are limited structural changes after reaching the native state) that is stabilized by forces due to bonding interactions between the side-chain groups of the AA. Tertiary structure refers to the overall folding of the entire polypeptide chain into a specific 3D shape. It is important to stress that the functional properties of the protein are completely dependent on the tertiary structure. *iv*) In the *quaternary structure*, many proteins are made up of multiple polypeptide chains, where these chains are represented by tertiary structures interacting with each other. The result is that these tertiary structures arrange themselves to form a larger protein complex.

The aforementioned levels of folding have been a convenient hierarchical description (in which successive layers of the hierarchy describe increasingly more complex levels of organization) of protein structure to simplify the daunting task of deciphering underlying patterns within the protein folding process. The final result of the folding process is the full 3D structure of the protein (i.e., the process of going from the primary to the tertiary [or quaternary] structural level); however, this process is currently not fully understood. Based on several seminal laboratory experiments involving an active form of the Ribonuclease A enzyme[2], the 1972 nobel prize winner Christian Anfinsen, demonstrated that RNase could be fully denatured (i.e., a protein loses its tertiary structure and catalytic activity) under extreme chemical conditions, and then renatured (i.e., a protein recovers its tertiary structure with full catalytic activity) when the chemical environment is returned to natural cellular conditions. This experiment demonstrates that the primary

[2]RNase A is one of the classic model systems in the study of proteins largely because it is extremely stable and could be purified in large quantities.

sequence in a protein contains all the necessary information to determine its 3D structure. Thus, the protein folding process may then be explained entirely by the physical and chemical interactions among AAs [6]. This experiment also allowed Anfinsen to theorize that the native structure of a protein is unique, stable and thermodynamically the most stable conformation [7] under environmental conditions at which folding may occur. Thus, proteins must follow energetically favourable pathways to form the most stable natural conformation (known as *native state*). In other words, proteins must form intermediate structures in a time ordered sequence of structural changes during the folding process (intermediates known as *folding pathways*). The existence of folding pathways agrees with the hypothesis that finding the native folded state of a protein by a random search among all possible configurations will never succeed [8]. Particularly, based on the fact that many naturally-occurring proteins fold reliably and quickly to their native states despite the astronomical-large number of possible configurations (estimated to grow as 10^n for an n-residue protein [9]), the folding process can not occur by random diffusion alone.

Since Anfinsen's experiments, the protein folding process has been researched with two mutually exclusive goals: *i*) achieving the most stable conformation (i.e., thermodynamics control) and *ii*) doing so quickly (i.e., kinetic control) [10]. Based on a thermodynamic control, the protein folding is independent of the pathway taken to achieve the global minimum. While kinetic control is pathway dependent and the final structure is different depending on the initial conditions of the system. Different folding models have been proposed on the basis of model systems and general physicochemical reasoning to characterize the thermodynamical and kinetic features occurring during protein folding [11]. Those models are qualitative and heuristic emphasizing essential features of the folding process and disregarding its complexities. Even if those models can only make approximate predictions of the folding pathway for a specific protein; they offer insights into possible folding landscapes. It is not possible to determine the correctness of a model for all proteins *a priori* because different studies have shown that some proteins combine features from different models (i.e., the various competing pathways need not be mutually exclusive). In the following paragraphs, we will explore the main characteristics

and properties of the hydrophobic collapse, framework, nucleation and energy landscape models.

The hydrophobic collapse model hypothesizes that the native protein conformation is formed by a rearrangement of compact hydrophobic collapsed structures [12, 13]. This model is based on the observation that the protein's native state often contain a hydrophobic core. This model is generally used to describe the initial stage of protein folding, where the hydrophobic collapse produces a molted globule with a SS but no tertiary topology [14]. The model proceeds in the following folding steps: *i*) Form initial SS, creating localized regions of predominantly hydrophobic residues, *ii*) nascent protein interacts with water, and *iii*) the thermodynamic pressures on the SS make them collapse into a tertiary conformation with the hydrophobic core. This model is consistent with various simulations [15–17] that predicts the hydrophobic collapse as a relatively early event in the folding pathway (occurs before the formation of SS). However, there are some experiments [18] in which some proteins do not appear to undergo an initial collapse reaction.

The framework model considers the protein folding process as a step-wise and hierarchical process [19]. In this model, local interactions guide the formation of SS, followed by the random diffusion collisions (i.e., the process of collision, combination and strengthen of conformational local elements) until a stable native tertiary structure is formed. This model proposes the following folding steps: *i*) start with an unfolded protein chain, *ii*) form the SS according to the primary structure, but independent of the tertiary structure, *iii*) allow SS to fluctuate around their native position(s), *iv*) SS elements interact with each other to form a native-like SS, and finish by *v*) identifying the folding pattern to the 3D conformation. This model has been shown to agree with experiments [20, 21] that indicate for some proteins the folding of SS elements occur many times long before the main folding event. It is important to note that the process of hierarchic condensation has not been verified experimentally for large number of proteins.

The nucleation model is based on the assumption that the protein folding process is similar to crystallization and that the limiting step is nuclear formation followed by

a rapid propagation of structure folding. In this model, the secondary and tertiary protein structures are formed simultaneously. This model features the following four main steps [22]: *i*) initial formation of a folding nuclei composed of residues close to each other not only in sequence, but also in structure with respect to the native protein conformation, *ii*) stabilization of the nuclei by encompassing contiguous residues, *iii*) two or more nuclei are then mutually stabilized by long-range interactions. Finally *iv*) the rearrangement step, where structured sections of the protein alter their conformation. The nucleation model is supported by studies [23] that identify proteins who fold via a nucleation mechanism with a preference for residues belonging to regular SS in the folding nuclei. However, there are other proteins for which the structural consolidation in the major transition state appears to have progressed beyond initial nucleation [24].

The energy landscape model states that the folding of a protein does not follow a singular, specific pathway but occurs through multiple routes down a folding funnel with a high level of irregularity [25]. This funnel represent an energy landscape where the y-axis is the internal free energy of a given protein configuration, and the lateral axes represent the conformational coordinates (and the degrees of freedom available to a polypeptide chain). In this model, a conformation is represented by a point on the energy landscape and as the conformation moves lower in the landscape, it is close to the native state. This *new view* of protein folding suggests that a protein (initially unfolded) folds through a heterogeneous population of intermediate folded structures in a fluctuant equilibrium, instead of through predefined pathways with compulsory intermediates. The downhill nature of the folding ensures that folding proceeds at random one AA at a time and not in a set of folding steps as described by the previous models. The landscape proposed by this model has been explored by theoretical and experimental studies, which have endorsed a multi-pathway view (i.e., proteins do not follow a singular pathway, but explore multiple routes down the folding funnel). Therefore, this model compliments experiments [26–29] where several proteins fold via parallel routes and those proteins switch their preferred folding pathways depending on the environmental conditions.

As one can see from the models described above, considerable research has been dedicated to the goal of achieving a fundamental understanding of the true folding mechanisms for proteins. In particular, the study of proteins has made significant progress by combining computational techniques and proteomics technologies, where the prediction of protein folding is still a major goal in the proteomics era and a great challenge in computational biology. Folding scenarios have found strong supporting evidence for some small proteins (in size and availability) through experimental and computational experiments; however, the conditions under which folding occurs (in-vitro vs in-silico) are vastly different to the conditions within a living cell (in-vivo). Specifically, *i*) the interior of a cell is a highly crowded and dynamic environment when compared with the carefully chosen in in-vitro (or in-silico) studies, *ii*) the time-scales of the folding processes observed by in-vitro and in-silico experiments are protein dependent, and *iii*) most studies on in-vitro and in-silico are usually done in the absence of other proteins in dilute conditions (avoiding aggregate processes). Meanwhile in living cells, a large and diverse group of proteins (like molecular chaperones) are available to assist in the formation of the native structure. Even if all the information needed to achieve the native structure of a protein is hidden in its AA sequence, the folding process in living cells depends on the presence of complex protein machinery and it is assumed that only a small proportion of the proteins may assume their native structure(s) on their own [30].

Improvements to in-vitro and in-silico techniques have revolutionized the scale at which we are able to study protein folding mechanisms. The combined efforts from biochemical and computational experiments have helped to confirm many predictions of analytical theory regarding protein folding mechanisms (such as the folding models previously described). Moreover, the combination of knowledge gained from theory, experimentation, and simulations have allowed for the creation of an atomistic level description of folding landscapes [31, 32]. Despite common goals, it has not always been easy aligning both areas of study in a shared goal. In particular, in-silico methods have focused on the simulation of small proteins molecules with no folding intermediates (at the microsecond timescale) due to limited computational resources. On the other hand, such fast folding proteins make the experimental detection of events during the folding process difficult, which might make them unsuitable for in-vitro folding simulation studies.

In the last 15 years, experimental methods have allowed for the resolution of fast folding reactions and visualizing single molecules during folding. Folding experiments in vitro, such as refolding studies and the determination of 3D structures by X-ray crystallography and Nuclear Magnetic Resonance spectroscopy (NMR) experiments, have provided the basis on which to study the folding process. In-vitro studies of protein dynamics and folding mechanisms continue to be an active area of research but currently are unable to provide a complete picture of the dynamic processes at a microscopic scale of size and time. While promising experimental devices and procedures are constantly being developed and enhanced, experimentalists are facing both technology limitations and increased complexity (e.g., the need of increasing resolution of experimental data and the demand to deal with larger molecular systems) that restrains their progress. Many have turned to in silico methods to gain new insight in experimental problems. For example, simulations are used to perform efficient explorations of potential protein folding solutions and spaces, to unravel hypotheses that are hard to explore experimentally and to construct more focused experimental designs that may suggest new interpretations of experiments [33].

In-silico methods offer an alternative by treating the folding phenomena with varying degrees of abstraction. As computational resources have begun to be more powerful, more detailed simulations of larger systems at a higher time scale have been possible. In silico simulations of protein folding have several advantages over its in-vitro counterpart [9]. Particularly, *i*) the protein is pure by definition, which allows avoiding aggregate processes, *ii*) the folding process can be modeled under unusual or unfeasible laboratory conditions, *iii*) the contributions of various interactions can be determined explicitly; allowing an unambiguously determination of the folding scenario, and *iv*) classical methods for protein structure analysis are very time consuming, error prone, and expensive.

The protein folding (PF) problem aims to predict the complete physical and dynamical process that transforms an unfolded protein sequence into a functional 3D structure. In other words, PF has been the task of predicting how the information coded in an AA sequence of proteins translates into the 3D structure of a biologically active protein. Anfinsen thermodynamic hypothesis has fuelled efforts to predict 3D structures and folding

pathways from sequences through the development of well-guided approaches; however, an enormous challenge for PF methods has been the ability to predict 3D native structures and folding pathways for the broad range of proteins currently known. This set of proteins is composed of thousands of different folds, different structural families, and an unknown number of unique folding mechanisms. In order to face this challenge, historically, the PF problem has been splitting in two related problems: *i*) the protein structure prediction problem (PSP) and *ii*) the protein pathway prediction problem (PPP). Both problems have been widely acknowledged as open problems; but due to its inherent complexity, PSP has been used as a necessary preliminary step toward PPP. Furthermore, the lack of computational biology tools to model the PPP results in information that is embedded within folding pathways to remain largely unexploited. In the following sections, the main in-silico methods developed to determine and study the folding structures and dynamics of proteins will be reviewed. The methods described in these sections are not meant to provide an exhaustive overview of all methods in the field, but rather to present the reader with a concrete set of solutions and methodologies that illustrate major techniques related to the PF prediction problem.

1.2 Protein Structure Prediction Methods

Due to the importance of understanding protein structure and the consequences of structure on function, a tremendous amount of research has focused on understanding the protein structure acquisition process. In contrast to genome-based methods, protein analysis struggle to attain the same level of throughput. The major limitations in the biochemical process are: *i*) proteins cannot be amplified in a manner similar to nucleic acids, *ii*) since the appearance of the first atomic-resolution protein structures in 1958, the complexity of examining protein structures has increased as new macromolecules are discovered, and *iii*) the cost of methods to characterize proteins is prohibitive. Furthermore, post-translational modifications, regulatory mechanisms and environmental factors can result in proteins with multiple forms and structures further complicating the analysis. As a result, the number of available protein sequences increases exponentially based

on the success of genome-scale sequencing projects. However, the number of experimentally known protein 3D models is very small compared to the total number of sequenced proteins [34]. To date, the last release[3] of the UniProtKB/TrEMBL data base contains 80'204,459 protein sequence entries, meanwhile the RCSB protein data bank (PDB) stores approximately 128,783 structures.

The theory of evolution and the laws of physics are the principles on which the techniques of protein structure prediction have been based [35]. These methods rely on the idea that functional proteins undergo a natural selection process preserving their function, and by consequence, their structure. On the other hand, proper folding dynamic properties enable proteins to fold quickly from a primary sequence state to a native 3D structure. Thus, a functional protein can be characterized by natural selection (i.e., theory of evolution) and/or folding properties (i.e., laws of physics). Figure 1.1 represents the three main methodologies involved in the PSP problem. Comparative models and fold recognition methods are based on the theory of evolution and they rely on the folding similarity between a target protein and known protein structures. These approaches are able to prune large search spaces of possible protein structures assuming that the protein whose structure is unknown (the target) adopts a structure that is close to experimentally determined structures (the templates) [36]. By contrast, *ab initio* methods use the laws of physics to predict a protein structure from its AA sequence without relying on similarity at the fold level between a target structure and a set of templates [37]. A successful *ab initio* modeling usually has the following methodology patterns: *i*) an accurate representation of conformations that overcome the high complexity in sampling protein conformations, *ii*) an accurate energy function with which the native structure of a protein corresponds to the most thermodynamically stable state, *iii*) an efficient search method which can quickly identify the lowest energy conformations in a vast conformational space, and *iv*) a method for selection and evaluation of native-like models.

The most accurate protein structures prediction tools integrate ideas from all three categories (i.e., comparative, threading and *ab initio*) [38]. Generally, if a sequence identity $> 30\%$ is attainable, then a relatively accurate folding prediction can be obtained from

[3]Release 2017-03 of 30-March-2017

comparative methods; but if the target-template has little similarity but there are models with similar structural motifs, then threading methods are the chosen option to perform the prediction. If none of the conditions described above are assured, then *ab initio* methods are the modeling choice.

1.2.1 Comparative Methods

Comparative approaches are based on the hypothesis that a small change in the protein sequence usually results in a small change in its 3D structure and sequences that have a common ancestor also have similar folds. Therefore, comparative methods rely on detectable sequence similarity between the target sequence and at least one known structure (i.e., the templates) [35].

The main steps in comparative model methods are as follows [35]. The first step consists of searching for structures related to the target sequence. Comparative modeling usually starts by searching for known protein structures in the PDB using the target sequence as a query by comparing to either a single or multiple sequence(s). The second step consists of selecting templates (i.e., known protein structures) from the chosen biological data base. The quality of a template increases with its overall sequence similarity to the target and decreases with the number and length of gaps in the alignment. The next step consists of building the model. This step is usually accomplished using strategies such as assembly of rigid bodies [36], coordinate reconstruction [57], or satisfaction of spatial restraints [58]. The final step involves evaluating the model to verify that the final protein model is correct and to check for possible errors [66]. An internal evaluation will check whether a model satisfies the constraints used to calculate the model, while an external evaluation will rely its conclusions on information (such as energetic and empirical methods) that was not used during the calculation of the model.

The usefulness of comparative modeling has been improved due to the increasing number of experimentally determined protein structures reported in specialized data bases such as PDB. However, the accuracy of comparative model is still highly related to the percentage of sequence similarity between the target and template sequences. Comparative models

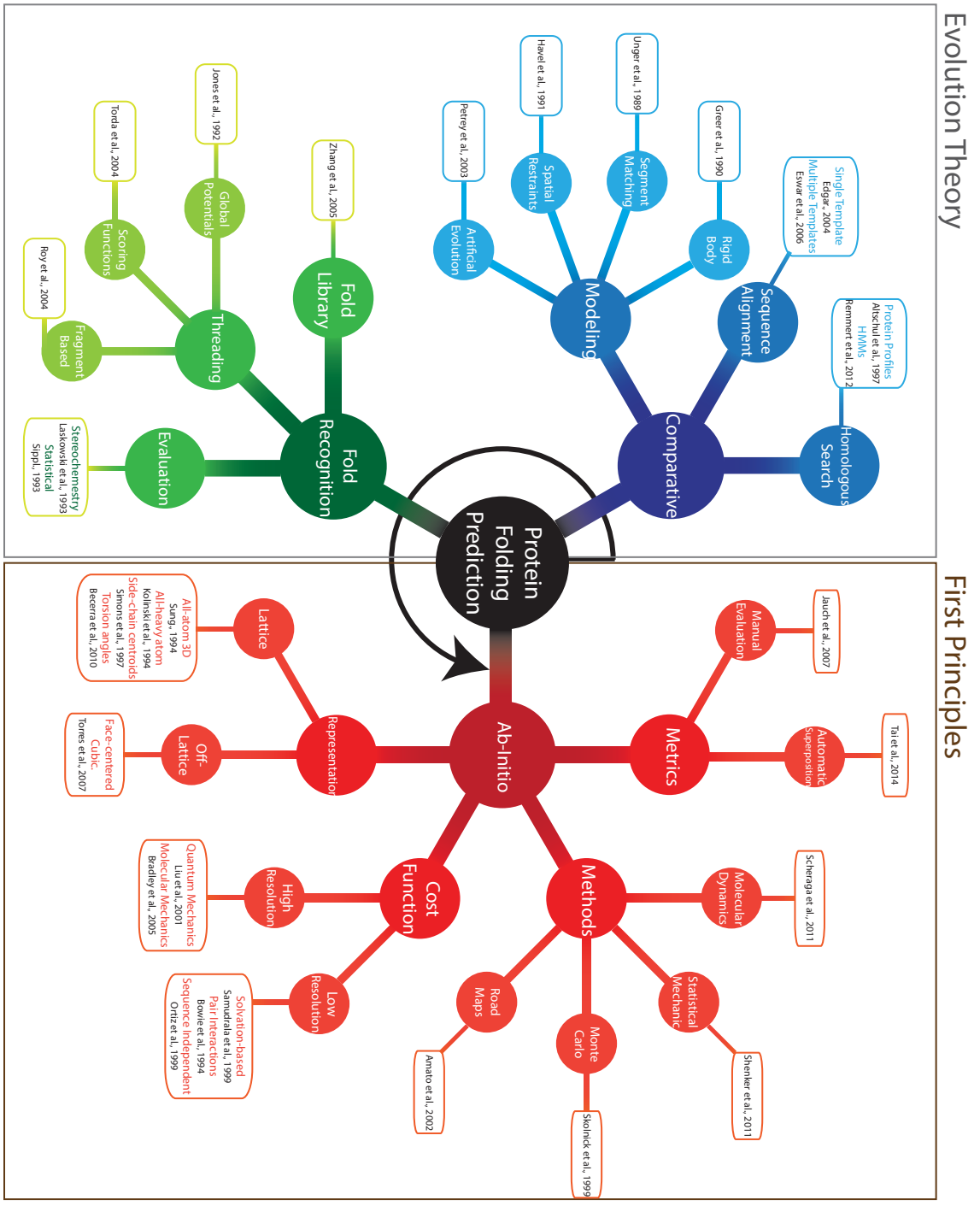

FIGURE 1.1: Mind-Map of the methods involved in the PF prediction problem
This taxonomy represents the splits made in the field to address two key scalability issues: *i*) the inherent computational complexity of the problem and *ii*) the fast-paced growing complexity of the simulation with respect to protein size and resolution. Each division is attached with one bibliography citation, which is an example of this division. The arrow in the centre of the map emphasizes the fact that the most accurate prediction tools integrate ideas from all three categories.

with a sequence identity $> 50\%$ to their templates can have a modeling accuracy of $1-2$Å measured by the root mean square deviation (RMSD) error for the native structure, which is comparable to the accuracy of a medium-resolution NMR structure or a low-resolution X-ray structure. Comparative models with sequences identities ranging from 30% to 50% often have 85% of their core regions modeled with an RMSD of $2-4$Å from the native structure, with errors mainly occurring in the loop regions. Loop regions are hard to be modeled using comparative techniques because they often correspond to unaligned regions in sequence alignments given their conformational flexibility and highly variable sequence. Finally, in comparative models for which the sequence similarity drops below 30%, the alignment errors rapidly increase the RMSD error and it is not advisable to use these methods for PSP.

1.2.2 Threading Methods

Fold recognition methods are based on the hypothesis that proteins often adopt similar folds despite no significant sequence similarity is found. This hypothesis assumes that protein structure is more conserved than protein sequence [67] and nature is restricted to a limited number of protein folds [68].

The general procedure of a fold recognition method consists of taking the AA sequence of a protein and evaluates how well it fits one of the known 3D protein structures or structural elements that have been experimentally observed. Construction of this sequence-to-structure alignment is not trivial, then, this process is usually complemented with scoring functions that determine the fit of a sequence to a given fold. The sequence and structural information are combined into a single-body energy term, which can be used in a dynamic programming algorithm (DP) for identifying the best alignment. The alignment (fold) with the best score is assumed to be the one adopted by the sequence.

Threading methods share characteristics with comparative and *ab initio* methods. It is similar to comparative modeling in the sense that both methods try to build a structural model by using experimentally solved structures as templates. The main difference with respect to the comparative modeling is that the prediction is determined by assembling

small structural components, where assembly is guided by an energy function. Then, threading methods (as *ab initio* methods) try to optimize a score function to measure the fit of the sequence in a spatial configuration.

1.2.3 *Ab initio* Methods

Comparative and threading methods are becoming increasingly accurate in predicting structures of proteins. Currently, the progress of those models is at a level of accuracy where such models can be routinely used to generate detailed biological hypotheses and are now capable of producing results directly comparable to experiments. These breakthroughs have been made possible due to knowledge based-methods that are able to identify relevant parts of extremely distant homologs and assemble them successfully to a protein target [69, 70].

Knowledge based methods are facing methodological limitations that are restraining their progress. Particularly, knowledge-based methods for predicting protein structures have been widely criticized because: *i*) they do not provide information about the mechanisms and forces that direct the formation of protein structures and *ii*) they can not be used when experimental resolved structures related with the target protein are not found, or the target protein has unique or different structural features to those characteristics that have been reported. *Ab initio* methods can be used on any AA sequence overcoming the inherent problems of comparative methods. Particularly, *ab initio* methods tries to directly predict the 3D structure (based on the first principle laws of physics and chemistry) without structural information of the target protein's family. *Ab initio* approaches are also able to discriminate between correct (native or native-like) from incorrect structures, and provide a deeper understanding of folding mechanisms.

There are three main issues to face in *ab initio* methods: *i*) The codification of protein conformations to reduce the high complexity of the conformational search without hampering the biological significance (represented in degrees of freedom) from the generated samples. *ii*) The formulation of an energy function (that works in the chosen low-complexity space created by the codification of protein conformations) to efficiently

model the different interactions contributing to protein folding and *iii*) The exploration of the space of possible conformations, which as mentioned before, can not be explored exhaustively. The following subsections deal with each of these tasks, and explain some of the most important proposed schemes for each task.

1.2.3.1 Conformation Representations

To represent protein structure conformations and overcome the high complexity in sampling, most methods need a significant reduction in complexity. Methods for reducing protein structure to discrete low-complexity models can be divided into two major classes: lattice and off-lattice models.

Spatial lattices or grids models can be used to represent the AA space and allow folding to have discrete degrees of freedom. Simplification in lattice proteins is performed by modeling each AA as a single 'bead' of a finite set of types and by restricting the location of each 'bead' to vertices of a (usually cubic-shape) lattice. These simplifications reduce the computational effort in the evaluation of energy functions and the sampling of conformations; however, the protein folding problem is an NP-complete problem even in this simplified scenarios [71, 72]. Lattice models are able to mimic the energy interaction between AA in real proteins by specifying an interaction 'energy' between neighbouring 'beads' (usually those occupying adjacent lattice sites). Lattice models have had a fundamental theoretical relevance given that they have allowed the efficient exploration of steric, hydrophobic and hydrogen bonding effects. For example, through lattice model simulations, Wolynes and coworkers [25, 73] postulated their hypothesis about the existence of a funnel-like energy landscape which guides the proteins toward their native structures (i.e., energy landscape model). However, lattice models can not be directly applied to predict protein conformations of real proteins, when carefully parameterized, they provide encouraging results [74].

Developing methods which are able to reliably predict native states has been a clearly necessity in the computational study of protein folding processes. In PSP the most used approaches have been based on off-lattice models. These models do not follow

a given lattice topology, but they fix the degrees of freedom and bond lengths of the polypeptide side chain. Although some algorithms use multiple representations, a few are commonly used: all-atom 3D coordinates, all-heavy-atom coordinates, backbone atom coordinates plus side-chain centroids, C_α coordinates, and backbone & side-chain torsion angles. A growing tendency in the community has been the development of atomic-level representations in attempts to improve the accuracy of the predictions. Direct simulation of protein folding using an all-atom model has been called the 'holy grail' of molecular biology because it tries to elucidate all the folding mechanism and the necessary protein processes to reach their native state[75]. Given the inherent complexity of all-atom simulations, there is still a need in the field to explore novel protein representations that produce a better equilibrium between the accuracy of the representation and its computational cost.

1.2.3.2 Scoring Functions

Once a protein representation model is chosen, a scoring energy or potential energy function that works in the chosen low-complexity space must be defined. The potential energy functions or force fields allow the evaluation of multiple proteins by returning a value for the energy based on the conformation of the protein. In other words, a potential function is an equation that relates structure to energy. The energy function must adequately represent the forces responsible for protein structure and should be efficiently calculated because it needs to be intensively computed while exploring the conformational space. To come up with some good functions it would be natural to use quantum mechanics, but it is too computationally complex to be practical in modeling large systems, then classical physics is a common approach to overcome the computational limitations. The main physical forces that drive a protein to its 3D folded structure (hydrogen bonds, the attraction of intermolecular forces between molecules known as van der Waals (vdW) interactions, backbone angle preferences, electrostatic interactions, hydrophobic interaction, AA's chain entropy) are modeled and described by the cost functions.

Traditionally, all atom energy functions have the form shown in equation 1.1, where V is the vector representing the conformation of the molecule (in Cartesian coordinates

or in torsion angles) and [*others*] refer to specific terms such as hydrogen bonding in biochemical systems.

$$E(R) = \sum_{bonds} B(V) + \sum_{angles} A(V) + \sum_{torsions} T(V) + \sum_{non-bonded} N(V) + [others] \qquad (1.1)$$

In general, equation 1.1 could be separated into two groups: *i*) the internal terms, including the bond, angle, and dihedral contributions, and *ii*) the non-bonded or external terms that include the electrostatic (or coulombic) and vdW terms.

The application of computer-based models using analytical potential energy functions within the framework of classical mechanics has proven to be an increasingly powerful tool for studying molecules of biochemical and organic chemical interest [76]. On the other hand, uncomplicated scoring schemes have also been proved to be functional in the prediction of protein structures [42], but they cannot be expected to consistently generate predictions with resolutions of better than $3 - 7\mathring{A}$ [77]. Then, there is still a need in the field to generate energy functions that can be applied in conjunction with novel search conformational methods to narrow the possible conformations (from an exponentially large number to a number small enough to create a tractably sized system).

1.2.3.3 Search Conformational Methods

The next task in determining a protein's native state is the search for the lowest energy conformation in a vast conformational space. Exhaustive exploration of conformation space is computationally intractable. Therefore, several algorithms that are currently used for PSP combine domain information with local search techniques to avoid the complexity of high-dimensional conformation spaces.

Simulation methods such as Molecular Dynamics (MD) assess the interactions between all atoms in a given system [78]. Specifically, MD methods generally simulate the motion of the atoms in the presence of thermal energy by numerically integrating Newton's equations of motion for the polypeptide chain. Evaluating the forces acting upon all

molecules at every time step of a folding protein is biologically accurate, but computationally expensive. Simulating the dynamics of a relatively small system for a nano-second timeframe can be an extraordinarily vigorous task for a typical industry-grade server machine. Thus, for complicated systems like proteins, canonical MD methods usually require a large amount of computational resources and their applicability is limited to small systems.

Given the impracticability and/or impossibility to generate all protein instances in the huge conformational space, Monte Carlo methods (MCM) use randomness to generate typical protein instances in an energy landscape. Unlike MD simulations, MCM are free from the restriction of solving motion equations. This lack of constrains allows MCM to generate different structural changes during the creation of new trial configurations. Although these moves may be non trivial, they can lead to considerable speedups in the sampling of equilibrium properties. Furthermore, MCM for protein folding are generally easily parallelizable [79]. MCM performs a series of randomly generated trial steps in the conformation space, each perturbing some degrees of freedom of the protein conformation. The MCM will accept a specific step (i.e., the MCM will move to the new conformation) based on a probability extracted from a desired distribution. In order to generate this distribution, the transition probability from an 'old' conformation to a 'new' conformation is controlled by the change in value of an energy function for these two conformations.

The determination of the probability density function of a protein system is a very difficult task with conventional methods (such as MCM). Statistical mechanical methods were originally conceived for modeling the behaviour of gas, but they have also been successful in studying protein folding and other problems in computational biology [80]. Statistical ensembles consist of states of a system with assigned probabilities, chosen to best represent physical situations. Statistical mechanic methods deals with statistical ensembles corresponding to equilibrium conditions [81]. According to statistical mechanics theory, one can characterize the composition of a system by taking advantage of the fact that a molecular state is in constant flux when at equilibrium, but that the proportion of molecules in each specific state remains constant. The main goal in a statistical mechanical treatment of protein folding is to determine the density function (i.e., the number of

protein conformations at a given energy level) of a specific protein system [82]. Particularly, in statistical mechanics, the thermodynamical properties of a protein under given environmental conditions are characterized by the computation of its canonical partition function.

State of the art methods for searching protein conformations are currently unable to compute, and even approximate, the complete 3D conformational landscape for all protein targets. The unfeasibility of adequately sampling the complete conformational landscape is accentuated by the lack of computational methods to accurately predict folding intermediates and pathways. Furthermore, the importance of pathway prediction to get valuable insights into the folding process and to guide the search of the conformation space have been neglected. We believe that the understanding of protein pathways will increase the level of accuracy of PSP methods whereby models could be routinely used to generate detailed biological hypotheses.

1.3 Protein Pathways Prediction Methods

PPP methods, motivated by Anfinsen and Levinthal's work, aim the identification of the native, transition, and denatured states, as well as intermediate states present in the folding pathway of a protein. PPP methods try to understand how an unfolded protein finds its native structure from the many possible conformations by characterizing all states along the folding pathway. Given the complexity of performing a complete characterization for all folding states, the problem of predicting folding pathways has been historically perceived to be harder than the PSP problem since the identification of folding pathways depends on first identifying the native state. Thus, the prediction of protein structures has received more attention than the pathway counterpart. The prediction of folding pathways greatly enhances structure prediction methods by providing valuable insights into the folding process and guiding the search of the conformation space. However, most PPP methods start from a known protein structure (i.e., the 3D structure). Currently, the information embedded in folding pathways remains largely underexploited by the PSP methods. The PPP problem is also interesting as protein

misfolding and aggregation have been associated with several pathological conditions, such as Alzheimer, Parkinson, diabetes, cystic fibrosis, and many others.

The search for folding mechanisms has driven major advances in experimental proteomics. In particular, sensitivity improvements have been obtained in the identification and characterization of intermediates states [83], mutational effects on folding rates [84], heterogeneity of folding and energy landscapes [85], fast temperature-jump methods [86], and more. In addition, protein model systems are the systems of choice in a vast body of experimental, theoretical and computer-based studies. Although experimental techniques have progressed over the years, our knowledge about protein pathways is still limited and does not provide an overall view of the protein folding system. A lack of reliable data prevents researchers from validating the protein folding prediction models. Corresponding advances in the theory and computation of folding pathways is also insufficient. There are few computational techniques for predicting protein folding pathways. Most of those techniques are limited by time and computational resources, or restrictive assumptions imposed during the modeling process. In the next section, we will review the current state of computational techniques used in the PPP problem.

1.3.1 Computational Methods for pathway prediction

Molecular dynamics (MD) is an invaluable tool to study protein pathways (such as structures). MD techniques aim to solve Newton's equation for obtaining coordinates and momenta of particles along (un)folding trajectories. These techniques provide information about the time dependence between inter-residue interactions underlying (un)folding pathways [49]. Using MD simulations to investigate protein pathways faces three main challenges. First, the timescale of protein folding and the computational expense required for adequate sampling. While proteins fold on the microsecond to millisecond timescale, most simulations are limited to nanoseconds. Even if folding events could be observed using conventional MD methods, we would be limited to observing a single folding event. Secondly, the accuracy of MD simulations depends on the underlying potential energy functions. The current generation of potential energy functions provides a good compromise between accuracy and computational efficiency. However, numerous approximations

are placed on them (based on the infeasibility to treat the protein systems using quantum mechanics) leading to certain limitations. For example, there are non-negligible interactions in vdW and hydrogen bonding which are due to non-classical electron interactions. Thirdly, the need for well-defined atomic coordinates of the denatured protein state to start a simulation. The heterogeneous nature and conformational uncertainty of unfolded proteins complicates the experimental derivation of structural coordinates for denatured states.

A Monte Carlo Method (MCM) is a common methodology that implicitly computes pathways and thermodynamic properties of proteins. It has been widely used to model protein folding dynamics. MCM allows for long-time dynamics of proteins to be obtained by properly designing local conformational transitions [48]. These simulations achieve results consistent with MD simulations. However, the design of an efficient model of MCM dynamics is not a trivial task. The main obstacle during the design of a MCM dynamic approach consists in that the local conformational transitions usually introduce very small changes in chain geometry and are commonly uncorrelated with the nature of the energy landscape. Trajectories obtained by MCM simulations are numerical solutions of a stochastic equation of motion, where a large extent of MCM dynamics are equivalent to Brownian dynamics[4] with a relatively long time step and large random force [40]. MCMs have been proposed for rendering simulations many orders of magnitude faster than molecular dynamics simulations, but simulations are still expensive if a custom-hardware supercomputer (i.e., a computer with a high level of computational capacity compared to a general purpose computer) is not used [87].

Some drawbacks of PPP methods based on classical MD/MCM simulations include the fact that they have no memory to recognize whether conformations have been visited in the past or not, they typically compute only one trajectory and have difficulties to escape local minima [88]. These deficiencies have led some researchers to look for methodological alternatives in other fields. For example, inspired by advancements in robot modeling, researchers began to adapt robot motion strategies to explore the conformational space of

[4]Brownian dynamics (BD) is a simulation technique used to describe the motion of molecules in molecular systems. In this technique, trajectories and interactions between key molecules are calculated directly, while other components of the system (e.g., solvent molecules) are replaced by a stochastic force

proteins. Given a description of the environment and a robot, motion planning in robotics consists of finding a feasible path that takes the robot from a given starting point to the goal. This motion planning has been applied to the protein folding problem by replacing the traditional collision-free constraint in robots to an energy minimization goal for proteins [89]. Probabilistic roadmaps methods (PRM), an example of robotics motion planning, in protein folding capture the connectivity of the high dimensional space via random sampling through the use of a graph. The root of this graph (i.e., starting point) is the unfolded structure of a protein. The goal configuration is the most energetically favourable conformation (i.e., native state). PRM methods start by building an approximate map of a protein's potential energy landscape. This map contains thousands of feasible folding pathways to the known native state that enable the algorithm to explore the global landscape. To accomplish the traversal task, PRMs proceed as follows: first, points are sampled in the protein's conformation space. This sampling procedure is biased to increase density near the known native state. Next, these points are connected to form a graph (i.e., the roadmap). Weights are assigned to directed edges of the graph to reflect the energetic feasibility of transition between the conformations corresponding to the two end points. Finally, folding pathways and structures are extracted from the roadmap using standard graph search techniques (e.g., Dijkstra's shortest path algorithm). Probabilistic and Stochastic Roadmaps are able to predict intermediate configurations on the folding pathway using a reasonable amount of computer resources. The protein sampling process is highly hampered in these approaches due to the need of *a priori* native conformation, algorithmic inefficiency due to the size of the configuration space, and a lack of biological significance from the generated samples.

The folding pathway of a polypeptide has also been modeled through knowledge-based models. These pathways are modeled by the hierarchical adhesion of structural fragments. The adhesion of the fragments simulate the hierarchy of folding events happening in nature. Each fragment represents a structural conformer for a segment of the AA sequence, usually defined by sequence statistics or motif patterns. The adhesion of fragments can be roughly described as a local to global hierarchy, where the formation of local structures simulates the earliest steps in folding [90]. Knowledge-based models are able to predict

conformationally stable and early-folding motifs with high-confidence. Mutations within these regions have been shown to have dramatic effects on the folding process [91].

One alternate approach to enumerate folding pathways, due to the availability of known experimental coordinates, is to start with a folded protein and unfold the protein in an ordered sequence of steps to its unfolded state [92, 93]. Given that folding microscopic reversibility hypothesis (i.e., an hypothesis that suggest that the unfolding process of a protein is a true reversal of the protein folding pathway) has been confirmed for some proteins under identical conditions [94], the reversal of an AA sequence may lead to a plausible protein folding pathway. The structural features of the native state are generally coded in a pairwise distance matrix, structural graph, or contact map. The idea of this methodology is to integrate the protein topology and structural information of intermediates states during the prediction of the most likely unfolding events. Simplified models for unfolding pathways assume that non-native contacts are off pathway and not essential to the folding process, such that only native interactions are considered during the simulation process.

Clear structural information on the intermediate states that bridge between the unfolded and native states (i.e., folding pathways) is required to acquire a clear identification of protein structure and function. Due to the inability of experimental techniques to elucidate this intermediate states in great structural detail, it has been fundamental to explore computational techniques to model such transitions and extract information related to folding pathways. Simulating protein folding pathways has been a very difficult task performed on small structures through computationally expensive methods. Coarse grained models have allowed the study of folding pathways on larger proteins, but these methods usually introduce constrains that limit the biological significance of the simulations. The balance between the information obtained and the resources conferred to obtain a global albeit coarse view versus a local but detailed view of protein folding is still an open discussion. In the following section, we will face this discussion by studying the scalability challenges of PF methods.

1.4 Scalability challenges of protein folding prediction

Anfinsen's thermodynamic hypothesis describes the native conformation of a protein as the most thermodynamically stable conformation within a particular environment. Based on this hypothesis, the AA sequence is all that is needed to know the order in which a protein folds into its biologically active shape because the AA peptide chain determines the pattern of folding [95, 96]. This hypothesis has fuelled efforts to computationally predict protein 3D structures from sequences, but these computational methods are currently unable to compute, or even approximate, complete 3D conformational landscapes. Then, one might ask, 'Why has no one been able to put his hypothesis into practice?'. Current computational approaches are highly hampered by two scalability issues: *i*) the inherent complexity of modeling the physical systems. The protein folding problem is an NP-complete problem in simple lattice models [71, 72], where simulations are limited by the required amount of time and computational resources. *ii*) The computational complexity of the simulation, which grows much faster than the size and resolution of the simulation. The energy functions devised to represent the protein energy landscape limit the size and resolution of protein simulations due to the unfeasibility of an adequate sampling to provide a complete conformational landscape. Also, proteins of interest fold on the microsecond timescale, where free energy calculations typically sample the nanosecond scale [97, 98].

There is an interplay between the scalability capacities of the predominant protein folding approaches described in previous sections. Detailed models (typically working on full atomic detail) have the obvious benefit of potentially greater accuracy. However, the computational demands and the increasing complexity of the simulations restrict their applicability to most protein systems (figure 1.2 plots the scalability capacity of predominant methods in the field given the size of a system in AA and the level of abstraction utilized to solve the PF problem). Some techniques such as replica exchange molecular dynamics [99, 100], tightly coupled molecular dynamics [101], sequential stabilization [87], and template-based threading [102] have become powerful approaches to explore protein

landscapes in a faster fashion compared with traditional MD and MCM. Additionally, different approaches have been proposed to handle the computational demand by deploying significant computational resources either to parallelize the simulations [79, 97] and to run a large number of multiple independent trajectories [103]. Custom hardware approaches with GPU simulations [104, 105] and special-purpose machines [106] have resulted in tremendous acceleration of simulations. Coarse-grained models have also played a fundamental role in protein folding research by allowing the simulation of biological systems relevant in size and timescale. These models typically make simplifying assumptions about the folding process [107], the protein representation [74], or both, to potentially yield insight into protein folding.

The complexity of today's computational molecular systems is a consequence of growing molecular system size, atomic granularity level requirements, longer simulation time-frames, and surrounding environment (i.e., solution types). New technologies aimed to study or work with molecular complexes need to take scalability facets into their system design and offering. Relying on hardware innovation and improvements to accelerate current algorithms is not a viable option to address problems in this field. Although faster processors and devices with larger memory are always met with great enthusiasm, alone they do not tackle the growing scalability issues in the field. Scaling molecular system size, while keeping the same level of granularity and expecting longer time-frame simulations, will have to come with new heuristic techniques, smarter algorithms, innovative modeling methods, and load distribution over a network of servers. In the following subsections, two of the most promising techniques to tackle the growing scalability issues are described.

1.4.1 Residue Contact Information

The idea of predicting residue contacts by co-evolution-based strategies has received a new twist due to new methodological advances and the increasing availability of protein sequences. Breakthroughs in the handling of phylogenetic information and disentangling indirect relationships have resulted in an improved capacity to correctly predict inter-residue contacts [108, 109]. It is still unclear what accuracy, coverage, and distribution of

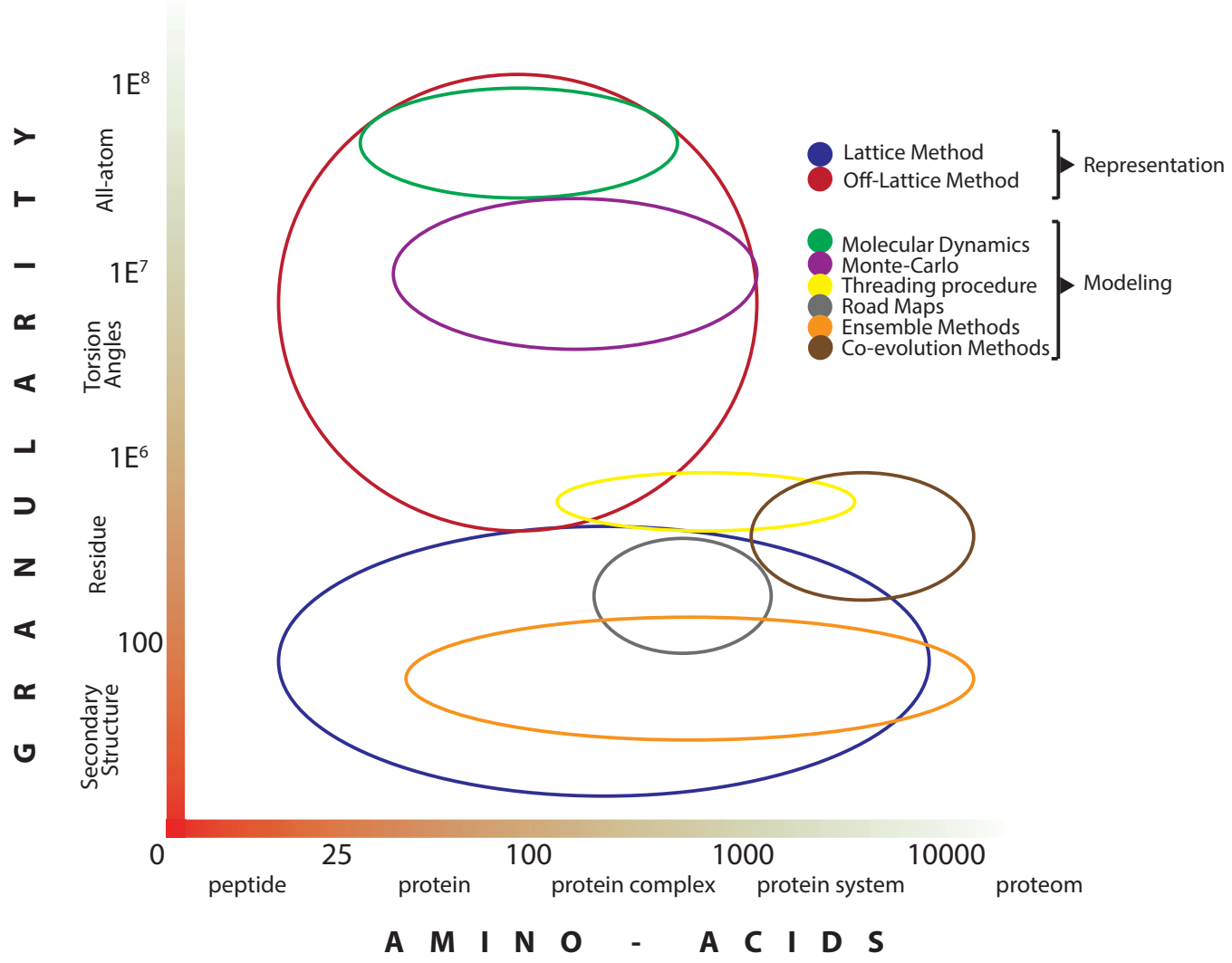

FIGURE 1.2: Scalability capacity of predominant computational structural molecular biology methods.
Overcoming scalability barriers involves the ability to deal with an increasing system size (AA) and levels of abstraction (granularity). Methods confronting scalability issues extend the limits in either direction, or both.

contacts along the sequence are needed to be useful in practice for such methods. However, predicted inter-residue contacts have already been used to increase the scalability capacities in different PF approaches [110]. For example, homology and fold recognition methods (see section 1.2.1 for more details of these methods) reduce the conformational space by filtering the most likely structural models based on predicted residue contacts. Regarding *ab initio* methods, residue couplings have been translated to a set of distance constraints for effective use in geometry generation of 3D structures and their refinement

by energy minimization and molecular dynamics methods [111, 112]. This new generation of contact prediction tools is enticing, but it is currently limited by the amount of rich evolutionary sequence data needed to obtain reliable models of structure. For accurate predictions, these methods require large numbers of homologous sequences using accurate alignments that are sufficiently diverse to reveal co-evolution patterns that cover most structural elements of the protein [109]. Work in this fundamental area of research focuses not only on predicting contacts with high accuracy to be used in structure modeling, but on building a reliable structure from incomplete/inaccurate contact data [113].

1.4.2 Ensemble Modeling

Current protein structure databases have accumulated and organized data following a singular one-to-one relationship between protein AA sequences and their 3D structures (e.g., the PDB [114]). PF prediction methods are usually constrained by a single-sequence/single-structure perspective during the modeling and validation phases of their respective methods. However, proteins have the ability to adopt many different conformational states *in vivo*, where multiple substrate minima could exist (each with different functional properties) [115]. These ensembles can be obtained by MD or other local deformation methods. These time-consuming high-resolution prediction methods are constrained to relatively small molecules (e.g., proteins with less than 50 AA). Given that these methods are only able to observe local variations, multiple simulations are required to construct an accurate set of ensembles. Newer computational modeling approaches, called 'ensemble methods', no longer search for an individual lowest energy structure, but rather aim to predict an ensemble of protein conformations and pathways to describe a more realistic landscape of conformational variants (without sacrificing efficiency or accuracy).

Ensemble methods are conceptually orthogonal to the mentioned approaches (see Figure 1.1) because they make use of both thermodynamics and fold similarities to describe the conformational landscape [116, 117]. Ensemble methods are able to address the complexity barrierof PF simulations by computing coarse grained (i.e., simplified) representations of complete energy landscapes at a large scale. These landscapes are then used

to simulate the dynamics of protein pathways. Ensemble methods unify the prediction of protein structures and pathways, instead of focusing solely on the native conformation.

Ensemble algorithms employ a coarse-grained structural model and dynamic programming to enable efficient computation of the complete conformational landscape. These algorithms allow for the application of statistical mechanics techniques to produce representative sets of structures. Coarse grained representations of complete energy landscapes at a large scale are used to simulate the dynamics and prediction of protein pathways. Ensemble methods in proteins were inspired by structural ensemble prediction algorithms that allowed for the accurate computation of RNA secondary structure energy landscapes from sequence information alone [80, 118, 119]. These RNA approaches may not be directly mapped to proteins, but they have been shown to be an excellent starting point to model complex scenarios more accurately. Ensemble methods have shown to be successful when used in combination with *ab initio* models [116], comparative models [120], and protein pathway prediction [121] of $\beta-$ proteins. This approach to the PF problem is able to describe a more realistic landscape of conformational variants allowing the study of larger systems such us amyloid structures [122].

1.4.3 Growing complexity of protein aggregation models

Proteins do not always fold into the native state. Under many conditions, proteins have been observed to mis-fold into abnormal structures that aggregate into large complexes, which play a role in more than 20 diseases [123]. Computational cost has limited the ability of PF algorithms to extend simulations beyond a singular, small protein. In order to understand the folding behaviour of more complex systems, such as protein aggregates that occur in living cells, the limits and scalability features of molecular computational methods must be refined to go beyond single molecule predictions. Shifting from single molecule experiments to modeling a system of biological events and molecular complexes will dramatically increases the complexity of computational methods [124].

The field of protein aggregation is one of the contemporary fields of research that is extending the limits of existing computational techniques by developing novel solutions

to high complexity and computational resource limitations [125]. Mis-folded proteins, known as amyloid proteins, appear to aggregate in many neurodegenerative diseases. Their beta-sheet rich structure compositions are believed to aggravate medical conditions and cellular toxicity [126]. Most computational studies of amyloids have been limited to studying the nucleation phase due to scalability limitations in modeling and simulation involving molecular complex size, atomic granularity, and experiment duration [127]. Recent computational studies have attempted to accurately model these large molecular aggregate systems at atomic level detailed-precisions by introducing a novel classification model that utilizes molecular symmetries to factorize and discretize the search space involved in the process of amyloid aggregation [128]. This novel method coincidentally introduces a few problems. Modeling some of the shortest amyloid aggregates for testing usually exceeds the atom and molecule limit in many cases. The side effects of such methods include a potential loss of support to most existing computational tools that support PDB files. Also, studying the dynamics of growing systems requires a substantial increase in computational power for even the smallest molecular systems. With large amyloid aggregates, the dynamics has been easier to model due to the symmetry of the aggregate filaments around their fibril axis. Determining how a few of the monomers interact may be sufficient knowledge to construct the entire aggregate structure of these types of proteins.

In addition to distributing computational tasks to improve running times, the advancement of energy models and force fields is key to increasing the scalability for the growing size of systems. It was not an uncommon practice in the field to run MD productions in a vacuum solution environment to reduce simulation complexity and running times [129]. With improvements in hardware and processing speeds, researchers have moved from vacuum solutions to the currently widely accepted implicit solvation methods to represent environment solutions [130]. Implicit models do not perfectly account for hydrophobic effects, viscosity, and solubility of ions, but they are considered acceptable heuristic approximations that reduce the complexity of molecular systems and allow for quicker simulations. However, even with the MD implicit models, calculating a term such as the 'solvation' energy (i.e., the enthalpy released when a solute dissolves in solvent) of an amyloid aggregate system is impractical [131]. Solvation energy of a molecule is

calculated by computing the Poisson-Boltzmann (PB) second order, elliptic, nonlinear partial differential equation. This method quickly increases in complexity as the number of molecules increase in a system, making exact computation unattainable. One solution to this problem was introduced by the solver for the dipolar Poisson-Boltzmann-Langevin equation called AQUASOL [132]. The AQUASOL solution involves using a dipolar solvent model and dissecting the problem into subproblems by using a lattice gas model to calculate the partial differential equation. Subproblems are able to be isolated and solved separately, allowing for parallelization and faster computations.

Computational methods have been critical in advancing protein (mis-)folding research. In light of the ever-growing necessity to perform research on larger molecules and on systems, it is necessary the development novel methods who re-evaluate the limitations, bottlenecks, and scalability capacity of current methods. These new methods must explore some of the major limitations in the protein folding field and outline current promising strategies to overcome the scalability gridlock.

1.5 Thesis roadmap

1.5.1 General thesis contributions to the field

The PF problem is considered to be one of the most compelling scientific challenges facing researchers today. Furthermore, the PF problem has been named the 'holy grail' of modern biological research and one of the 125 big questions that face scientific inquiry over the first quarter-century of the 21^{st} century [133]. For decades, scientists have studied the complex processes that determine the folding of proteins. However, a solution is still far from being achieved. This thesis contributes to the field by addressing the main difficulties faced by PF methods.

1.5.1.1 Contributions to the general view of the problem

The PSP and PPP problems have been widely acknowledged as open problems given that our knowledge and methods are still inadequate to effectively have an overall view of the protein system. The prediction of protein structures have received more attention than their pathway counterpart because the latter requires a full understanding of the folding process (a task perceived to be harder). It is clear that the ability to predict folding pathways can greatly enhance structure prediction methods, but the importance of pathway prediction to provide valuable insights into the folding process and guide the search of the conformation space has been neglected. A successful algorithm for describing folding pathways should enable predicting both the protein pathway and structure (two intertwined issues that generally have been treated separately), but many of the state of the art PPP methods require *a priori* knowledge about the native structure of the protein (i.e., the native 3D structure must be know before starting the PPP simulation). Considerable work to understand folding intermediates via molecular dynamics and experimental techniques has been completed, but there is an increasing need for novel PPP computational techniques. Most PPP techniques are limited by the required amount of time and computational resources, or the restrictive assumptions imposed during the modeling process. Currently, one piece of the PF puzzle is missing (i.e., the PPP is neglected in the PF problem), and that one piece is crucial to completing the true picture of PF. We believe that understanding protein pathways better will increase the level of accuracy of PF methods.

1.5.1.2 Methodological contributions

Functional proteins are known to undergo natural selection processes that preserve their function and structure. The preservation of natural folding dynamic properties enables proteins to fold quickly from an unfolded state to the native structure. These properties describe an energy landscape that has been molded by evolution, such that the native protein structure and set of folding pathways are conserved [134]. We believe that a description of the true underlying protein energy landscape may be attained by including:

i) protein structure and pathway information and ii) statistical analysis of physical predictions (i.e., low free-energy models) and evolutionary based predictions (i.e., sequence variation methods). A new set of tools for understanding the connection between protein structure, folding, and function is required. We believe that combinatorial algorithms for protein ensemble modeling and co-evolution-based strategies to predict residue contacts provide a solid basis for this toolset.

Early theories of protein folding envisioned a singular mapping between a protein's AA sequence and its 3D structure/folding pathway. Modern databases have accumulated and organized data that supports this perspective. As a result of this design, computational prediction tools for protein folding have been forced to adhered to a single-sequence/single-structure model. A growing body of evidence indicates that proteins exhibit a variety of folding pathways simultaneously [135, 136], where some paths are expected to be more populated than others. It is important to recognize the statistical implications of the protein folding process and consider that protein ensembles may mimic the ability of other proteins to adopt different conformational states *in vivo*. Ensemble algorithms, proposed in this thesis, address our hypothesis by predicting a statistical distribution of topologically allowed pathways through the use of the Boltzmann probability function and simulation of population dynamics to statistically characterized the protein ensembles. To the best of our knowledge, this is the first time in the literature that residue contact information has been integrated into the Boltzmann sampling process of ensemble methods for the purpose of predicting protein folding pathways.

1.5.1.3 Contributions to scalability issues

Current computational approaches are hampered by two scalability issues: *i*) the inherent complexity of modeling the physical systems and *ii*) computational complexity of the simulation, which grows much faster than the size and resolution of the simulation (PF is a NP-complete problem in the simplest versions). Indeed, an approximate solution to the PF is hard to find given the required amount of time and computer resources. Additionally, the size and resolution of protein simulations are limited by the inaccuracy of the

energy functions devised to represent the protein energy landscape, and the unfeasibility of adequate sampling methods to complete the conformational landscape.

In this thesis, we aim to construct efficient and effective computational methods to predict protein structures and pathways given an AA sequence. We introduce a general computational framework that enables efficient calculation of complete energy landscapes by coarse grained models. These models predict protein structures and pathways using only the primary sequence as input and, when available, evolutionary sequence information. Evolutionary sequence information is integrated into a Boltzmann sampling process to circumvent the limitations of potential energy scoring schemes and narrow the conformational search space (the two most important bottlenecks in protein folding prediction). The ensemble framework presented in this thesis represent an improvement in performance (i.e. speed and accuracy) with respect to state of the art methods. In principle, the proposed approach predicts secondary structure energy landscapes from sequence information alone and it does not require any *a priori* knowledge of the native protein structure (in the case of pathway predictions) and known secondary structures (in the case of residue contact predictions). This characteristic is a differential feature of our approach with respect to most of the state of the art predictors. The need of *a priori* information by other PSP and PPP predictors impose restrictive assumptions and make unclear how those predictors would perform in practice when that information is not present.

The ability of the proposed framework to formulate quick, coarse-grained predictions in a matter of minutes or hours, rather than days of atomistic-detail simulation, is an attractive approach to predicting many folding routes and transition states for protein sequences. Our approach can be used to support the initial stages of more complex and detailed models such as folding diseases, drug affinities, membrane proteins and disorder proteins. We believe that the proposed framework will allow for large scale studies of folding dynamics and the annotations of proteomes. Our method represents an alternative to high computational cost approaches because it includes the predicted pathway during the integration process of sequence comparison and fold recognition. The

proposed framework crosses traditional borders of structure prediction and we hope to provide meaningful biological inferences.

1.5.1.4 Availability contributions

There are numerous internet services which are available to predict protein structures and functions (i.e., PSP web-services), but there is a lack of online servers to predict protein folding pathways (i.e., PPP web-services). Many pathway prediction algorithms, present in the literature, are built on complex implementations that are generally available to users via source code. Attempts by structural biologists or experimentalists to generate pathways predictions, before embarking on time-consuming experiments and simulations, are often hampered by the lack of a system to generate, manage, store and analyze pathway predictions. Thus, potential researchers are faced with the daunting task of generating, evaluating and validating the predicted pathways. The proposed algorithm framework in this thesis contributes to the field by developing a web-service that can present data to the user community in both a human and machine readable manner, where the proposed suit of algorithms is able to support large-scale protein simulations and single experimental designs.

Simulating protein folding through statistical ensemble methods is another way to gain new insight into experimental problems (i.e., wet-lab experiments). These methods can unravel and suggest hypotheses (about how proteins fold) that are hard to explore experimentally. Ensemble techniques construct better experimental designs and suggest new interpretations of *in-vitro* experiments. Despite their relative immaturity, ensemble methods have already begun to influence our view of protein folding. However, to make ensemble-based approaches a reality, plausible protein folding pathway predictions and an appealing information system to easily administer those predictions must be developed. In this thesis, we describe a computational pipeline that conveys the content of the folding pathway through an interactive exploration of the network data. This tool aims to provide the means to disseminate the protein pathways/structures and comprehend their biological importance.

1.5.2 Thesis outline

A schematic pipeline of the proposed computational framework and thesis outline can be seen in Figure 1.3. *Chapter 2* analyzes the input of the proposed framework. Special emphasis is given to the analysis of evolutionary information encoded in Multiple Sequence Alignments (MSA). Two novel approaches will be proposed and analyzed to improve the quality of MSA by humans and algorithms. *Chapter 3* provides an in-depth review of the methodology used to model protein ensembles and sequence information. *Chapter 4* analyzes the results obtained by the proposed algorithm framework. *Chapter 5* reports the proposed visualizer of protein folding pathways. Finally, *Chapter 6* concludes this thesis with an insightful discussion of the previous chapters.

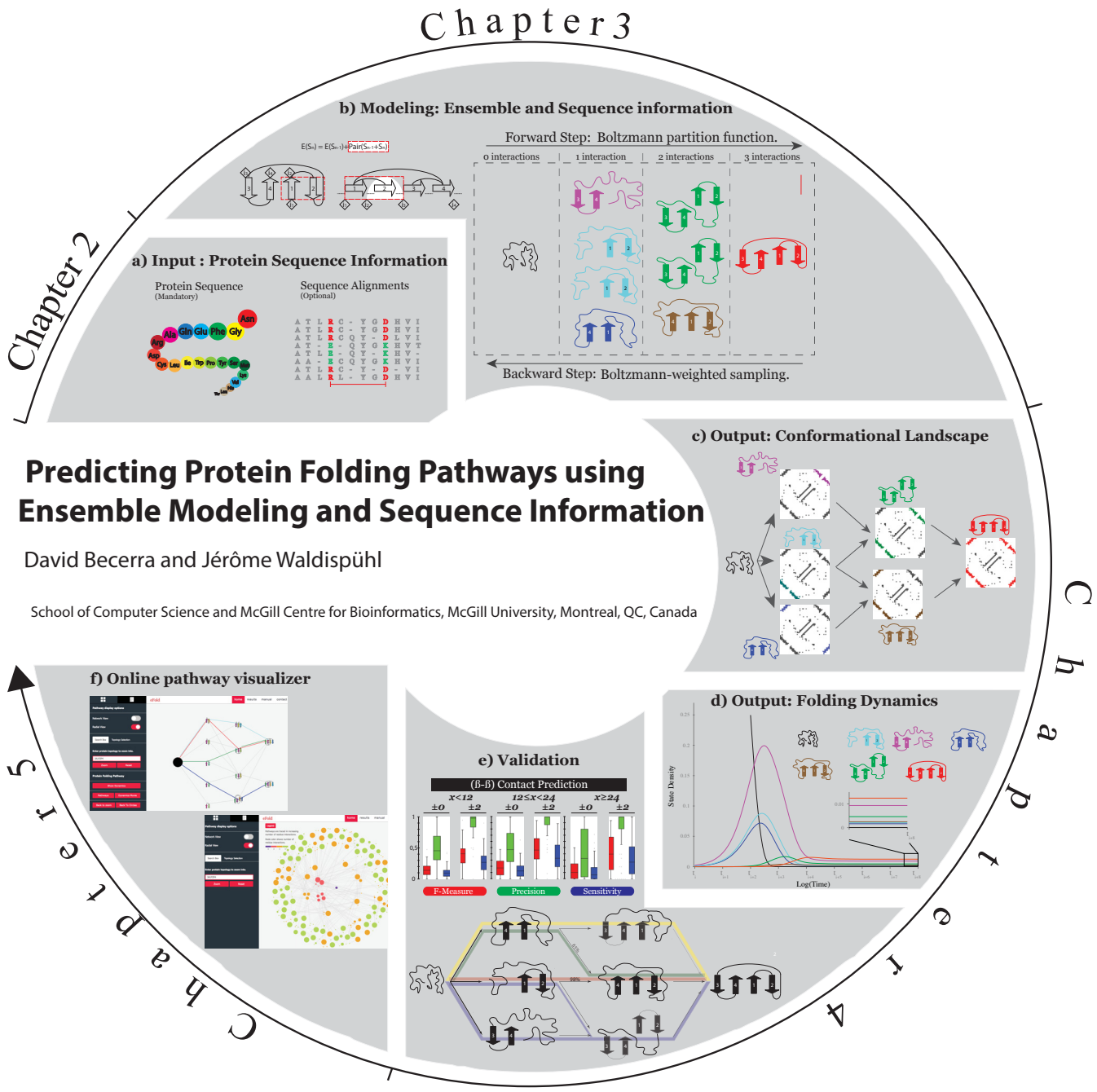

FIGURE 1.3: Framework to predict protein folding pathways using ensemble modeling and evolutionary-based information.

a) Input consists of a single AA sequence, a set of parameters controlling the size and complexity of the conformational landscape to be explored and an optional multiple sequence alignment (MSA) of the input sequence with homologous proteins. **b)** The ensemble technique to predict β-sheets structures consists of a forward and backward traversal over the data structure that models the hierarchical folding mechanism and stores all possible proteins states characterized by an energy objective-function. **c)** The conformational landscape is represented as a graph, where nodes represent clusters of energetically accessible conformation states and edges model the presence of structural similarity between the states. **d)** The dynamics of the system are calculated by treating the folding process as a continuous time discrete state Markov process. **e)** The transition from a random coil to the native state is represented as a path in a graph (or flow network) of varyingly folded protein conformation states. The predicted structural conformation is coded as contact residues. **f)** The proposed proposed visualizer of protein folding pathways

Chapter 2

Work on Multiple Sequence Alignments

2.1 Multiple Sequence Alignments

The Multiple Sequence Alignment (MSA) problem refers to the computationally hard problem of aligning two or more sequences and identifying the evolutionary and/or structurally related positions. In general, a MSA arranges sequences into a rectangular array such that residues in a given column are homologous (i.e., derived from a common ancestor, either through faithful inheritance or through substitutions), superimposable (i.e., they can be placed on top of a rigid local structure and they will occupy the same position) or play a common functional role. Assembling a suitable MSA is a computationally intense and biologically complex task because computing exact MSAs is a NP-hard problem under most reasonable scoring schemes [137, 138]. In practice, approximate algorithms (heuristics) are used to align sequences, by maximizing their similarity. The accuracy of a MSA is of critical importance due to the fact that many bioinformatics techniques and procedures are dependent upon MSA results, where inaccuracies in the alignments have been shown to limit the accuracy of downstream analyses.

Despite massive research efforts aiming to solve the MSA problem and its variations, this problem remains an active area of research with many studies focused on developing

faster and more accurate algorithms [139]. Novel computational methods that improve the accuracy of MSA tools have been applied to next generation sequencing [140], genome-annotation [141], correction of protein multiple structural alignments [142], structural and functional prediction [143], phylogenetic studies [144] and data base searching [53]. The creation of a MSA is also an ubiquitous task in PF prediction, and has a wide variety of applications including homology detection [35, 145], fold recognition [68], residue coupling prediction [143], and SS prediction [146] to name a few. All of these applications depend on the correct alignment of thousands of diverse sequences to predict protein structural information. However, these applications do not account for alignment uncertainty (treating the input MSA as free from errors and unbiased). Moreover, biological alignments (such as the ones reported in Pfam) have not been widely assessed in MSA benchmark studies leaving their validity to be questionable [147]. Given that the folding pathway predictor proposed in this thesis might use a MSA as input, we investigate if one should trust the accuracy of MSAs to improve the accuracy of pathway predictions. We expect that future work will provide guidelines toward the best methods for adapting MSAs for their intended use with PF prediction methods [145].

Progressive alignment [148] is the most popular heuristic used by MSAs. Progressive alignment builds the final MSA by 'progressively' completing a series of pairwise alignments on successively less related sequences. The progressive heuristic is greedy in nature (it only looks at the two most related sequences at a given time) and therefore cannot guarantee an optimal solution (nor indicate how much the solution presented differs from the optimum). Mistakes made during the initial stages of the alignment process propagate to later stages and the effect of these mistakes is increased as the number of sequences increase. Progressive alignment is the foundation of several popular algorithms such as MUSCLE [54], T-Coffee [149], Clustal [150], ProbCons [151], PRANK [152], MAFFT [153]. However, more phylogenetically-aware scoring schemas have also been proposed [152, 154]. MSAs may also be constructed by incorporating structural information into the alignment [155].

MSAs belong to the family of combinatorial optimization problems with exponential time complexity. Given a set of sequences, most mathematical formulations of MSA aim to

identifying a maximum-scoring alignment over the set of all possible MSAs. This score (or cost) function indicates which alignment (over the set of all possible MSA combinations) is the best by quantifying a distance between (set of) sequences. This scoring assumes that the presented MSA with optimal cost is also the best explanation of biological relationships between the sequences (ideally, the maximum of the optimization criterion should coincide with the 'true' biological alignment). However, none of the existing MSA methods have yet delivered MSAs that are perfectly supported by biology and there is no consensus on a strategy that produces optimal results [156]. Each state of the art algorithm has its own advantages and drawbacks when facing a particular set of sequences, which can make it difficult to make a rational selection of an appropriate alignment tool. Some methodologies have recently been designed to combine distinct MSA algorithms to obtain improved consistency with a final alignment [157]. Some optimization and computationally intelligent techniques have been applied to assemble sequences aligned by state of the art algorithms predicting the expected accuracy of each alignment [158–160].

The cost functions used by state of the art MSA algorithms make explicit assumptions upon which the combinatorial optimization is based on [161]. These assumptions can be formally studied to determine the biological and mathematical accuracy of the resulting alignments. The main objective of these studies is to improve the accuracy of MSA by providing insights that can not be entirely replicated by the cost functions used in heuristics-based algorithms. In this chapter, we analyze MSAs that have been improved by humans (see section 2.2) or state of the art algorithms (see section 2.3). We use common MSA programs to generate an initial MSA and then improve the mathematical and biological accuracy by one of the two proposed methodologies.

2.2 Multiple Sequence Alignments improved by Humans

Crowdsourcing is the practice of getting contributions (usually online) to a task or project by enlisting the services of a crowd of people. Thanks to our growing connectivity (i.e., new internet technologies and social media), it is now easier than ever for individuals to collectively contribute to a common goal, project, or cause (whether it be with ideas, time, computer resources, problem solving, expertise, or funds). Citizen science is a narrower subset of crowdsourcing, in which the members of the crowd (best known as citizens) actively participate and contribute (gathering, evaluating and/or computing various scientific data) in a scientific research project [162]. Projects often take a variety of formats and the task required of the citizen scientists may vary in its level of complexity. For example, in distributed projects, the only activity required by the citizen is the installation of software that will automatically analyze the 'scientific' team's data. Other projects require a greater cognitive involvement and citizens may be required to annotate or classify data. In other projects, users are motivated (usually through a performance ranking) to play and compete in a multi-player computer game that involves a repackaged scientific problem [163]. Bioinformatics is one scientific field that has widely benefited from citizen science (and online citizen science games). Several studies have taken advantage of emerging approaches for harnessing such distributed human intelligence encompassed by citizen science to address molecular biology-level problems that may benefit from human involvement. For example, tasks like genome annotation and alignment, image analysis, knowledge-base construction and protein structure determination have benefited from citizen science studies [164].

The MSA problem is one of the most fundamental question in computational biology and it is involved in many downstream analyses performed in molecular biology. MSAs are at the core of most comparative genomics and proteomic studies because they are able to derive molecular function based on the evolutionary patterns of molecular sequences. Bioinformaticians typically rely on algorithms (which are based only on statistics) to align molecular sequences (i.e., the MSA problem) by finding molecular subsequence that

match between samples. Computational solutions are not guaranteed to be optimal (see section 2.1 for further details) and researchers sift through the data manually looking for local inaccuracies to improve the predicted alignments. Researchers have been known to use human pattern recognition skills to identify problematic areas in the MSA. Visual pattern recognition skills are not a consequence of a researcher's academic preparation, but are an inherent skill of human vision. In principle, we can take advantage of human visual intelligence to identify problematic areas of MSAs given by algorithms. We can benefit from hundreds of thousands of players willing to play a crowdsourcing-science online game to help researchers in the identification of these problematic MSA areas.

`Phylo` is a citizen science framework designed to solve MSA problems. `Phylo` uses human pattern recognition skills and comparative genomics to address the MSA problem [165]. `Phylo` is a horizontal tetris-like game, where players attempt to match up four coloured blocks in a column that represent the four nucleotides of the genetic code. A web-browser displays the nucleotide blocks inside a matrix of up to 24 columns and 8 rows (a problem size beyond the capacity of exact MSA algorithms) woth its associated phylogenetic tree. Each row stores the genetic sequence of different species. By moving blocks horizontally, players try to create columns with identical colors (maximizing conservation across rows) while avoiding gaps when possible (see figure 2.2). Much of the science behind `Phylo` is hidden from the player by turning the NP-hard MSA optimization problem into a casual game (i.e., a simple game targeted at a mass audience of casual gamers). `Phylo` has a broad spectrum of participants that have varying degrees of genetic knowledge. These players make contributions regarding the determination of phylogenetic relationships, the impact of mutations, and their potential role in disease without knowledge of the underlying phylogeny.

`Phylo` aims to harness the intelligence and processing power generated by crowds of online gamers to solve the MSA problem. However, the selection of the data to be analyzed through `Phylo` is under the exclusive control of the game designers and so are the results produced by gamers. Even if exact methods can not be applied on MSAs with sizes similar to those used in `Phylo` [166], the question remains whether `Phylo` could increase the size of the puzzles while maintaining the playability of the game. To address these

concerns, a new model for human-computing platforms, one that is powered by the public and is open for the public was developed. `Open-Phylo` is an open and freely accessible web interface that enables scientists to enter their own sequences into our system and manage the efforts of the crowd toward aligning them. In addition, we developed an advanced version of the game[1], where advanced players can align larger MSAs (up to 300 nucleotides long). This allows `Open-Phylo` to benefit from the skills of the most experienced users more efficiently in solving the hardest MSAs.

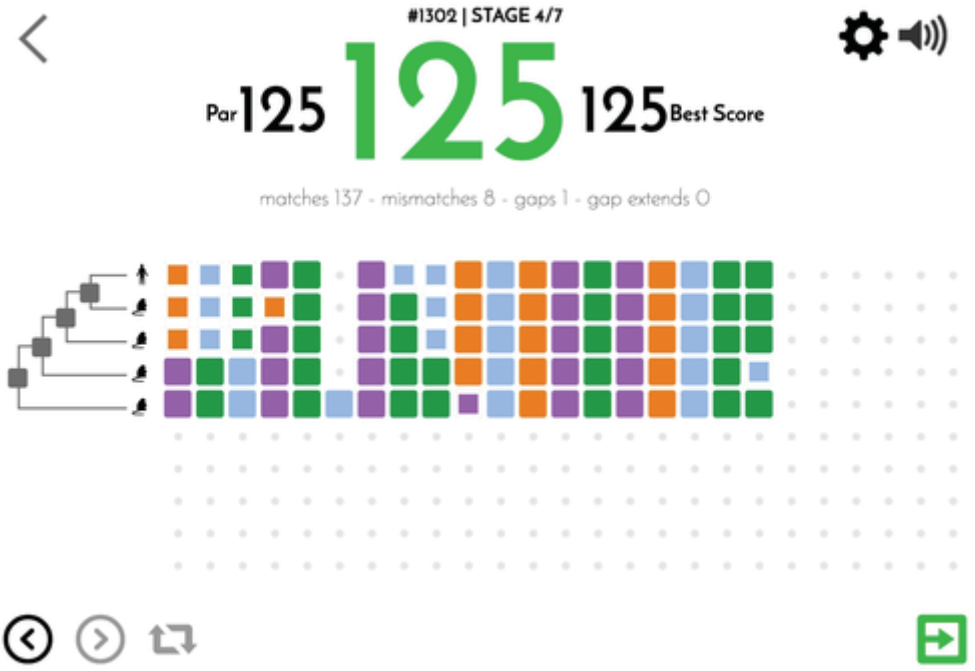

FIGURE 2.1: A screenshot of `Phylo`'s GUI

2.2.1 An open crowd sourcing approach

2.2.1.1 Improvements of `Open-Phylo` with respect to `Phylo`

`Open-Phylo` is the first crowd-computing system that is open for the benefit of the whole genomics community [1]. This version of `Phylo` enables any scientist in the world to benefit from crowdsourcing and human-computing technologies to help solve MSA puzzles. `Open-Phylo` is based on `Phylo`'s player interface (i.e., it keeps all the functionality and

[1] available at http://phylo.cs.mcgill.ca/expert/

advantages of `Phylo`), but features several key innovations that significantly broaden its player base (see figure 2.2 for further details).

`Open-Phylo` implements a *crowd manager*, which is a private interface that allows the user to manage and monitor his/her data. This manager allows users to identify and manage portions of the alignment on which the crowd-based improvements should be focused. Additionally, the user can manage the work of the crowd by removing or adding MSA puzzles from the pool that the user considers have (not) been played sufficiently often. The *crowd manager* allows tracking in real time the number of times each puzzle has been played and the magnitude of improvement to the alignment score achieved by the crowd.

`Open-Phylo` features a new expert gaming interface, which allows the most experienced users to play MSAs up to 300 nucleotides long. `Open-Phylo` can display sequences for up to 12 species with up to 300 nucleotides (the original version could only display up to 8 species with 24 nucleotides). This increase in sequence/features enables us to motivate the best players to use more efficient alignment skills that they have developed.

The `Open-Phylo` submission interface has several new key functions. First, users can select the objective function for identifying the best alignment. Users can select the function from a set of classical scoring functions (such as Ancestor, MUSCLE, T-Coffee) or they can also directly select the highest scoring alignment in the game. Secondly, a new graphical user interface (GUI) allows users to intuitively create casual puzzles by selecting a desired area of the MSA. Thirdly, the user can promote his/her research, initiate communications and knowledge transfer between the scientists and the player community.

2.2.1.2 Scoring Scheme

To evaluate a given alignment, `Open-Phylo` infers ancestral nucleotides or gaps at each ancestral node of the phylogenetic tree using a maximum parsimony approach called the Fitch algorithm[167][2]. The scores for induced pairwise alignments, each evaluated using

[2]The Fitch algorithm is run considering a gap as a fifth character, independently for each position.

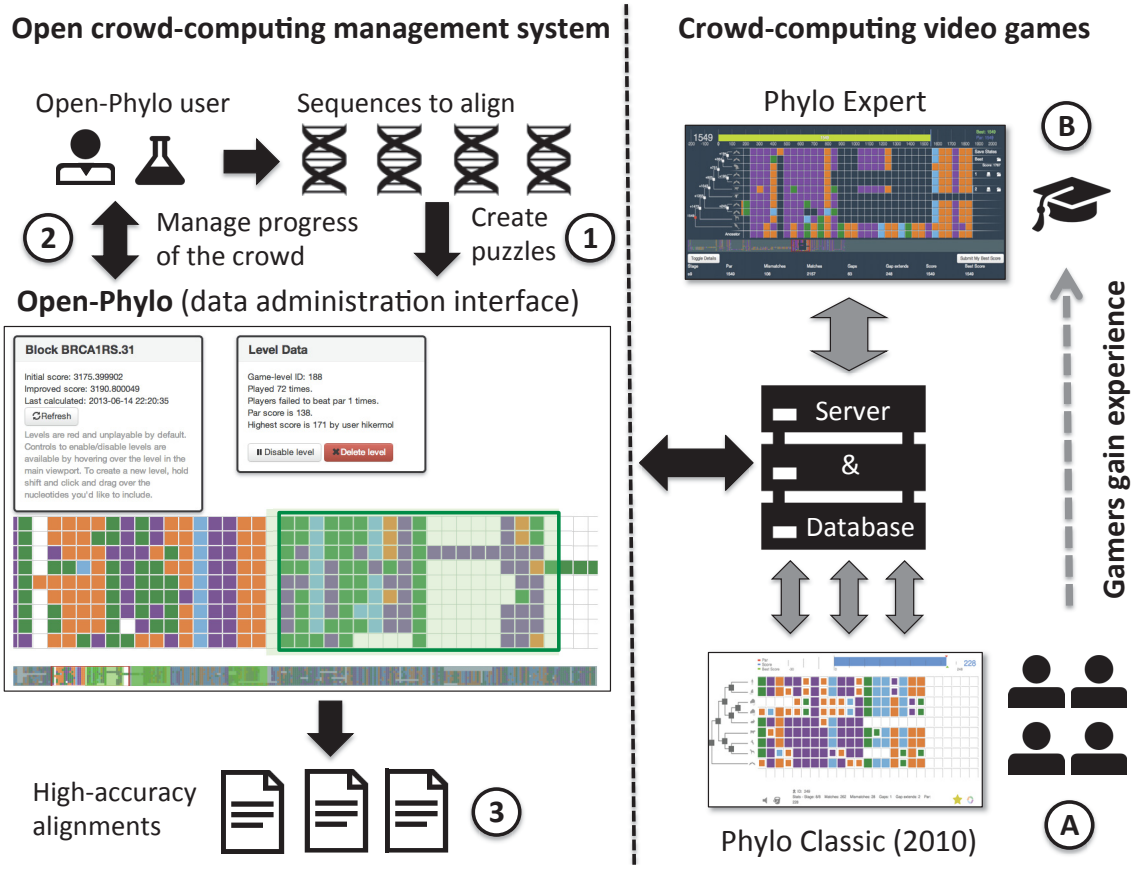

FIGURE 2.2: Open-Phylo crowd-computing system.
(1) Scientists upload their sequences to the database, validate the alignment puzzles built by the system (see green box in the data administration interface), or select new ones. (2) The same users monitor the progress of the crowd in improving their alignments, close, open puzzles, and finally (3) download the best solutions. The crowd-computing engine is powered by (A) many casual gamers playing classic puzzles and (B) a smaller number of experienced players, who have access to larger and more difficult puzzles. This figure and the caption have been extracted from [1]

an affine gap cost model[3], are summed over all edges of its associated phylogenetic tree. The original Fitch algorithm is not designated to accommodate an affine gap cost model and may result in sub-optimal ancestral sequences, which would yield a less than optimal alignment evaluation by `Open-Phylo`. To address this issue, `Open-Phylo` enables users to modify the ancestral sequences reconstructed with our variant of the Fitch algorithm. Therefore, players are able to identify and improve sub-optimal ancestors calculated by the game, and observe good MSAs that would be missed by the classical scoring scheme.

[3]This models statistically considers that the occurrence of d consecutive deletions/insertions is more likely than the occurrence of d isolated mutations. In other words, this model penalize the opening gap more than consecutive insertion/deletion events

The proposed scheme for scoring MSAs is evolutionarily realistic while being intuitive and fast to compute (as it is recomputed in real time every time the player modifies the alignment).

2.2.2 Results of a study case

To illustrate and evaluate the alignment capabilities of `Open-Phylo`, we used it to align sets of orthologous promoter sequences (regions of 1,000 bp located upstream of the transcription start site) of three key cancer genes from 12 different species of mammals. Each set of orthologous promoter sequences was initially aligned using one of four state of the art algorithms: Multiz [154], MUSCLE [54], PRANK [152] or T-Coffee [149]. The resulting MSAs ranged in size from 1,222 to 3,346 columns. For each initial MSA, we used `Open-Phylo`'s crowd-computing management system to direct the crowd efforts to a set of 79 (overlapping) expert-level puzzles of 300 alignment columns each. From the MSAs calculated by each of the four alignment programs, 1,014 casual-level puzzles (20 nucleotides long) were extracted and these were used as initial configurations for the levels of the casual game (also referred to as the classic game). Whereas solutions to expert-level puzzles can be directly evaluated using a given objective function, solutions to casual-level puzzles need to be reinserted into the larger alignment context before they can be scored.

Between 3 December 2012 and 3 April 2013, 12,961 unique visitors proposed solutions for 1,352 puzzles, including 338 expert-level and 1,014 casual-level puzzles. We assessed the extent to which the quality of an MSA could be improved through `Open-Phylo`. There is no single well-accepted scoring scheme for MSAs and each of the four aligners considered uses a different objective function. We thus evaluated each of the MSAs obtained using each of the following four scoring functions: Ancestor (a likelihood score reflecting both substitutions and indels on a given tree, which is approximated by the scoring function that `Open-Phylo` players try to optimize) [168], MUSCLE, GUIDANCE (a program that calculates the confidence score used by PRANK) [169] and T-Coffee. We evaluated the percentage of the 338 alignment blocks whose score was improved through `Open-Phylo`

(either in casual or expert mode), starting from the alignments produced by Multiz, MUSCLE, PRANK or T-Coffee, and using each of the four scoring functions (See figure 2.3). More precisely, we evaluated all solutions submitted by casual gamers and advanced players and kept only the best for each objective function. Our experiments revealed that, depending on the objective function used, `Open-Phylo` improved 32% to 97% of Multiz alignments, 16% to 93% of MUSCLE alignments, 24% to 90% of PRANK alignments and 43% to 99% of T-Coffee alignments. In practice, our data suggest that the top 40% of casual solutions, ranked using the game scoring function, are sufficient to reproduce our results (see subsection 2.2.2.2).

`Open-Phylo` appears to have the potential to improve a significant fraction of alignments calculated by any method for any scoring function. We obtained the largest improvements with the Ancestor and GUIDANCE scoring functions. Interestingly, these functions are precisely those that use the same user-defined phylogenic tree to score an alignment as the game. In both cases, and also for the MUSCLE objective function, we observed that for up to 62% of the cases, the solutions calculated from casual puzzles outperform those submitted by experts. This suggests that the work of many casual gamers can in some cases compensate for the lack of experts. Casual gamers are an important processing resource, who should not be neglected. However, this might not be the case for alignments calculated with T-Coffee, as the 44% improvement (using the T-Coffee objective function) was obtained almost exclusively from expert submissions. This discrepancy could be explained by the differences between the scoring scheme used in T-Coffee and the one used by our game. Nonetheless, since the latter achieved satisfactory performance with all other programs as well as with the T-Coffee objective function using the expert submission, we consider that the scoring scheme used in the game provides reasonable performance.

Overall, the magnitude of the improvement of the score is modest. For the classic version, the score improved by +1.9% (using the ANCESTOR objective function on MSAs calculated with Multiz), +28.4% (GUIDANCE/PRANK), +1.7% (MUSCLE) and 1.5% (T-Coffee). The expert version produced slightly larger improvements: +3.3% (ANCESTOR/Multiz), +10.9% (GUIDANCE/PRANK), +3.7% (MUSCLE) and 1.9% (T-Coffee).

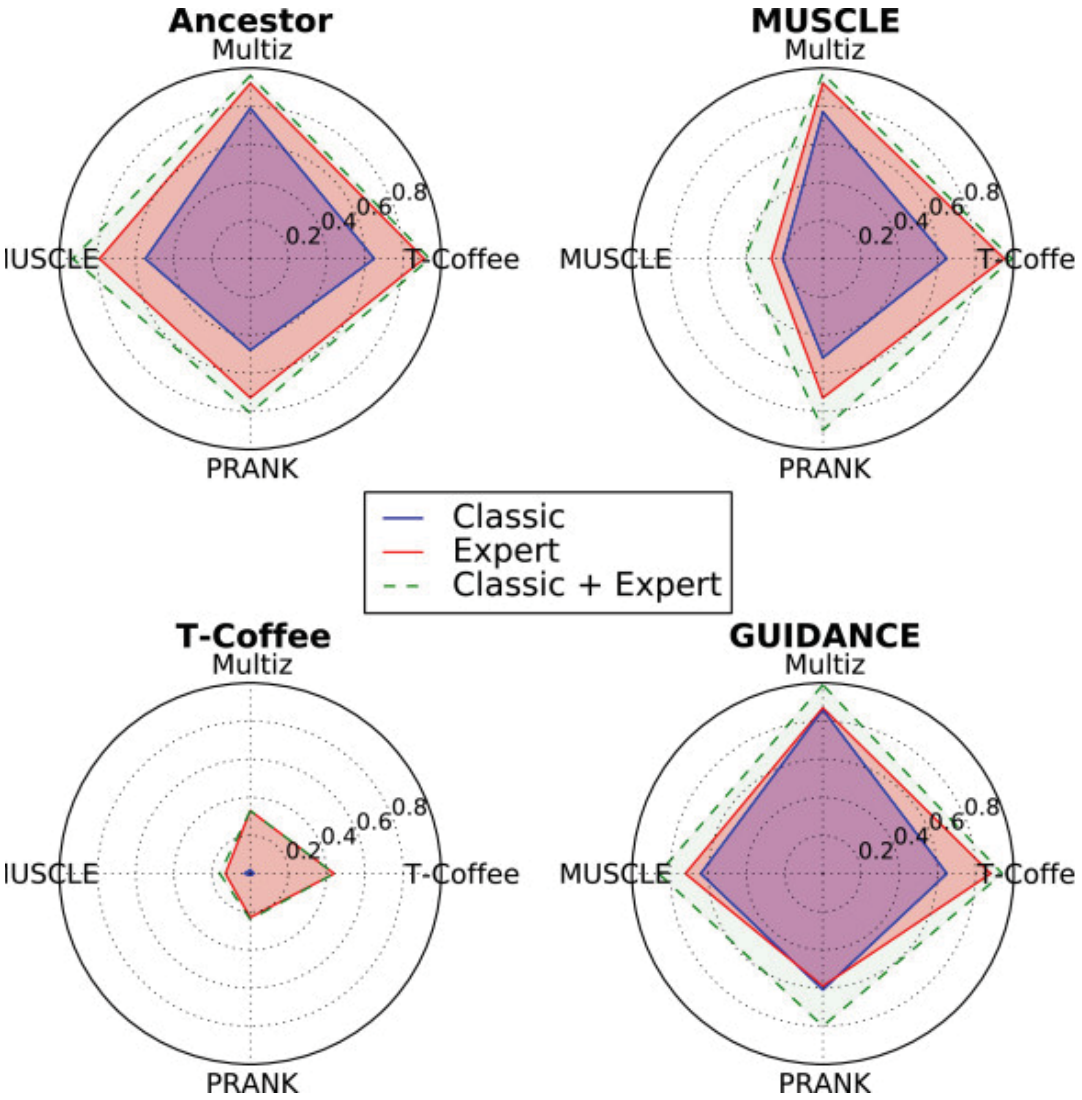

FIGURE 2.3: Performance of `Open-Phylo` using the casual or expert version of the Phylo video game.

The ratio of optimal solutions obtained with the casual version is shown in the area surrounded by a blue line, and the ratio obtained with the expert version in red. Each radar chart corresponds to one of the objective functions: Ancestor (top left), MUSCLE (top right), GUIDANCE (bottom right) and T-Coffee (bottom left). The alignment program used to calculate the initial MSAs is indicated on the axis of the radar charts: Multiz (north), MUSCLE (west), PRANK (south) and T-Coffee (east). This figure and the caption have been extracted from [1]

These values may appear low but are significant if we consider that the alignments calculated by the computers are already very well optimized. It also suggests that the players' solution remains in the vicinity of the initial MSA. Nonetheless, even if the magnitude of the improvement is not very large, some alignments may present significant qualitative improvements. An illustration of such a case is shown in figure 2.4. Here, an alignment of a portion of the promoter of the P53 gene originally produced by MUSCLE was improved by an expert player, resulting in an increase in the alignment score from 3,533 to 3,906, and clearly improving the conservation of several alignment columns.

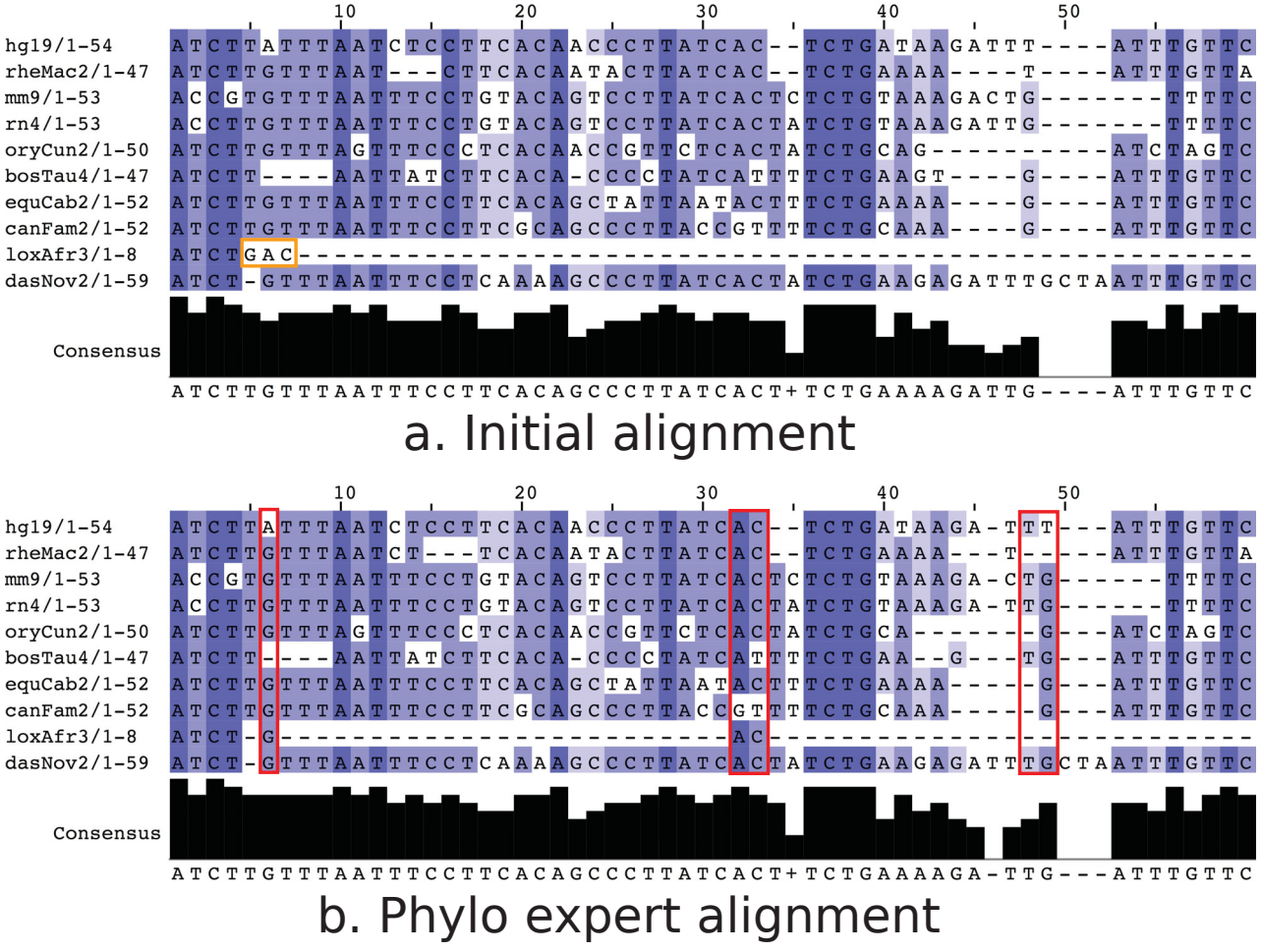

FIGURE 2.4: A multiple sequence alignment improved by `Open-Phylo`
(a) A section of the input alignment of the P53 gene calculated with MUSCLE. (b) The improved alignment obtained with the expert version of Phylo. Three nucleotides from the elephant sequence (loxAfr3) have been moved to increase the conservation of alignment columns 6, 32 and 33. The player also improved the alignment of columns 48 and 49 and revealed similarities not found in the original alignment. Image produced with Jalview [170]. This figure and the caption have been extracted from [1]

2.2.2.1 Comparison of classic vs expert games

To better understand the relative performances of the classic and expert versions, we compared the results of the casual (classic) game to the results from the advanced player (expert) version. In particular, we investigated whether the classic or the expert version of the game provided the best improvement. We show these data in figure 2.5. For each objective function (Ancestor [18], MUSCLE [15], GUIDANCE [20] and T-Coffee [17]) and each data set (initial MSAs computed with Multiz [14], MUSCLE [15], PRANK [16] or T-Coffee [17]), we determined which method (that is classic or expert) provided the highest improvement. The areas plotted in the radar charts correspond to the percentage of alignments improved by either the classic or expert version of `Open-Phylo`, which are also plotted in green in figure 2.3. When we normalized the data, we observed that 34% to 62% of the best solutions were produced by the classic version using GUIDANCE as a scoring function. At the other end of this spectrum, 76% to 96% of the best solutions were generated by the expert version with T-Coffee. Ancestor and MUSCLE provided intermediate results with, respectively, 26% to 40% and 20% to 43% of optimal solutions calculated with the classic version of the game. These data suggest that *i*) casual gamers might provide a processing power that should not be neglected and *ii*) the performance of the classic version depends on the objective function used by the `Open-Phylo` MSA submitter.

2.2.2.2 Improvement of MSA with casual levels

All solutions generated by gamers for casual puzzles with a score (using the scoring scheme of the game) higher than or equal to the score of the initial level are stored in our system. We have to find those that provide the best improvement (if any) from the initial levels. Since the scoring function used in the game is not identical to the objective function we wish to use to select the best alignment (for example, Ancestor, MUSCLE, GUIDANCE or T-Coffee), we inserted all of the proposed solutions into their original location in the full MSA and evaluated the global improvement using the desired objective function.

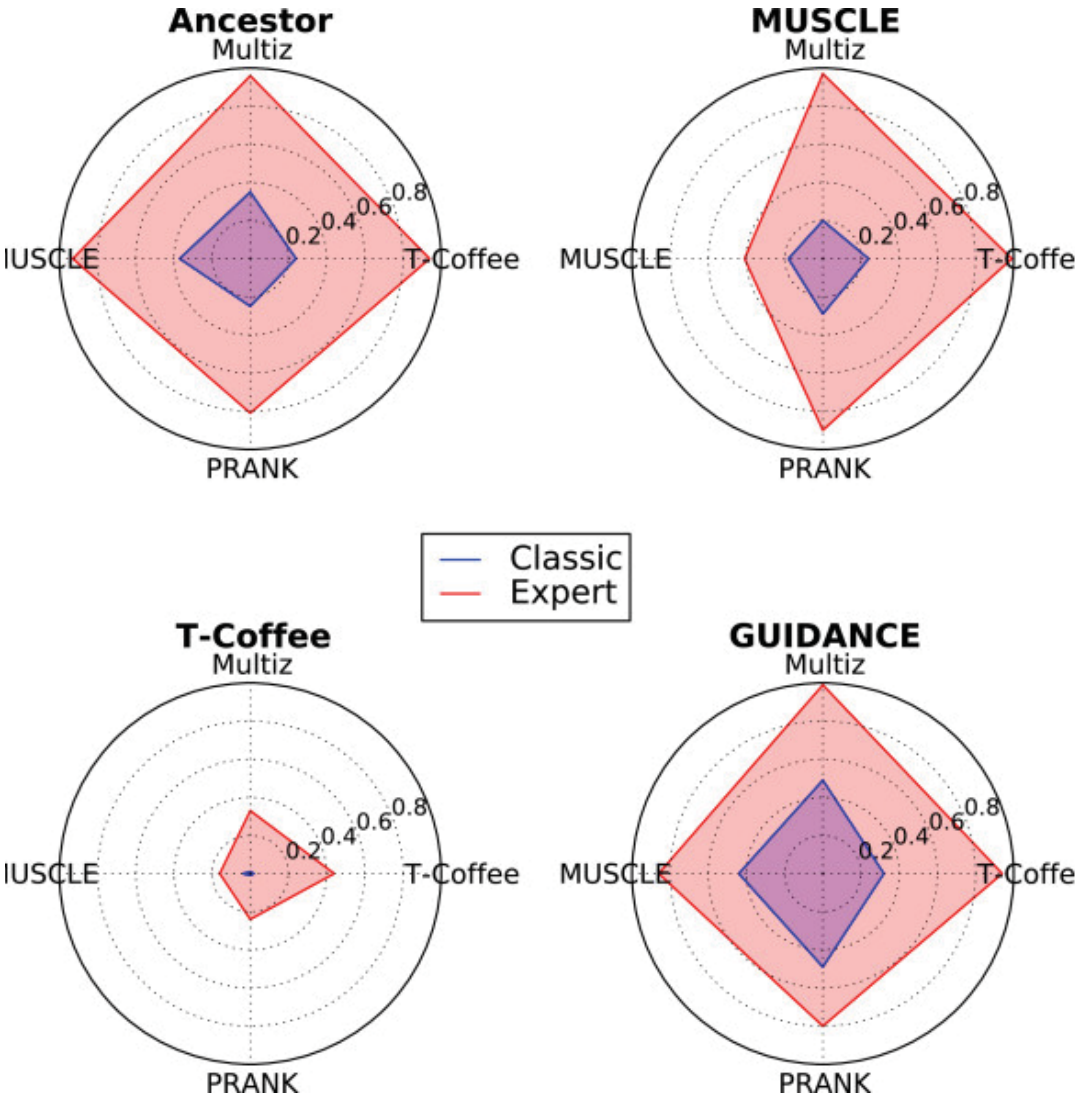

FIGURE 2.5: Comparison of the improvements provided by the casual and expert versions.

The ratio of optimal solutions obtained with the casual version is shown in the area surrounded by a blue line, and the ratio obtained with the expert version in red. Each radar chart corresponds to one of the objective functions: Ancestor (top left), MUSCLE (top right), GUIDANCE (bottom right) and T-Coffee (bottom left). The alignment program used to calculate the initial MSAs is indicated on the axis of the radar charts: Multiz (north), MUSCLE (west), PRANK (south) and T-Coffee (east). This figure and the caption have been extracted from [1]

The performance of the human-computing system was thus determined by the agreement between the scoring scheme used in the game and the values returned by the objective function used to identify the best alignments. To evaluate this correlation, we plotted (see figure 2.6) the distributions of the rank (based on the scoring scheme of the game) of the inserted solutions (for the submissions providing the best improvement). Our data reveal that 98% of the best alignments belong to the top 20 best ranked solutions (using the game scoring scheme). On average, we collected approximately 40 to 50 solutions for each casual puzzle. This suggests that instead of trying to insert all submissions, we need only consider the top 40% of solutions for improving the initial MSAs, while keeping the same performance and saving time and processing power.

2.2.3 Conclusions and Discussion

Our results suggest that humans can provide insights that cannot be entirely replicated by heuristics-based algorithms. This performance is most likely due to the capacity of humans to use their (visual) intuition to explore promising but abstruse configurations neglected by the heuristics implemented in alignment software.

Interestingly, we also observed that the scores of the best solutions from the four different initial alignments rarely converged to the same (or even similar) scoring alignments, suggesting that the players' solution remained in the vicinity of the initial MSA. Indeed, even if two different scoring functions agree on the global features of the 'best' MSAs, it is very unlikely that they will have the same global optima for all MSAs. Therefore, the performance of the system seems to be significantly influenced by the choice of the initial configuration, thus by the alignment program chosen by the submitter. Nonetheless, our results also suggest that `Open-Phylo` is able to improve alignments for the most popular objective functions.

`Open-Phylo` is the first open-science platform that enables any scientist in the world to benefit from crowdsourcing and human-computing technologies to help in solving one of the most fundamental and widely used problems in bioinformatics. We believe that

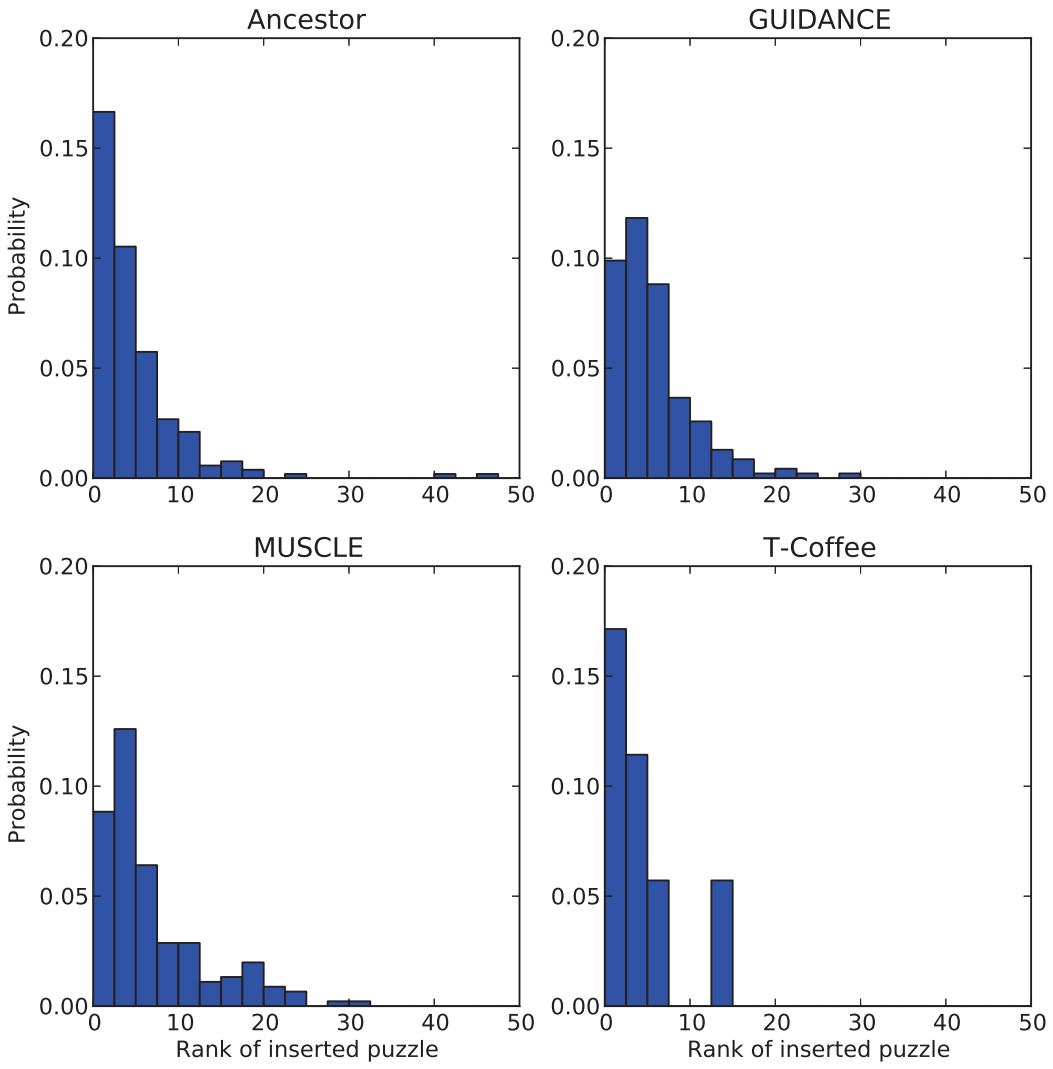

FIGURE 2.6: Performance of the game scoring function in identifying the best alignments.

The graphs show the distributions of the rank (calculated using the scoring function used in the game) of the best solutions found in the casual or classic game (that is where casual submissions were inserted into the initial MSA and found to have the best score). Each histogram corresponds to a different objective function: Ancestor (top left), GUIDANCE (top right), MUSCLE (bottom left) and T-Coffee (bottom right). This figure and the caption have been extracted from [1]

`Open-Phylo` is a pioneer for the next generation of crowdsourcing frameworks in biology: human-computing tools will be run by the people for the people.

2.3 Multiple Sequence Alignments improved by Algorithms

Despite the relatively long history and number of recent improvements, there is still a need for novel methodologies involving MSAs. Specifically, there is an increasing need for faster MSA tools to deal with big data and overcome the limited performance on remote homologs and low similarity sequences. MSA algorithms are bottlenecked by their dependence on particular sequence features. Multi-objective optimization (MOOP) approaches offer several characteristics that are desirable for MSA optimization. MOOPs operate on a set of MSA candidate solutions to provide approximate solutions for optimization problems, which may involve multiple conflicting objectives.

We propose a fast, scalable, and effective algorithm to optimize previously aligned sequences through a multi-objective approach. This algorithm is validated using a database of refined MSAs (BAliBase) and four standard metrics to evaluate the quality of predicted alignments.

2.3.1 Materials

The proposed experimental framework involves six state of the art algorithms to provide seed alignments for optimization, namely: Clustal W [150], Clustal Omega [171], Muscle [54], MAFFT [153], ProbCons [151] and TCoffee [172]. It is important to stress that the proposed algorithm can be scaled with the insertion or deletion of any other MSA approach.

BAliBase [173] is a database of refined MSAs commonly used to compare the performance of MSA programs. BAliBase performs validations on a wide spectrum of test cases in order to prevent the over-training of methods on a specific dataset. Each test case belongs

to one of five possible reference sets [173]. (RV11 and RV12) is a set of equidistant PDB sequences in which any two sequences share $< 20\%$ identity. (RV20) are orphan sequences[4] that all share $> 40\%$ identity. (RV30) are subfamilies where the sequences within a given subfamily share $< 40\%$ identity, but also where any two sequences from different subfamilies share $> 20\%$ identity. (RV40 and RV50) are sequences that share $> 20\%$ identity with at least one other sequence, but that include sequences containing large N/C-terminal extensions or internal insertions, respectively. BAliBase maintains a high quality of alignments through the use of 3D structural superpositions combined with a manual validation and refinement step. The BAliBase benchmark is composed by 6255 sequences and 218 alignments.

2.3.2 A Multi-Objective Evolutionary Approach (MOEA)

MOEAs belong to a class of stochastic optimization algorithms that mimic natural evolution to solve a MOOP. A MOOP problem can be defined as the problem of finding a vector $x = [x_1, x_2, \ldots, x_n]^T$ such that x:

i) satisfies the r equality constraints $h_i(x) = 0, \;\; 1 \leq i \leq r$,

ii) is subject to the s inequality constraints $g_i(x) \geq 0, \;\; 1 \leq i \leq s$

iii) optimizes the vector function $f(x) = [f_1(x), \ldots, f_m(x)]^T$.

The vector x is an n-dimensional decision vector and $\mathcal{X}$ is the decision space (i.e., the set of all expressible solutions). The objective vector $f(x)$ maps $\mathcal{X}$ into $\Re^m$, where $m \geq 2$ is the number of objectives. The image of $\mathcal{X}$ in the objective space, is the set of all attainable points.

The concept of optimality is introduced through the notion of Pareto optimality. A vector $x \in S$ is said to be Pareto optimal if all other vectors $x^* \in S$ have a higher value for at least one of the objective functions of $f(x)$. A Pareto front is the image of all solutions. The points that form the shape of the Pareto front are called non-dominated points. The

[4]sequences that are highly dissimilar to existing sequences with known structures

solutions in the final set are expected to be close to optimal and non-dominated with respect to one another.

MOEAs are algorithms that simulate natural evolution by an iterative computation process in which a set of candidate solutions are subsequently modified and improved through selection and variation procedures until some level of acceptable quality is met. The algorithm takes a given MSA (i.e., original population) and returns an improved alignment after performing a series of mutation and crossover operations. In other words, the procedure 'evolves' the original MSAs to produce a better one. Typically, a MOEA schema can be divided into the following phases: generation, evaluation, and selection of individual (see figure 2.7).

In a MOEA, individuals contain the information of solutions (i.e., alignments). Alignments are represented as a $n \times m$ multi-array of integers, where n is the number of sequences and m is the length of the alignment. If a cell in the multi-array corresponds to a 'match', then it contains the position of the AA in the protein sequence. On the other hand, if a cell corresponds to a 'gap', then it contains the position in the protein sequence of the previous 'match' (See the multi-array depicted in the boxes Crossover or Mutation in figure 2.7). This representation was chosen given its suitability in the implementation of genetic operators [158].

The proposed MOEA algorithm constructs an initial population (i.e., the set of individuals) using alignments produced by state of the art MSA algorithms. This population is composed of alignments returned by each algorithm, plus some genetically modified versions (i.e., versions that genetic operators have been applied to) of those alignments. Subsequent populations are then improved using a genetic algorithm. Genetic operators (crossover and mutation) then produce new generations through the selection of individuals based on a fitness function. Finally, a stopping criterion is achieved when for a fixed number of iterations no significant improvements have been made.

The mutation and crossover operators represent mechanisms that perform exploration (increasing the diversity of a population) and exploitation (increasing the depth of the search) procedures within the search space. An implemented mutation starts by randomly

selecting the position(s) of one gap (if available) and one amino-acid in an individual. It then introduces the gap after the amino-acid position and performs a shift of one position for all the positions between the selected amino-acid and the end of the sequence or the position of the selected gap, whichever occurs first (see figure 2.7). A two-point crossover, where each point represents a column in the alignment, is then implemented. Once two points have been randomly selected in one individual (parent one), the positions between the two points, which correspond to the same string of position indexes, are then sought in another random individual (parent two). Finally, the strings of positions are exchanged between the two parents (see figure 2.7).

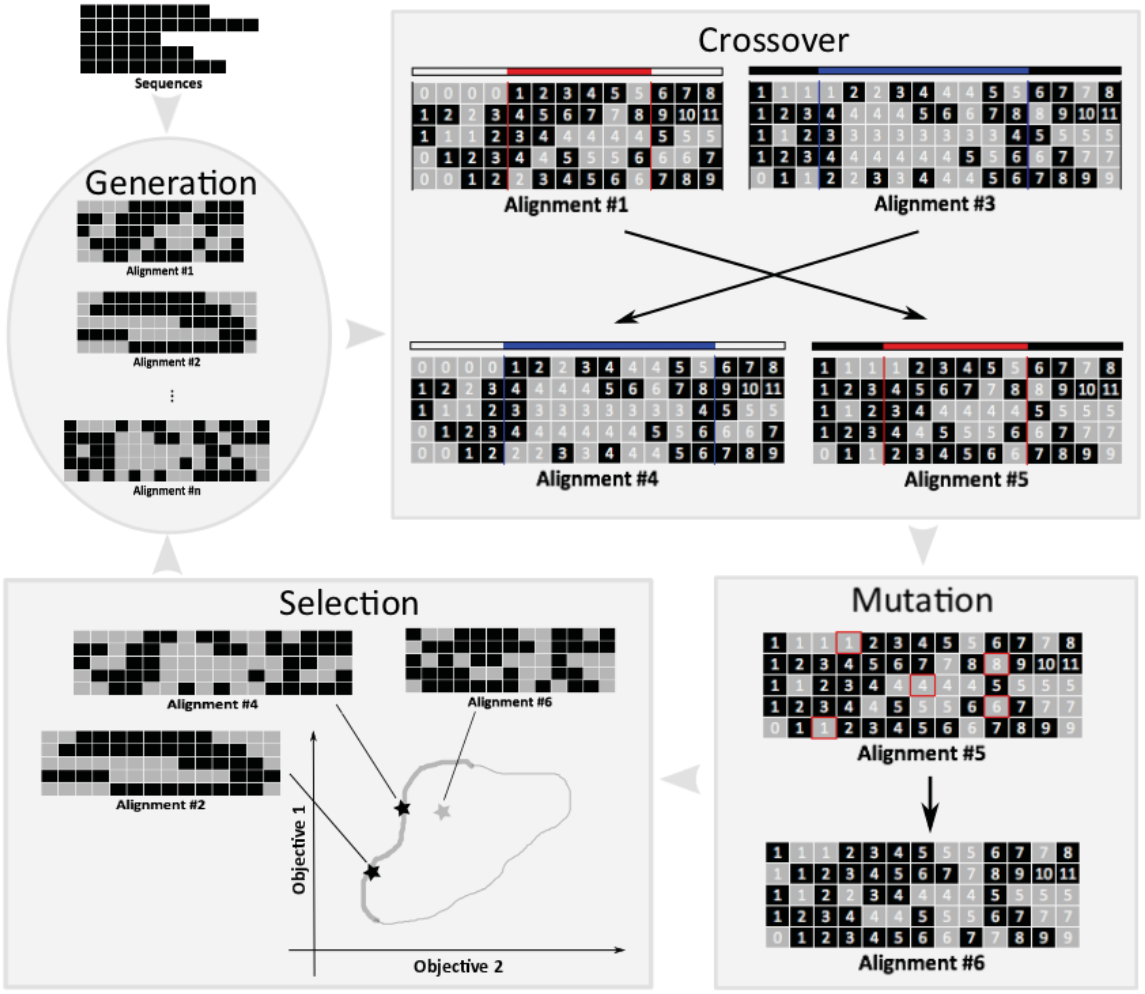

FIGURE 2.7: The proposed MOEA algorithm for improving MSAs

2.3.2.1 Evaluating Individuals

The proposed MOEA evaluates individuals through a fitness function that considers two objective functions: *i*) entropy and *ii*)the metric MetAl [174]. Entropy measures the variability of an MSA by defining the frequencies of occurrence for each letter in a given column. Entropy is minimal when only one symbol is present in a given column and maximal when all symbols are present with the same frequency. The total entropy for a MSA is the sum of entropies for each column (see equation 2.1), where P_i is the fraction of residual AA type i, M is the number of AA types plus the gap character, and N is the number of columns in the MSA. For the purposes of this algorithm, good alignments are considered to be those that minimize the total entropy.

$$H = \sum_{j=1}^{N} - \sum_{i=1}^{M} P_i \log_2 P_i \tag{2.1}$$

Frequency-based approaches, as a measure of entropy, do not consider the sequential and evolutionary characteristics of a residue presented in a MSA. The approach proposed here uses a combination of four metrics that incorporate position and evolutionary information, which are processed by MetAl software to compute a single score between a target (the offspring) and a set of sequences (the parents). This score is based on four factors: *i*) a simple correction to the Sum of Pairs (SP) score; *ii*) raw gap information; *iii*) positions of gaps occurring within a sequence; and *iv*) positions of indel events occurring within a sequence and phylogenetic tree. This single score is then used as a second objective and minimized.

2.3.2.2 Selecting Individuals

The selection process drives the algorithm to search regions containing the best individuals relative to the Pareto front. In order to generate the Pareto front for each generation, each population is sorted according to a non-dominated approach. A population is sorted into different non-domination levels, and the Pareto front is filled with individuals belonging to the best ranked levels until the desired number of individuals for a population is reached.

Since MOEAs require a high-level information phase to report one solution from the Pareto front, an algorithm based on the identification of knees (i.e. the regions in the Pareto front where small displacements produce a big detriment to at least one of the objectives) was used [175].

2.3.2.3 Validating Individuals

We validate our predictions using the Sum of Pairs (SP), Total Column (TC), MetAl and hypervolumen metrics based on the BAliBase benchmark.

The SP and TC metrics (equations 2.2 and 2.3) were computed by the *BaliScore* script, where M_r is the number of columns in the reference alignment, S_{ri} is the score for the i-th column in the reference alignment and $p_{i,j,k}$ is equal to 1 if the pair of residues $A_{i,j}$ and $A_{i,k}$ are aligned with each other in the reference alignment, otherwise 0. C_i will be equal to 1 if all the residues in the i-th column are aligned in the reference alignment. The possible values for SP and TC range from [0,1], and a score equal to one represents an exact agreement between the alignments.

$$SP = \frac{\sum_{i=1}^{M} S_i}{\sum_{i=1}^{M_r} S_{ri}}, \quad where \quad S_i = \sum_{j=1}^{N} \sum_{k=1,k\neq j}^{N} p_{i,j,k} \tag{2.2}$$

$$TC = \sum_{i=1}^{M} \frac{C_i}{M} \tag{2.3}$$

SP and TC are standard scores for comparing the performance of MSAs. However, many concerns have been raised about their use. The MetAl score [174] can be used as a metric to fix inaccuracies and to incorporate additional information in the evaluation process. The MetAl metric incorporates a correction to the SP score as well as gap and indel information when computing the score between a target and a set of sequences. The range for the MetAl score is [0,1], where a perfect match is found when the score is equal to 0 (plotted as $1 - MetAl$ in figure 2.8). It is important to note that the MetAl score uses the BAliBase reference alignments as its target in the validation process, in contrast

to the evaluation process, where the parents are used as targets. Therefore, the BAliBase alignments are strictly used only during the validation process to guarantee a blind and fair comparison between the MSA tools.

The hypervolume is a metric used by researchers to measure the quality of a Pareto front. Hypervolume measures the volume of the dominated portion of the objective space. This indicator represents the spread of solutions along the Pareto front, as well as the distance for a given set of solutions from the Pareto-optimal front. The hypervolume has two highly desirable features: it is sensitive to any improvements, and guarantees that any approximation set that achieves the maximally possible quality value for a particular problem contains all Pareto-optimal objective vectors [176]. The hypervolume indicator, I_H, for a solution set $A \subset \Re^d$ can be defined on the basis of a reference set $R \subset \Re^2$, as shown in Equation 2.4. In that equation, λ stands for the Lebesgue measure and the set $H(A, R)$ denotes a set of objective vectors that are enclosed by the front $F(A)$ given by A and the reference point R.

$$I_H(A) := \lambda(H(A, R)) \tag{2.4}$$

2.3.3 Results

The MOEA algorithm was executed based on the following parameters: seven independent runs on 218 alignments, with a fixed population size of 56 individuals. The crossover and mutation probabilities were set to 0.3 and 0.1, respectively. A stopping criterion was considered to be achieved when five consecutive generations showed no improvements in hypervolume.

The input of our algorithm is a set of pre-aligned sequences and the output is the prediction of a single alignment belonging to the non-dominated set. The MOEA's output is compared with the alignments predicted by six state of the art algorithms, having the BAliBase benchmark as target and four validation scores.

Figure 2.8 shows the proportion of alignments with the best scores reported for each MSA tool. The proposed algorithm demonstrates excellent performance for all six groups. It performed outstandingly for groups RV11 and RV12, where the proposed algorithm outperformes other MSA tools for all four validation scores. The excellent performance of the proposed algorithm is more easily observed in the MetAl and TC scores than the SP score. For example, 60% and 48% of the best MeTAl and TC scores are found in the RV12 and RV11 sets, respectively. The proposed approach ranked second best (or better) for all groups based on the MeTAl and TC scores. For the TC score, our approach achieved optimum results in 5 out of 6 groups, and second best in the remaining group (RV50). For the MeTAl score, the MOEA achieved optimum results in 3 out of 6 groups (RV11, RV12, RV30), and second best in the three remaining groups (RV20, RV40, RV50). With respect to the SP score, the proposed approach is in the top 2 for 4 out of 6 groups. The proposed approach predicted the best alignments for two groups (RV11, RV12), second best in two groups (RV30, RV50), and third best in the remaining groups (RV20, RV40).

Figure 2.9 reports the average error obtained by each MSA tool. This error is defined by equation 2.5, where $S \in \{MetAl, SP, TC\}$. The functions $score(i)$ and $best(i)$ return the score and the best score of alignment i generated by metric S on all the MSA tools. From figure 2.9, we can observe that the proposed approach works very consistently with all groups for each of the three score schemas. Its degree of error is especially low in RV11 and RV12, where the MOEA obtained for each of the three score schemes the best error compared to the other MSA tools. These results concur with an analysis of the same sets in figure 2.8. The ClustalOW algorithm behaved similarly for groups RV30 and RV40. However, ClustalOW had a higher number of errors for RV11 and RV12 sets.

$$error_S = \sum_{i=1}^{M} |best_S(i) - score(i)| \tag{2.5}$$

Figure 2.10 reports the quartiles of the gained hypervolume through the computed generations of the MOEA. The gain of a specific alignment is computed as the difference between the hypervolume of the last and first Pareto fronts. In figure 2.10, it is clear that the volume of dominated objective space increased through the generations (i.e, values

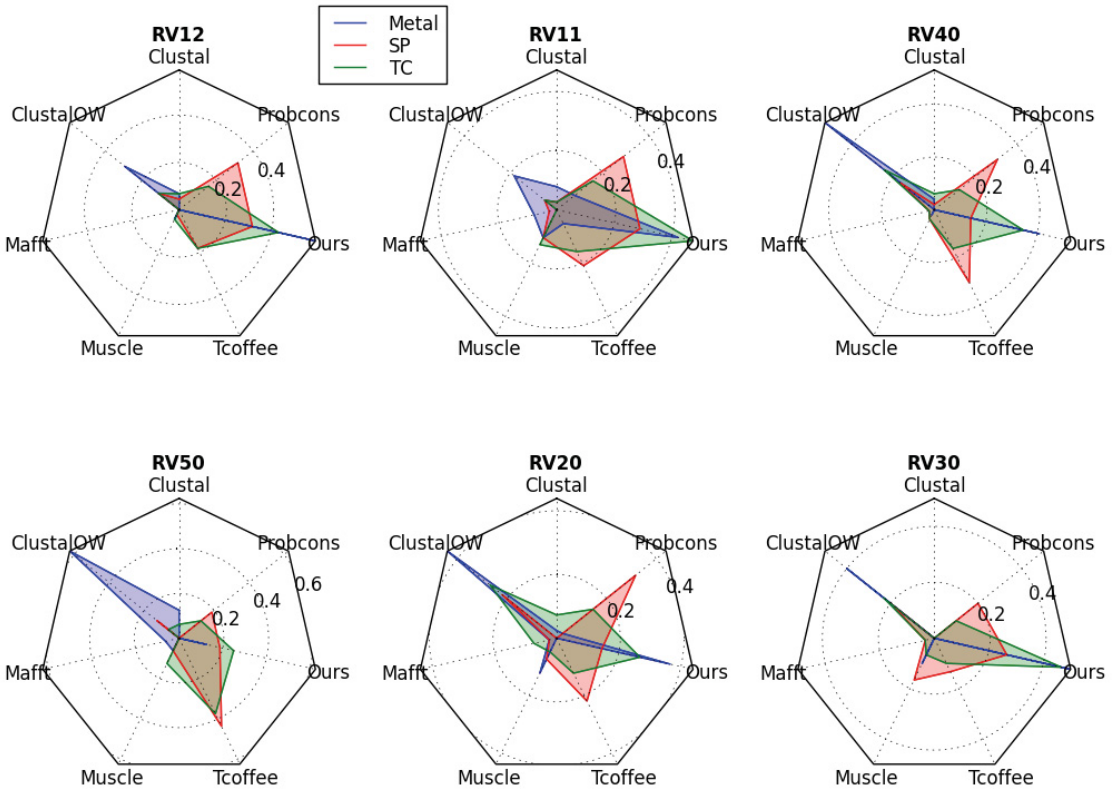

FIGURE 2.8: Proportion of alignments with the best scores reported for each MSA tool.
The results are clustered in 6 different hierarchical groups as defined by BAliBase; each vertex in the polygon represents an MSA tool. An area is plotted with regards to the proportion of best alignments reported by the SP, TC and (1-MetAl) scores.

greater than zero). The proposed algorithm was able to then push a set of already aligned sequences (the initial population) towards the Pareto optimal solutions (increasing the hypervolume in 204 (94%) alignments). The worst performance is found in the RV11set, where seven alignments did not improve their hypervolumes. On the other hand, the RV50 set improved the hypervolume of all its alignments.

2.3.4 Conclusions and Discussion

This work contributes to the MSA problem by proposing a novel algorithm to optimize previously aligned sequences. The proposed model is based on a MOEA, which, as this work demonstrated, provides an adequate exploration of the search space. The proposed

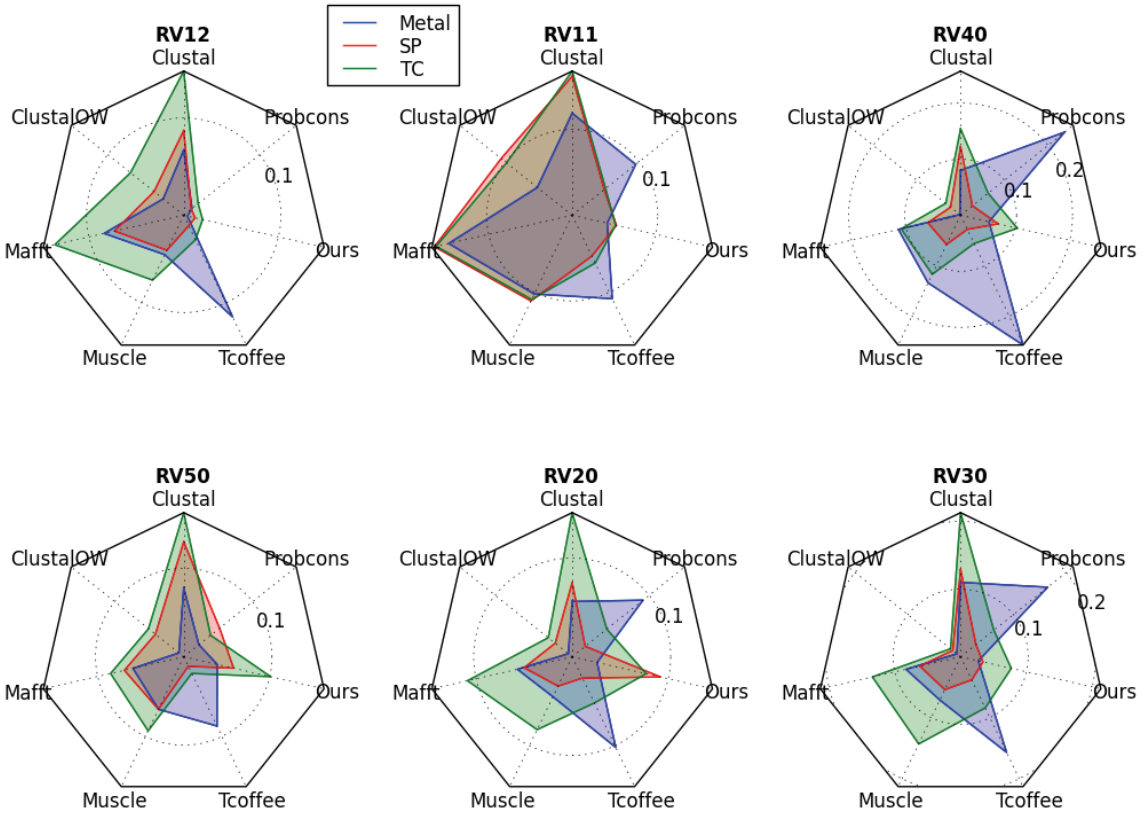

FIGURE 2.9: Average error obtained for each MSA tool.
The results are clustered in 6 different hierarchical groups as defined by BAliBase. Each vertex in the polygon represents a MSA tool and an area is plotted with regards to the average error (see equation 2.5) obtained by MSA tools.

strategy improves the accuracy of the MSAs used as inputs for the model. Be aware the proposed algorithm is not the holy grail of MSA tools, and is not considered to outperform all the other MSA tools on any possible sequence. However, it is a method that, with a very reasonable cost in CPU time, produces more accurate alignments from seed MSAs. The proposed algorithm proved to be less dependent on specific features of sequences and very robust when used on diverse biologically targeted sequences.

The proposed approach is more than just a static algorithm. It is also a pipeline that allows for the optimization of different MSA tools. In this work, six different MSA tools were chosen to develop a study case, however, these can be replaced with any other. This feature is important because it allows our algorithm to be tested on a diverse range of representative situations and will accommodate new MSA tools as they become available.

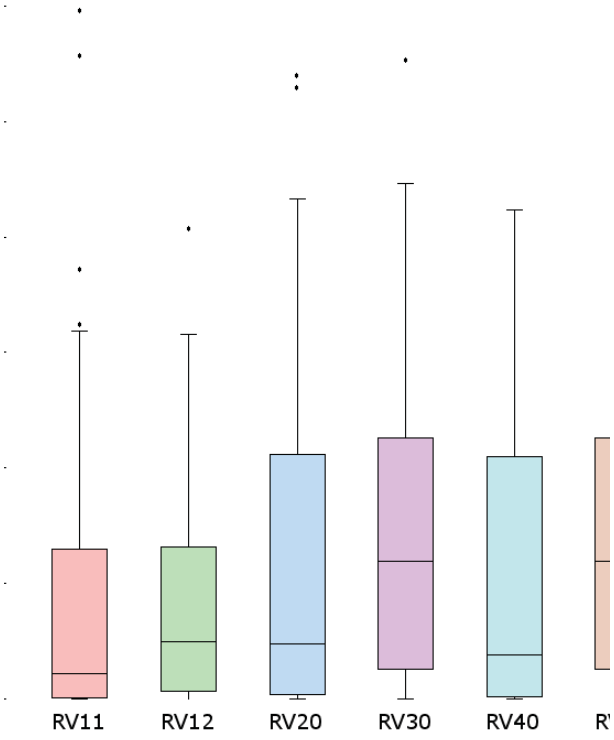

FIGURE 2.10: Hypervolume boxplots
gained on progression of the MOEA algorithm.

Chapter 3

Modeling Protein Folding Pathways and Structure

3.1 Abstract

The PF problem aims to predict the complete physical and dynamical process that transforms an unfolded protein sequence into a functional 3D structure. To date, classical approaches to obtain this information rely on time-consuming high-resolution molecular dynamics (MD) simulations or fragment assembly methods that are primarily limited to relatively small molecules ($\leq$ 50 AA). In this chapter, an alternate, yet complementary, strategy that offers a better trade-off between resolution (i.e., modeling secondary structures) and efficiency (i.e., speed and scalability) is defined through the use of `efold`. `efold` is a novel divide-and-conquer algorithmic framework that combines ensemble modeling techniques with evolutionary-based sequence information to compute accurate coarse-grained representations of the conformational landscape for large proteins. This landscape is then used to predict folding dynamics and dominant folding pathways.

3.2 Sequence Information

Recent advances in estimating the 3D structure of topologically complex proteins have been achieved by the combination of residue level tools with high-resolution computational methods. This hybrid methodology has allowed for the generation of models with great accuracy at high-resolution on larger proteins structures that was not previously reachable [113, 177, 178]. The general idea behind this approach is to increase the scalability capacity of methods by using protein sequence information as constraints to massively reduce the search space of possible protein conformations sampled by high-resolution methods. Experimental and bioinformatics methods for residue contact and secondary structure determination have been shown instrumental as an intermediate step toward accurately predicting of the 3D structure [179]. For the first time, `efold` allows for the integration of protein sequence information into the Boltzmann sampling process that is performed by ensemble methods to predict protein pathways. When possible, `efold` takes advantage of protein sequence information to use evolutionary information during the modeling of folding pathways. Secondary structure and co-evolutionary predictions are incorporated into a statistical residue contact potential function to improve the representation of key interactions in the protein folding prediction. `efold` is able to handle a variety of diverse sources that contain evolutionary information. In the proposed experimental framework (see Chapter 4), the predictions of `efold` were computed using secondary structure assignments (derived by the `DSSP` software), secondary structure predictions (derived by the `PSIPRED` software), residue contacts (derived by the `EVfold` software) and $\beta - \beta$ residue contacts (derived by the `bbcontacts` software). `efold` was shown to be able to extend those prediction sources to more diverse alternatives.

3.2.1 Secondary Structure Information

Given the difficulty of PSP and PPP, many algorithms have been developed for an easier task: predicting secondary structure (SS). The SS of a protein can be described as the local conformation of the polypeptide backbone. This local conformation plays a crucial role in the formation of a protein's native structure and provides invaluable information

about a protein. The prediction of a protein's SS is a core step in most state of the art PSP methods, where a reliable SS prediction is commonly used as an intermediate for attempting the far more difficult task of predicting its complete 3D conformation.

There are two main backbone configurations to a SS, either α-helix or β-sheet. An α-helix is when hydrogen bonds form between residues that are close to each other in the protein's sequence. In contrast, β-sheets are formed by hydrogen bonds between two parts of the polypeptide chain that have close proximity in 3D space, but may be sequentially far away from each other. Predicting a protein's SS usually involves classifying its AA as either helices(H), sheets (E[1]), or coils (C).

Since the 1970's, four main generations/stages in SS prediction methods have been developed. It is important to stress that each successive generation/stage includes the features of a previous generation/stage. SS methods that belong to the first generation compute the propensities of finding a specific AA in each backbone configuration (i.e., α helix or β-sheet). These propensities are computed using proteins reported in structural data bases. Second generation methods are based on the propensity for segments (typically 11-21 adjacent residues), as opposed to isolated AAs in the first generation. In third generation methods, information from homologous sequences and machine learning approaches were used. The assumption behind a SS prediction by homology is that if a sequence of unknown SS has a homologue of a known structure, we can perform an accurate prediction if we 'copy' the known secondary structure over to the unknown sequence. In the fourth generation, a combination of secondary and tertiary protein structure is used. Information about the 3D protein conformation is added to SS predictive methods. In spite of the progress achieved by SS prediction approaches, they have plateaued with an average prediction accuracy around 80% per residue in unknown protein sequences using fourth generation approaches. Some authors believe [180, 181] that substantial improvement on the accuracy of SS prediction methods can only be possible if better representations of SS features are known.

[1]The character 'E' is used by the Dictionary of Protein Secondary Structure (DSSP) to describe extended strand in parallel and/or anti-parallel β-sheet conformations

Regarding SS information, two different state of the art algorithms (i.e., `DSSP` and `PSIPRED`) were included in the algorithm suite to provide sequence information to `efold`. `DSSP` standardizes the secondary structure assignment of a protein given its atomic-resolution coordinates [182]. The `DSSP` algorithm calculates the most likely SS assignment given the 3D structure of a protein. This algorithm reads the position of atoms in a protein (i.e., 'ATOM' records in a PDB file) to calculate the potential H-bond energy between all atoms. Next, the two best H-bonds for each atom are then used to determine the most likely class of SS for each residue in the protein. In comparison, `PSIPRED` is an accurate SS prediction algorithm that incorporates two feed-forward artificial neural networks (ANN) to perform the SS prediction based on output obtained from PSI-BLAST (i.e., Position Specific Iterated - BLAST) [146]. PSIBLAST retrieves sequences related to the target protein (i.e., the protein whose SS is unknown) to build a position-specific scoring matrix. This matrix is then processed by a learned ANN to predict the SS of the input sequence. It is important to note that `PSIPRED` predicts SS, meanwhile `DSSP` assigns SS based on known 3D structures.

3.2.2 Residue-contact Information

Predicting residue contacts by co-evolution-based strategies received a new twist thanks to new methodological advances and increasing availability of protein sequences. Breakthroughs in handling phylogenetic information and disentangling indirect relationships of AAs have resulted in an improved capacity to correctly predict inter-residue contacts [108, 109]. It is still unclear what accuracy, coverage, and distribution of contacts along the sequence are needed to be useful in practice for PF methods. However, predicted inter-residue contacts have already been used to increase the scalability capacities in different PF approaches [110]. For example, homology and fold recognition methods (see section 1.2.1) reduce the conformational space by filtering out the least likely structural models based on predicted residue contacts. Regarding *ab initio* methods, residue couplings have been translated to a set of distance constrains for effective use in distance geometry generation of 3D structures and their refinement by energy minimization and molecular dynamics methods [111, 112]. This new generation of contact prediction tools

is enticing, but it is currently limited by the amount of rich evolutionary sequence data required to obtain reliable structural models. For accurate predictions, these methods require large numbers of homologous sequences using accurate alignments that are sufficiently diverse to reveal co-evolution patterns that cover most structural elements of the protein [109]. This fundamental area of research focuses not only on predicting contacts with high accuracy to be used in structure modeling, but on building a reliable structure from incomplete/inaccurate contact data [113].

`EVfold` and `bbcontacts` were included in the algorithm suite to provide contact residue information as input to `efold`. `EVfold` uses evolutionary variation (coded in MSAs) to calculate a set of co-evolved residue pairs in a protein family using information theory algorithms (e.g., maximum entropy method and direct coupling analysis) [109]. `efold` uses the ranked set of evolutionary inferred contacts (EICs) predicted by `EVfold` to represent a network of interactions across the protein and reflect the co-evolution of pairs during the evolutionary trajectory of the protein. It is important to stress that the predictions of `EVfold` were constrained to not include SS predictions. `bbcontacts` uses matrices of predicted co-evolved residues to estimate interactions between SS elements [183]. This algorithm predicts $\beta-\beta$ contacts by detecting structural regularities of paired $\beta-$strands in the 2D maps of coupling scores using two Hidden Markov Models (HMMs), one for parallel and another for antiparallel contacts.

3.3 Ensemble Modeling

Classical approaches for PSP typically search for a singular, lowest energy structure. However, there is compelling evidence that indicates proteins have the ability to adopt different conformational states *in vivo*, where multiple substrate minima could exist with different functional properties [115]. The folded state can then be understood as a small ensemble of conformational structures compared to the conformational entropy present in the unfolded ensemble. Additionally, Anfinsen's hypothesis implies that, because the native state of a protein is an ensemble of many similar conformations, the goal of PSP methods should be to predict this ensemble rather than just a single conformation. PSP

prediction methods often assume that the ensemble of conformations is very narrowly distributed around the mean and then the representative conformation (i.e., the predicted native state) would approximate the mean of the ensemble. However, theoretical analyses of the folding landscape suggest that there are a large multiplicity of low-energy, partially-folded conformations near the native state [184]. Therefore, it is more appropriate to represent a protein structure as an ensemble of alternative conformations [185].

Regarding the PPP problem, a large collection of folding routes traversing a multidimensional energy landscape within a folding funnel model has replaced the idea of a single folding pathway. Intermediate conformations are not considered as discrete states, but as ensembles of structures, where the transition between consecutive ensembles on the folding pathway occur in parallel routes [186]. There is strong experimental evidence supporting the heterogeneity of folding pathways [187]. For example, it has been shown that with structurally homologous kunitz-type protease inhibitors, there is a heterogeneity of folding intermediates and folding kinetics [134]. Multiple pathways and ensembles of structures can be obtained by molecular dynamics or other local deformations methods, but these time-consuming, high-resolution methods are primarily limited to relatively small molecules. Given that these methods can only look at local variations, multiple simulations would have to be performed to construct an accurate set of ensembles.

Free energy minimization is the most popular method for determining the lowest energy structure from a single protein sequence. Although energy models have undergone refinements resulting in more accurate characterization of folding thermodynamics, there is still uncertainty in the experimental estimates of these parameters. The energy value computed for a structure is an approximation as well, as the absence of quantum mechanics computations in the determination of the energy equation do not allow for an adequate representation of the forces responsible for protein folding. As a result, finding an intermediate state with lower energy than the *minimum free* energy state is not an uncommon problem in free energy minimization based methods.

Newer computational modeling approaches, called ensemble methods, are no longer performing a search for an individual lowest energy structure, but rather aim to predict an ensemble of protein conformations and pathways to describe a more realistic landscape

of conformational variants without sacrificing efficiency or accuracy. Furthermore, these methods bring a conceptual breakthrough when compared to previous PF methods because they are able to address the complexity barrier of simulations by computing coarse grained representations of complete energy landscapes at a large scale. Based on ensemble methods, the protein energy landscape is characterized using a statistical mechanics perspective. According to statistical mechanics theory, molecular state is in constant flux when at equilibrium, but the proportion of molecules in each specific state remains constant, allowing one to quantify the architecture of a system. The reader can refer to section 1.4.2 to review current state of the art of ensemble methods. In the remaining sections of this chapter, we will cover the methodological protocol followed by the proposed protein folding predictor.

3.4 Algorithm Design

In the following sections we will present `efold`, a novel divide-and-conquer algorithmic framework that combines ensemble modeling techniques with evolutionary based sequence information to compute accurate coarse-grained representations of the conformational landscape for large proteins. `efold` enables efficient simultaneous prediction of a protein's folding mechanism(s) and structure using only the primary sequence as input and, when available, evolutionary sequence information. `efold` addresses the following two main tasks (see figure 3.1 for a schematic pipeline of `efold`'s methodology):

1. **Modeling ensembles:** The main goal of this task is to compute a set of protein states with the highest likelihood. Our approach is two-step:

   (a) The **forward step** of the algorithm computes the equilibrium partition function of all possible β SS using a divide-and-conquer approach and memoization techniques. We compute the Boltzmann partition function over the set of all possible protein states, where the protein states have been modeled through a coarse-grained representation based on SS. Each protein is assumed to fold into a complete set of unique structural states with a single energetic value assigned

(Boltzmann distribution) and evolutionary contact prediction scores if available. Clusters of low-energy states with similar conformations are produced based on a structural similarity metric.

(b) The **backward step** computes the set of statistically representative samples.

2. **Modeling Folding Pathways:** The main goal of this task is to derive the likelihood of dynamic state-to-state transitions and assemble a set of complete folding paths. The transition from a random coil to the native state is modeled as a path in a graph of varyingly folded protein conformation states. The system dynamics are calculated by treating the folding process as a continuous time discrete state Markov process. Finally, the folding pathways are predicted by combining the predicted dynamics with the predicted conformational landscape.

With `efold`, it is conventional to consider that each protein sequence folds into a complete set of structural states. The first step in developing an ensemble approach is the choice of an appropriate model for representing the protein (section 3.5.1). The second step is to calculate the complete set of all possible structural states (section 3.5.2). This state set represents a description of a more realistic landscape of all conformational variants. For each conformational state, `efold` performs a third step that computes a single energetic value according to the entire conformation space (see section 3.5.3). Next, the Boltzmann partition function (BPF) is computed from the set of all possible protein structures to estimate the significance of all conformations and their likelihood of occurrence (see section 3.5.4). The partition function is a thermodynamic normalization constant that encodes the statistical variation of a system in equilibrium, and can be used to identify significant structures within an ensemble. We assume that the minimum ensemble in the energy landscape contains the native state and must have a partition function that is larger than any other ensemble of structurally similar conformations. The complete enumeration and examination of suboptimal folding is an arduous task. The proposed ensemble model generates a statistically representative sample from the computed landscape to perform a statistical characterization of the Boltzmann ensemble (see section 3.5.5). Since thousands of states are sampled at any given time, a tractably sized system is desired. The proposed method partitions the state space into macro states

by clustering (see section 4.2.4.4). The clustering procedure allows for the largest set of structurally related low-energy conformation to be searched, instead of focusing only on the lowest energy conformation. Next, `efold` represents clusters as nodes in a graph of varyingly folded protein conformation states. Coarse folding transitions are performed by modeling the transition between two clusters, which is represented as an edge in that graph (see section 3.5.7). The main goal of this task is to derive the likelihood of dynamic state-to-state transitions. The dynamics of the system are calculated by treating the folding process as a continuous time discrete state Markov process (see section 3.6). Finally, `efold` predicts folding pathways by combining information from the folding dynamics of the predicted conformational landscape (see section 3.6.2). The main goal of this step is to generate and assemble the complete set of folding pathways.

3.5 Modeling Protein Ensembles

3.5.1 Model representation of a protein

`efold` is an algorithm designed to make simultaneous coarse-grained predictions of *i*) the complete conformational energy landscapes, *ii*) folding pathways and *iii*) residues contacts of a protein sequence. Conceptually, each protein structure is described by a coarse-grained residue-level representation. This representation models residue contacts between SS elements. A protein structure is defined by the set of residue-residue contacts that form hydrogen bonds between β-strand backbones. `efold` focuses on β-strand backbones because they are present in the majority of reported folds (60%). Additionally, information about α-helices could be accurately inferred from evolutionary sequence information (decreasing the complexity of the algorithm). The protein representation by `efold` includes side-chain orientation and long-range contacts. The proposed level of abstraction by `efold` enables efficient design of a combinatorial scheme for exploring the complete conformational landscape while retaining enough information to allow further 3D structure reconstruction. Furthermore, representing proteins in this manner allows

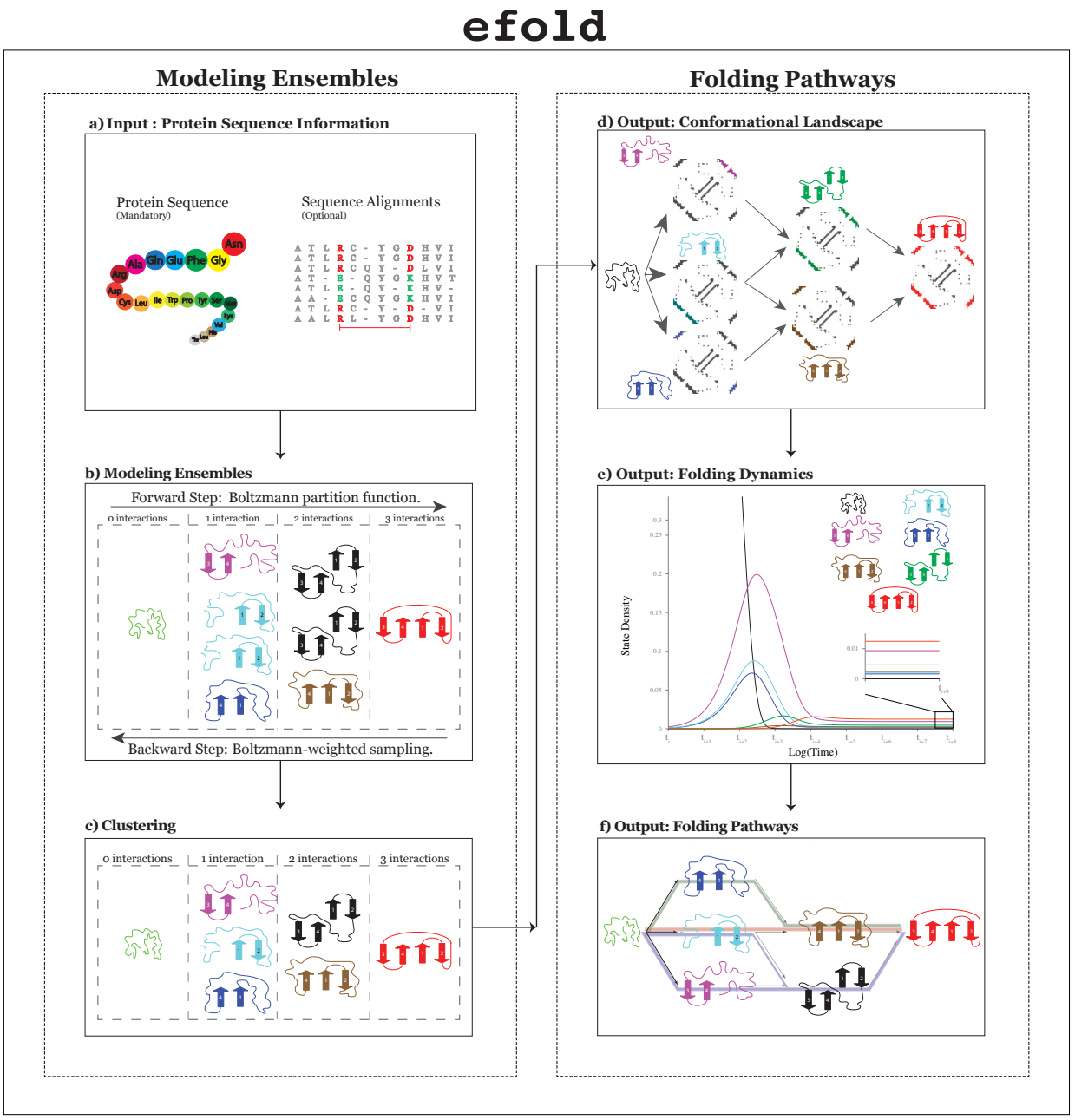

FIGURE 3.1: Schematic representation of `efold`, the proposed algorithm for predicting protein folding pathways using ensemble modeling and evolutionary-based information. **a)** Input consists of a single AA sequence, a set of parameters controlling the size and complexity of the conformational landscape to be explored, and an optional MSA of homologous proteins. **b)** The ensemble technique to predict β-sheets structures consists of a forward and backward traversal over the data structure (i.e., tree) that models the hierarchical folding mechanism(s) and stores all possible proteins states characterized by an energy objective-function. **c)** `efold` partitions the state space into macro states to work with a tractably sized system. **d)** The conformational landscape is represented as a graph, where nodes represent clusters of energetically accessible conformation states and edges model the presence of structural similarity between the states. **e)** The dynamics of the system are calculated by treating the folding process as a continuous time discrete state Markov process. **f)** The transition from a random coil to the native state is represented as a path in a graph (or flow network) of varyingly folded protein conformation states. This graph is built using information extracted from the conformational landscape and the folding dynamics (items d and e, respectively).

for a reduction in the complexity of the conformational search space. The number of protein conformations are still greatly flexible (e.g., permutation of strands, strands' sizes, orientation of side chains, secondary structure motifs, etc.), and structures may take on various conformations that are vastly different between neighbors and the native form.

3.5.2 Calculating the complete set of all possible structural states

The conformational landscape is computed through the enumeration of all coarse-grained protein states that a protein may attain. The SS topologies are encoded using a stepwise permutation algorithm through the labeled set of β-strands $\{1 \ldots n\}$. For each permutation, the set of all β-strand/β-strand pairings were computed such that each interaction in the β-topology is assigned to be parallel (**P**), anti-parallel (**A**), or none (**N**) (see figure 3.2 for an example of the computed topologies and strand pairings of size three). We store this set of topologies in a balanced tree, where each level m of the tree contains the topologies with m β-strands (see figure 3.3 for an example of the computed tree). It is important to stress that each possible topology (see figure 3.2 **a)**) will have a tree as the one depicted by figure 3.3. In these trees, each node (except the root) has one parent node, $m-1$ sibling nodes, and m children nodes. All parent nodes share a common structure with their children, where two topologies share their structures if they are identical to each other, modulo the addition or removal of a single β-strand pairing. The added (or removed) β-strand pairing has to be located at the right (see figure 3.3 a)) or left (see figure 3.3 b)) of the topology represented by its parent (or child). This last feature is very important because it defines a Dynamic Programming (DP) approach. A DP problem should have optimal sub-structures,where the solution for the sub-problem is part of the solution of the original problem. In the next section, the implementation of a DP algorithm to efficiently compute the energy value of the SS stored in the tree will be described.

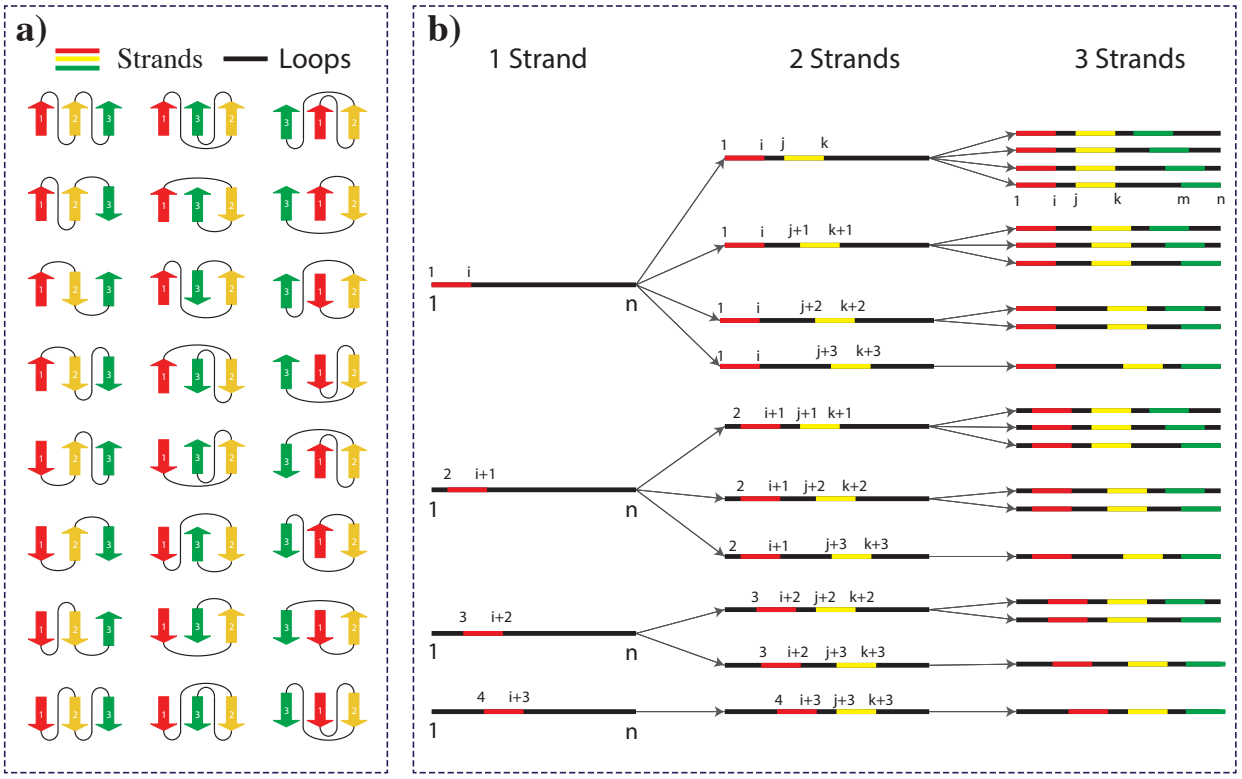

FIGURE 3.2: Secondary structure topologies and β-strand/β-strand pairings for up to three strands

a) The SS topologies computed using a stepwise permutation algorithm through the labeled set of 3 strands. Two strands (arrows in the figure) pointing to the same(different) direction represent a parallel(anti-pararell) interaction. **b)** The set of all β-strand/β-strand pairings computed by `efold`. Each topology represented in **a)** has an associated hash table indexing all possible pairings computed for each topology.

3.5.3 Computation of a single energetic value

An energy function is an equation that relates, mainly, structure to energy. It is desired that this function adequately represent the forces responsible for protein folding. Quantum mechanics computations must be considered in an energy function in order to have an accurate representation of molecular interactions. However, the computational complexity of this approach makes its implementation impractical. Classical physics is a common approach to overcome with modern computational limitations. The protein energy calculations become much more tractable when turning to empirical potential energy functions. However, the process is still computationally demanding when routinely used in a simulation. The parameters of each component's contribution in an energy function have been optimized to effectively represent real protein systems. Typically, a potential

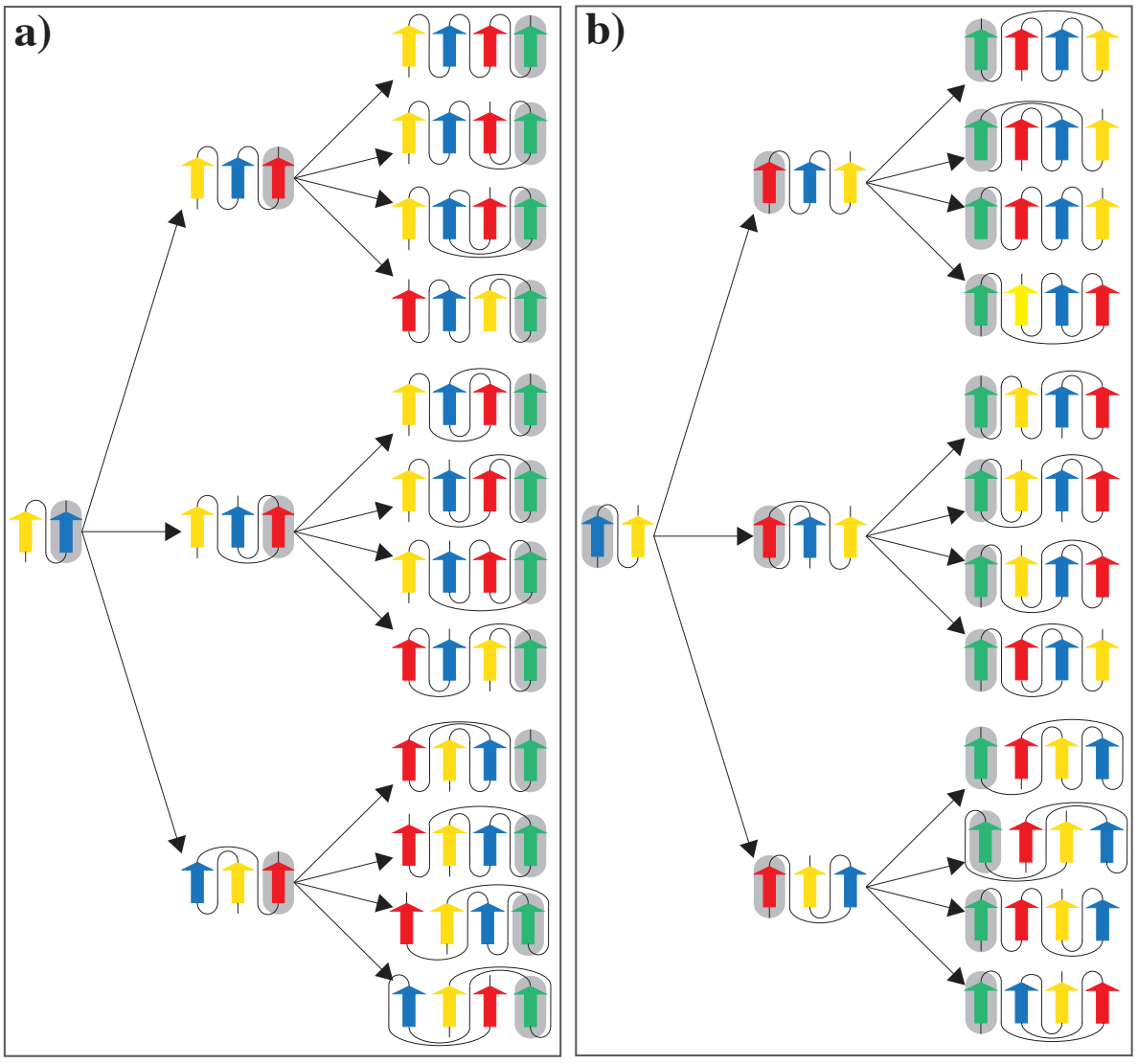

FIGURE 3.3: Tree data structure that stores all the set of secondary structure topologies.

The proposed data structure to store all the set of protein topologies is a balanced tree, where each level m of the tree contains the topologies with m β-strands. In this tree, each node (except the root) has one parent node, $m-1$ sibling nodes, and m children nodes. This data structure can be filled by adding new strands (i.e., green strands in the figure) to any of the extremes of previous topologies. A reader should note that if a green strand is deleted, the previous topology (i.e., the parent node) will be obtained. Figures **a)** and **b)** show the tree when new strands are added to the right and left of the base topology, respectively.

energy function models contributions from the hydrophobic effect, the electrostatic force between AA, the Lennard-Jones potential, and vdW force etc. With the need to intensively compute the potential energy function while exploring the search folding space and lacking an appropriate energy function to minimize, an efficient approximation is to only consider pairwise force contributions.

The use of the Boltzmann distribution to define energy functions with terms involving AA pairwise forces has been widely explored in the field [188, 189]. The main idea is that we can compute the AA potentials from data stored in representative protein databases, where the average distance between a given pair of AA corresponds to the average energy contribution of this pair. The assumption is that AA residues in close proximity within a folded 3D protein structure exhibit marked statistical preferences. We compute a frequency for distances between AA pairs and use the Boltzmann distribution to compute the energy term (i.e., $E(w)$) that corresponds to AA pair potentials. Recalling the definition of the Boltzmann distribution (see equation 3.1), we compute the probability $p(v)$ of state v from a protein database as a frequency for distances between each pair of AA.

$$p(v) = \frac{\exp[-E(w)/RT]}{\sum_{w \in V} \exp[-E(w)/RT]} \tag{3.1}$$

`efold` uses the Boltzmann-distributed pseudo energy function described in [190, 191]. The pairwise frequencies used to compute the energy potential were determined from the specific residues involved in the SS topologies found in a large set of globular proteins whose tertiary structure is known. The proteins were extracted from the PDB-select 25% data base list of June 2000. In order to work with a non redundant version of the PDB, this set guarantees that no two proteins have sequence similarity greater than 25%. These pairwise probabilities are computed according to their environment (i.e., it distinguishes contacts occurring in a hydrophilic environment from those occurring in a hydrophobic environment), but independent of the context in which they are used. Finally, these probabilities are transformed into an energy potential by taking the negative logarithm of the frequencies (see equation 3.2). An energy $E_{i,j}$ is assigned to each residue/residue pair following equation 3.2, where Z_c is a statistical re-centering constant and $p(i, j)$ is

the pairwise probabilities of two residues appearing in a β-sheet environment as observed across all nonsequence-homologous solved structures in the PDB.

$$E_{i,j} = -RT[log(p(i,j)) - Z_c] \tag{3.2}$$

`efold` uses a statistical-mechanical framework to characterize the energetic landscape of a protein through the modeling of all coarse-grained possible conformations that this peptide can attain. The potential energy of a protein conformation is related to the sum of potentials for all residue/residue interactions (equation 3.3), where i and j represent the positions of two AA being computed that belong to the set of all the possible residue pairs, λ. We then assign separate likelihoods based on the hydrophobicity of the environment on either face of a β-sheet (i.e., an AA buried in the core of the protein or exposed at the surface).

$$E(S_n) = \sum_{i,j \in \lambda} E_{i,j} \tag{3.3}$$

For a structure with n strands, the computation of the energy by summing the energetic contribution of each strand interaction is a prohibitive approach. It has previously been shown that a much more efficient method exists using DP. `efold` uses a DP algorithm to exploit the shared structure between protein topologies by performing the recursion shown in equation 3.4. In equation 3.4, $E(S_{n-1})$ is the interaction energy between the first $n-1$ strands and $Pair(s_{n-1}, s_n)$ is the energy of the pairing of strand $n-1$ with strand n (see figure 3.4a for further details). Each recursive call determines the energy of a specific protein conformation and stores this value in a hash table. Subsequent recursive calls, which involve the same conformation, perform a table lookup to prevent re-computation of the energy function. Conformations are partitioned by the location of four indices i_1, i_2, i_3 and i_4, which denote the boundaries of region(s) occupied by the n strands (see figure 3.4).

$$E(S_n) = E(S_{n-1}) + Pair(s_{n-1}, s_n) \tag{3.4}$$

3.5.4 Computation of the Boltzmann partition function

Boltzmann reasoned that in an ideal gaseus environment having N molecules, the number of N_i molecules having energy E_i must satisfy equation 3.5, where k is the *Botlzmann constant*, and Z is the *partition function* that satisfies equation 3.6 (n is the number of possible energy states).

$$N_i = N\frac{\exp[-E(i)/kT]}{Z} \tag{3.5}$$

$$Z = \sum_{i=1}^{n} \exp[-E(S_i)/RT] \tag{3.6}$$

For applications in computer science, the Boltzmann constant k is often dropped, and the Boltzmann distribution (a.k.a. Gibbs distribution) is defined by equation 3.7. At high temperature T, the Boltzmann distribution is close to a uniform distribution and at positive temperature close to 0, the distribution is concentrated near the global energy minimum. The latter fact has been exploited to devise algorithms to solve complex combinatorial problems.

$$P_T = \frac{\exp[-E(i)/T]}{Z} \tag{3.7}$$

`efold` computes a BPF over all n structural states to characterize the energetic landscape of a protein as stated by equation 3.6. In this equation, $E(S_i)$ is the free energy of the structure for the input sequence as computed by equation 3.3. With the partition function Z available, the Boltzmann probability for all structures may then be computed (equation 3.8). Therefore, the Boltzmann probability statistically characterizes the protein ensemble.

$$P(S_i) = \frac{\exp[-E(S_i)/RT]}{Z} \tag{3.8}$$

`efold` needs to coherently incorporate the evolutionary sequence information into its statistical residue contact potentials. Protein conformations evaluated by our mechanical

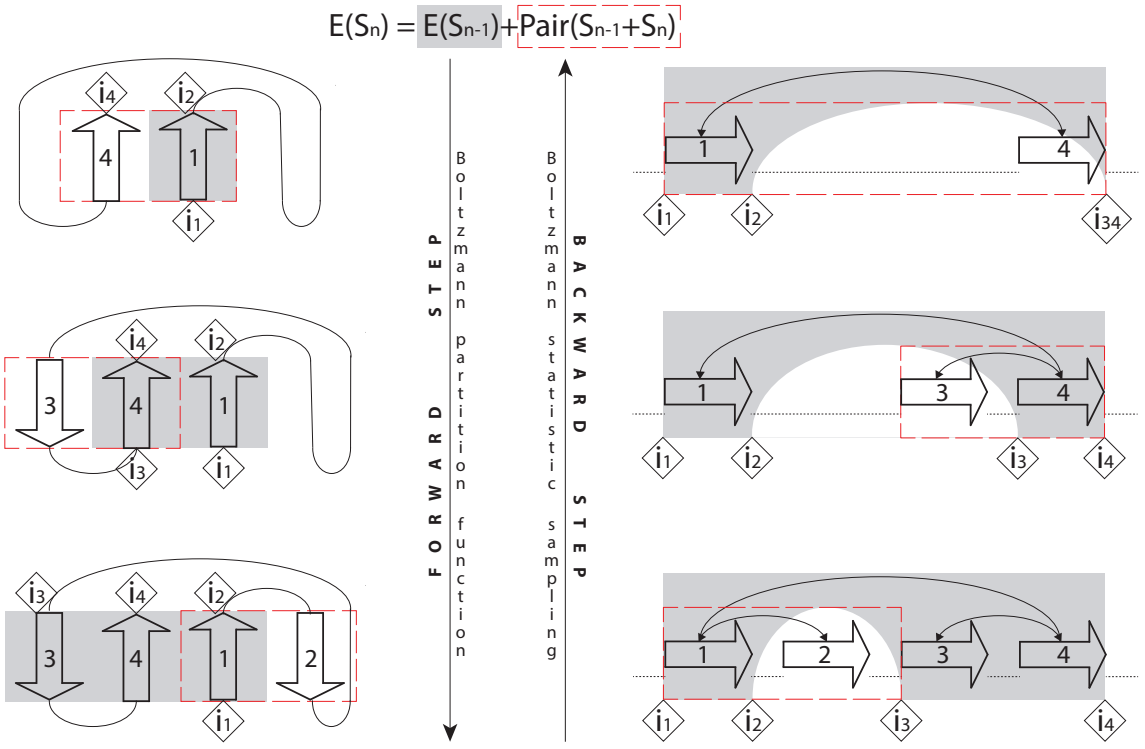

FIGURE 3.4: Dynamic programming strategy encoded by `efold`.
An illustration of how `efold` encodes a permutable β-template representing the Protein G (i.e., $\beta 3\mathbf{A}\beta 4\mathbf{P}\beta 1\mathbf{A}\beta 2$). `efold` recursively computes the Boltzmann partition function through an energy function composed by the sum of the contributions of the last two strands and the remaining structure (left column). A sampling procedure is performed through the traceback of intermediate structures (right column).

statistical approach ($P(S_i)$) are penalized or rewarded depending on the agreement of the residue-residue interactions predicted by the evolutionary-derived residue information through the function $P_{evol}(S_i)$. The level of the contribution for this information contained by each approach (i.e., mechanical statistical and evolutionary) is the total score of the objective function adjusted using parameter γ. A value of one for this parameter puts all weights on statistical contact potentials, while a value of zero has weights relying on evolutionary inferred contacts (see equation 3.9).

$$ObjectiveFunction_i = \big(\gamma \times P_{evol}(S_i)\big) + \big((1-\gamma) \times P(S_i)\big) \tag{3.9}$$

`efold` is required to calculate the proposed objective function for all attainable protein topologies. `efold`'s time complexity depends on the computation of the BPF for each

topology. Figure 3.5 depicts the dependency of `efold`'s complexity on the two primary factors influencing this calculation (the length of sequence and the number of strands in the topology [i.e., the depth of the recursion]).

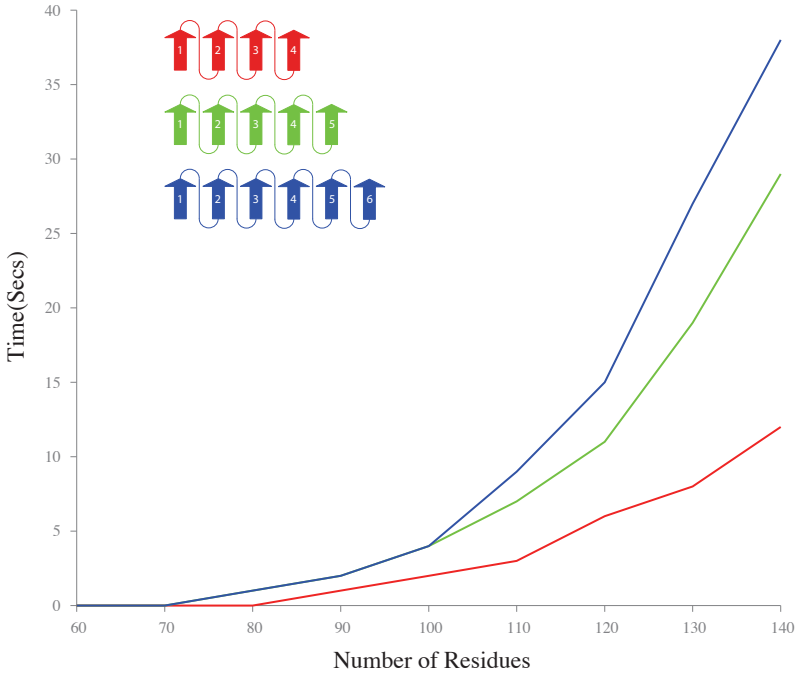

FIGURE 3.5: Runtime complexity curve for the computation of the partition function by `efold`.
The time was computed by averaging over five independent runs, for sequences ranging from 40 to 140 residues in length, with 4, 5 or 6 strands. The experiment environment is a 2.8 GHz Intel Core i7 system, with 4GB of RAM, under the Mac OS X operating system.

3.5.5 Generation of a statistically representative sample

Once `efold` has characterized the energetic landscape through the computation of the objective function for every possible SS, it extracts a statistically representative sample of SS from the computed landscape. `efold` uses a recursive statistical algorithm inspired by [80] to sample protein conformations according to their values as reported by the objective function (see equation 3.9). Since we do not know the native structure, we have to sample configurations from all possible permutations of the β-sheet topology. The algorithm works as follows: *i*) For each topology, we use a fitness proportionate selection algorithm (also known as roulette wheel selection) to select the potentially useful solutions

from all the conformational states computed for this β-sheet topology (see section 3.5.4). In this approach the candidates solutions with a higher fitness will be more likely to be selected by a random selection procedure. This algorithm simulates the behaviour of a roulette wheel in a casino, where an equal proportion of the wheel is assigned to each of the possible selections based on a fitness value. The fitness value for each β-sheet topology is obtained from the computation of equation 3.9. A random selection process is performed mirroring the process of how the ball (in a casino) falls in one of the wheel's compartments after the roulette wheel is rotated. *ii*) Once a candidate solution is selected (i.e., a β-sheet conformational state), we extract from it the location of a single strand. This single strand corresponds to the last strand added during the computation of the DP algorithm (see section 3.5.3). The location of a single strand is sampled from the region indicated by indices i_2, i_3 in Figure 3.4. The extraction of this single strand creates a different β-sheet topology (i.e., one β-sheet with exactly one less strand). Then, `efold` comes back to step *i*) to perform the statistical sampling on the new β-sheet topology. It is important to mention that the new fitness selection will be performed on structural conformations that do not conflict with the strands that have been already extracted by previous recursions of the same call of the sampling algorithm. *iii*) Finally, the algorithm terminates when the size of the sampled β-sheet topology is equal to one. The aforementioned process is repeated many times to generate the population of protein conformations.

3.5.6 Clustering protein conformations

Since thousands of states are sampled from each strand topology and a tractably sized system is desired, `efold` partitions the state space into macro states using clustering. The clusters are built based on the structural similarities of samples, where similarity is measured using a conformational metric. Given a sequence of AAs, an ensemble may contain thousands (or even millions) of distinct structures. In a clustering approach, we need to be able to efficiently quantify how similar these different structures are in order to group by similarity. In the literature, a range of metrics for comparing folded structures are already available based on contact, segment overlap and mountain values as similarity measures of SS[192, 193]. `efold` implements a version of each metric, but

a contact based metric was selected as the default option given that it performed best empirically for computing the conformational similarity of protein SS. The contact based distance between two SS is the total number of AAs apart that the contacts occurring in one structure are located with respect to the other structure. Once a distance is defined between pairs of structures, it is straightforward to compute an array of distances between all pairs of structures in a sample. A standard hierarchical clustering algorithm is then used to partition the foldings into clusters based on the array of distances. Finally, the clusters with highest energy values are chosen to represent energetically accessible and favourable conformation states (i.e., intermediate folding states).

3.5.7 Modeling folding transitions

Next, `efold` associates each cluster with an intermediate folding state. These clusters represent energetically accessible and favourable conformation states that can be represented as nodes in a graph of varyingly folded protein conformation states. `efold` predicts coarse folding transitions by modeling the transition between intermediate folding states as edges in the graph. For every pair of vertices in this graph, a transition edge is added if both states have compatible topologies and structural similarity. Two topologies are compatible if they differ only by the addition or the removal of a single strand pairing. In the case of `efold`, the mechanism of protein conformational change (i.e., addition of single strands) implies that any state linked in the trees shown in figure 3.3 would be topologically compatible. The similarity between two clusters (each represented by a node in the graph) is estimated using the structural metric used to cluster the samples (see section 4.2.4.4).

3.6 Folding Dynamics

Novel computational paradigms are required to model and understand the conformational dynamics of proteins using an ensemble view of protein structure. In principle, MD simulations provide the time evolution of proteins, but the computational cost and

long time scales associated with this technique make it difficult to compute an ensemble of trajectories. It has already been shown that a discrete model of RNA SS folding is capable of describing the folding dynamics at macroscopic timescales beyond the reach of methods that operate at an atomic resolution (i.e., MD) [194], but the simulation of protein dynamics have been shown to be harder. Simplifications regarding the spectrum of time scales and number of sampled conformations are needed to reduce the complexity of the protein folding dynamics into manageable models. This reduction provides significant advantages in terms of effectively focusing on the thermodynamically and kinetically relevant motions by eliminating the information from irrelevant protein motions. Given the appropriate degrees of freedom that describe rate-limiting protein motions, an energy landscape has been shown as a suitable protein dynamic model. Furthermore, some statistical frameworks (such as the Markov model that will be described in this section) have already been developed to describe the conformational dynamics of proteins in terms of conversions between the conformational states. Similar conformations are categorized into states based on structural metrics (such as the metrics described in section 4.2.4.4). The rate of interconversion between states are estimated from simulated trajectories or energy estimations (such as the energy function described in section 3.5.3). A methodology that provides a statistical framework for a human-readable view of folding dynamics is presented in this subsection.

Protein ensembles (see section 3.3) provide key features to model protein dynamics. This approach provides a structural definition of key conformational states for a protein (i.e., nodes in a graph of varyingly folded protein conformation states), the mechanism of protein conformational change (i.e., addition of single strands), and the height of barriers connecting these key conformations (i.e., the difference between the free energies of the states). A protein energy landscape can be constructed by connecting the conformational states together and estimating the transition rates between pairs of interconverting structures. This results in an ordinary differential equation (ODE) system (also known as a master equation) that can be solved to predict and characterize protein dynamics. In other words, the folding dynamics of a protein is described as motion on its conformational energy surface.

Master equations are a set of ODEs used to describe the time-evolution of a biological system that can be modeled as being in exactly one of many countable states at any given time, where switching between states is treated probabilistically. These ODEs model the variation over time as probabilities that the system occupies for each of the different states. Starting at some initial time t_0 (i.e., when the protein is exactly in the unfolded state), we compute the probabilities $P_A(t)$ and $P_B(t)$ that one given protein system is in state A or B, at time t. We derive a differential equation for $P_A(t)$ by relating the probabilities at the two closest times, t and $t + dt$. The previous formulation implicitly uses the Markov assumption as the transition rate from one state to the other during the time interval $(t+dt)$ does not depend on the previous states $[(t-k\times dt)$ where $k > 0]$ but only on the state at time t. The master equation describes the evolution of probabilities for Markov processes of the system.

Modeling conformational changes by solving the master equation (i.e., by modeling the problem as a continuous time, discrete state Markov model) is a good approximation to overcome the lack of quantitative and qualitative data of folding kinetics for proteins. The folding process of RNAs has been already modeled as a time-series of RNA SS such that the elementary transitions are the opening or closing of a single base pair nucleotide(s) [195]. The formation and deletion of helices as the move set were also proposed by other methods for simulating RNA folding dynamics [196]. The master equation formalism has been also developed for protein folding kinetics, where the folding of simple lattice chains allows transitions between states at only local conformational changes [197]. The folding mechanism of SS formation and the enumeration of all the conformations of a simple lattice model [198] have been proposed to explore possible kinetic and folding pathways. Previous research developed in our research laboratory has already allowed the computation and visualization of the dynamics of the folding process occurring in $\beta-$sheet proteins [121].

3.6.1 Continuous time discrete state Markov model

Following conventional stochastic kinetics theory [199], the probability distribution P of protein SS as a function of time (i.e., the kinetics of the molecule in terms of its

macrostates) is given by the master equation shown in 3.10. Within this stochastic formulation, k_{ij} is the probability that a transition from a distinct state i to another distinct state j occurs within the infinitesimal time interval dt.

$$\frac{dP_t(i)}{dt} = \Big[\sum_{j \neq i} P_t(j)k_{ji} - P_t(i)k_{ij} \Big] \tag{3.10}$$

The formulation of a square transition matrix $R = (r_{ij})$, which contains the transition rates between different states of the system, is needed to find a solution of the equation 3.10.

$$r_{ij} = \begin{cases} k_{ji}, & \text{if } i \neq j \\ \text{-} \sum_{l \neq i} k_{li}, & \text{if } i = j \end{cases} \tag{3.11}$$

Equation 3.11 can be rewritten in matrical form as follows:

$$\frac{d}{dt} P_t = RP_t \tag{3.12}$$

We are interested in calculating the temporal distribution vector P_t (i.e., the distribution over states of the system at time t), which can be calculated from the explicit solution of equation 3.13, where P_0 is the initial distribution vector. The probabilistic behavior of a discrete state continuous time Markov model is completely described by the initial state (or distribution) and the transition rates between distinct states. `efold` computes a distribution vector P_t for 241 different time-steps as default (from time -6 to $+6$ with steps of 0.05).

$$P_t = \exp(Rt)P_0 \tag{3.13}$$

What still needs to be established is a rule for the rate-constant k_{ij} between two SS i and j of the conformation space. All elements of the transition matrix R that do not correspond to single moves (i.e., transition between nodes that are not connected by an

edge in the graph of folding transitions [see section 3.5.7]) are assumed to be zero. The rate between two non-zero transition elements is computed using the symmetric Kawasaki rule [200]. This rule was chosen given that it takes uphill and downhill steps into account, which avoids an intrinsic diffusion limit (i.e., forming a favorable contact does not increase with the contact's favorability) [201]. Let G_i be the free energy of the SS i from which an allowed move to structure j with the free energy G_j is made. Then, the transition probability k_{ij} is given by the Kawasaki rule as:

$$k_{ij} = \exp(-\Delta G_{ij}/2RT) \tag{3.14}$$

where $\Delta G_{ij} = G_j - G_i$. This gradient ΔG is an important determinant of the speed at which the system moves uphill or downhill. By using the Boltzmann coefficient, the uphill/downhill steps become more rare as ΔG increases. The Kawasaki dynamics approaches the Boltzmann distribution at equilibrium because it satisfies microscopic reversibility [202]. `efold` calculates the ensemble free energy difference ΔG_{ij} between two macro states i and j by summing over all the states from which both states are composed (equation 3.15). The non-zero transition elements are considered to be consistent with the free energy difference of the two conformations involved. Given that two states are connected in the graph, the rate at which they interconvert is proportional to the difference between the free energies of the states (ΔG). Note that under this approach, energy barriers are not explicitly incorporated into the model, since entire SS (i.e., $\beta-$strands) are either added or removed between states without partially-formed intermediates.

$$\Delta G_{ij} = E(\chi_i) - E(\chi_j) = \sum_{x \in \chi_i} E(x) - \sum_{x \in \chi_j} E(x) \tag{3.15}$$

3.6.2 Folding Pathways

A structured folding pathway is defined as a time ordered sequence of folding events in which the unfolded protein is able to assume its native state. The folding pathways are fully described by the complete conformational energy landscape, where proteins fold

through distinct intermediate conformations via multiple routes. Similar to other models that represent a protein at the level of SS elements [92, 203, 204], `efold` represents intermediate conformations as ensembles of fully folded structures containing a set of interacting SS (see section 3.5.7). `efold` predicts coarse folding pathways by finding a collection of intermediate transitions that connect the unfolded and native protein states. In other words, `efold` represent folding trajectories (from a random coil to the native state) as paths in a graph (that represent an energy landscape) of varyingly folded protein conformation states. The intermediate conformations are modeled by `efold` as obligatory sequential states in the overall folding process. Unfolding transitions (represented by cycles and backward transitions in the graph) are not considered during the computation of folding pathways. The sequentiality of folding events (assumed by `efold`) implies that certain transitions occur before others. `efold` extracts the order of folding events from the folding dynamic simulation previously performed (see section 3.6). `efold` combines folding dynamics and conformational energy landscape information to model folding pathways as weighted paths that connect the unfolded, intermediate, and native protein states.

The fact that `efold` does not require *a priori* information about the native structure of the protein is a differential feature with respect to state of the art PPP predictors. Unlike most of the PPP predictors, `efold` is able to extract the (predicted) native topology out of all sets of possible protein topologies. If `efold` has access to sequence information (see section 3.2), the HMM reported in [183] is used to derive a predicted native topology. This HMM detects patterns in matrices (formed from sequence information) corresponding to predicted interactions between SS elements. These interactions are defined on the basis of regular patterns of hydrogen bonds, which connect residues in different β-strands in either a(n) parallel or antiparallel fashion. The output of the HMM is a sequence of interactions between two β-residues, a direction for the interaction (parallel or antiparallel), and a measure of confidence (Viterbi score) of the corresponding prediction. From this information, `efold` retains the set of the best consecutive β-residues that form k topologies, where k is the number of strands provided by the user as input. It is important to stress that k selected topologies must satisfy the topological constraints associated with β-strand pairing (i.e., each β-residue can have a maximum of two β-partners and

there must be a minimum sequence separation between β-strands). Based on this list of well-ranked β-residues, `efold` infers the topology of the native protein by rearranging and labelling the β-strands. If `efold` is constrained and does not have access to sequence information (i.e., the user explicitly wants to generate a prediction based solely on ensemble modeling), `efold` will predict the native topology as the topology of size k with the best probability as computed by equation 3.8.

Once a native topology has been predicted, `efold` selects a subgraph (out of the complete conformational energy landscape graph) that contains all transition states involved in trajectories connecting the unfolded state with the predicted native topology. `efold` assigns a weight for each transition (i.e., edge in the subgraph) based on the folding dynamic predictions previously computed (see section 3.6). The weight of a connection between two conformational states i and j (i.e., an edge in the subgraph) is computed by equation 3.16. In this equation $P^i_{t_\ell}$ corresponds to the occurrence likelihood of the conformational state i at time ℓ as computed by equation 3.13. ℓ is the simulation time for which `efold` computed the maximum likelihood for state j. $Q_{i,j}$ is the set of states that share an incident edge with j. Equation 3.16 can be understood as the transition probability of going from state i to state j in time ℓ.

$$weight_{i,j} = \frac{P^i_{t_\ell}}{\sum_{s \in Q_{i,j}} P^s_{t_\ell}} \tag{3.16}$$

The size of the computed subgraph is many orders of magnitude smaller than the original graph and its tractably size allows for a complete exploration of protein pathways. `efold` performs a DFS traversal in this subgraph to extract the set of trajectories that connect the unfolded, intermediate and native states. The weight of each path recovered from the DFS traversal is estimated as the minimum of edge weights between two nodes comprising the folding trajectory. This weight represents the probability that this specific trajectory arises from the network of folding pathways.

The analogy of considering protein folding as a flow arising in a network of (un)folding pathways at a coarse grained free energy landscape has been already adopted by previous studies [205–207]. Following this analogy, each edge's weight (a.k.a. capacity) represents

the maximum amount of flow that can pass through this specific edge. The weight of each reported pathway (a.k.a flow) represents the probability that this specific trajectory arises from the network of folding pathways. The unfolded (which will be called the source) and native (which will be called the sink) nodes are distinguished from this network. In this modeling, the flow satisfies the restriction that the amount of flow into a node equals the amount of flow out of it (except for the source and sink, which have only outgoing and incoming flow, respectively).

Chapter 4

Experimental Framework and Results

4.1 Experimental Framework

`efold` models secondary structure elements through a coarse-grained representation. This representation enables an efficient design of the combinatorial scheme for exploring the complete conformational landscape, while retaining enough information to allow further 3D structure reconstruction [208]. `efold` allows for the complete enumeration of all admissible β-sheet topologies and computation of the Boltzmann partition function over all attainable protein topologies. These topologies are then scored (or penalized) based on their agreement with evolutionary inferred contacts. Finally, `efold` enumerates and ranks all β-sheet topologies to build a coarse-grained energy landscape that is then used to calculate the folding dynamics (see section 3.3 to review all the methodological details).

`efold` predicts folding transitions from a random coil to the native state as a path through varyingly folded protein conformation states. This transition through conformation states is represented as a graph, where vertices are the most energetically accessible and favourable conformation states for a given topology (previously generated by the Boltzmann ensemble sampling method). Graph edges represent the transition

between two conformation states, where conformations connected by an edge are compatible topologies with similar structure. Finally, the graph is modeled as a flow network to represent folding trajectories as paths in the graph (see section 3.6.2 for a review of `efold` methodological details).

Given the stochastic nature of `efold`, we run one hundred independent runs of the algorithm for each prediction during the proposed experimental framework. `efold`'s predictions are then computed as the average across the 100 runs. The main goals of performing multiple runs are the obtention of an accurate representation of what `efold` really predicts and the obtention of trajectories with small folding flux (which could not appear if the number of simulations is small). The prediction of trajectories which are less populated is important given that those trajectories could represent new insights about alternative folding routes in proteins. The number of runs (i.e., 100) was selected because it represents a fair balance between the required computational resources and the goals aforementioned.

Due to the limited experimental folding dynamics data available, a complete validation of the protein folding predictions is difficult. Nonetheless, the versatility and speed of `efold` allows for a broad range of experiments that aim to validate directly (when possible) and indirectly the performance of our method. Figure 4.1 depicts the proposed experimental framework to validate the predictions of `efold`. Four main experiments were performed to estimate the precision of `efold`'s pathway and structure predictions.

In **Experiment 1**, the precision of `efold` to predict coarse-grained folding pathways (i.e., sequences of representative intermediate states modeled as β-residue contacts) is computed. This computation is performed by simulating the folding landscape of proteins for which pathways have been elucidated through previous experimental studies and MD simulations. To be more specific, the folding landscape of two gold standard proteins (Protein G and Ubiquitin) are explored (See section 4.1.2 for more details). These proteins were chosen because they have played a central role in protein folding studies by being the system of choice in a vast body of experimental and theoretical studies. The results of this experiment can be seen in section 4.2.1.

Given the diversity of protein structures (i.e., the native and denatured states), and differences in sequence AA compositions, extracting the general rules of protein folding is difficult when only studying the folding of individual proteins. Two different strategies have been shown suitable in biophysical studies to extract general rules from the protein folding process. The first approach works by comparing the mechanisms of proteins with a high degree of sequence identity, but different 3D structures (or pathways). The second strategy studies proteins sharing the same overall fold, but different sequence (i.e., members of the same protein family).

Experiment 2 follows the first strategy and investigates the folding properties of engineered proteins with a high degree of sequence identity, but with different 3D structures and/or different folding pathways [209–213] (see section 4.1.3 for more details). This set of proteins represents a great opportunity to elucidate relationships and dependency between `efold`, sequence information, protein structure and folding mechanisms.

Experiment 3 adopts an opposite point of view and analyzes the rules that govern the folding of several proteins families (Pfam) [214] by comparing proteins that differ in sequence but share the same overall fold. Sixteen proteins belonging to four Pfam families (PF00018, PF00014, PF00240, and PF01423), with available folding pathway data, were selected to investigate the existence of common folding intermediates between the members of the same protein family (see section 4.1.4 for more details of the data benchmark). The results of **Experiments 2 and 3** will be explored in sections 4.2.3 and 4.2.2, respectively.

The prediction of residue-residue contacts has already been proved useful in reconstructing protein backbones by providing information to determine accurate 3D protein structures [179]. Accurate prediction of SS and residue contacts is a major step to correctly predicting folding pathways. In general, the predictors of residue-residue proximity in folded structures are based on the existence of interdependent changes in groups of variable AA belonging to a protein family of homologues. Even though `efold` is not an algorithm developed to predict residue-residue contacts, we evaluated the prediction capabilities of `efold` to recognize contacts involved in SS. Therefore, **Experiment 4** evaluates the precision and sensitivity of `efold` for predicting residue-residue contacts and SS on a

data set of 125 proteins (see sections 4.1.7 and 4.1.5 to review the validation metrics and protein benchmark, respectively). **Experiment 4** also studies the performance of `efold` when compared to the state of the art algorithms. The results of this experiment are described in the section 4.2.4.

The complete set of proteins used during the experimental framework are listed in Appendix A.

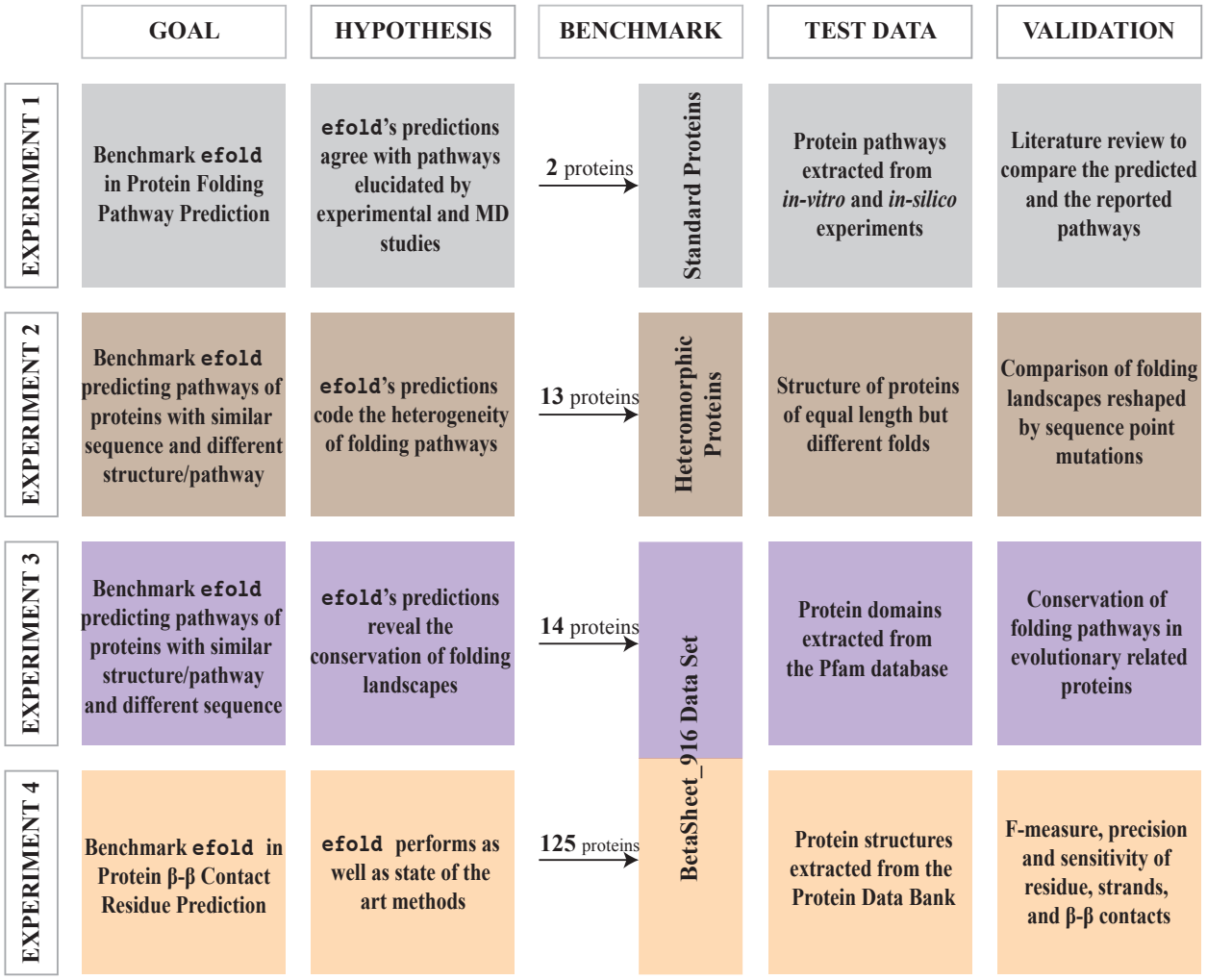

FIGURE 4.1: Experimental Framework to validate `efold`
The experimental framework is composed of four main experiments (represented by rows in the matrix). Each experiment focuses on one research goal hypothesis (represented by the first and second column in the matrix) through the analysis of `efold`'s simulations on a specific protein data benchmark (represented by the third column in the matrix). The predicted data is compared agains protein test data (represented by the fourth column in the matrix) using different validation protocols and metrics (represented by the fifth column in the matrix).

4.1.1 Input and Parameters

`efold` inputs consist of *i*) a single AA sequence, *ii*) a set of parameters controlling the size and complexity of the conformational landscape to be explored, and *iii*) an optional string (of same length than the AA sequence) representing evolutionary sequence information. For the proposed experimental framework, the AA sequences of proteins are extracted directly from a FASTA file[1] stored in the PDB archive. The set of parameters that control for the size of the conformational landscape includes the number and lengths of β-strands, and the minimum inter-strand loop size. These parameters control the size of the landscape by imposing steric and biologically derived constraints that valid foldings must satisfy. The values of these three parameters are directly extracted from the corresponding pdb file[2]. The number and (maximum and minimum) length of $\beta-$strands are extracted from the pdb section called 'secondary structure assignment', and the minimum inter-strand loop size is defined as the minimum difference of AA positions between any two $\beta-$strands. The last input (a string representing evolutionary sequence information) allows for the identification of residue contacts from evolutionary information and/or predicted secondary structures. The contribution of the evolutionary information to the score computed by the ensemble-modelling is adjusted using parameter γ. The value for γ is determined by the number of homologous proteins in the MSA associated with the tested protein (i.e., the amount of information available in the MSA). Particularly, this value is obtained from a sigmoidal function that has as range $[0, 4 \times L]$, where L is the length of the sequence. A value of one for this parameter puts all weights on statistical contact potentials (i.e., ensemble-modeling), while a value of zero has weights rely completely on evolutionary inferred contacts (see equation 3.9 in Chapter 3 for further details). The sequence information used in the following experiments were obtained from $\beta - \beta$ residue contact predictions computed by `bbcontacts` (see section 3.2.2).

[1] A FASTA format is a text-based format for representing protein sequences, in which AA are represented using single-letter codes

[2] A pdb format is a text-based format for representing protein molecules. The file provides a description and annotation of protein structures including atomic coordinates, observed sidechain rotamers, secondary structure assignments, as well as atomic connectivity

4.1.2 Protein Benchmark to validate Experiment 1

The precision of `efold` to predict coarse-grained folding pathways (i.e., sequences of representative intermediate states modelled as β-residue contacts) is computed by simulating the folding landscape of proteins for which pathways have been elucidated through previous experimental studies and MD simulation. More precisely, the folding landscape of the Protein G and Ubiquitin is explored. Due to their small protein domains, these proteins have represented ideal candidates for the elucidation of their folding pathways [215, 216].

The B1 domain of protein G, generally called GB1 or Protein G, has represented an ideal candidate for a vast number of different studies because of its small size and simple highly symmetrical topology. Protein G is 56 AA in length (regular α/β structure). Protein G fold consists of 4-stranded β-sheet and an α-helix tightly packed [217]. Protein G folds through three intermediate states. These states feature a near-native helix along with hairpin 1 (I_1 intermediate), hairpin 2 (I_2), or the $\beta 1 - \beta 4$ sheet (I_3). Previous work [218, 219] reported an early formation of the second hairpin ($\beta 3 - \ turn - \beta 4$) and its fundamental role in the folding process. Additionally, this second hairpin centers around known nucleation points W43, Y50, F54 that are strongly stabilized by three hydrophobic residues W43, Y45 , F52 [216]. Different folding pathways are observed, each involving the formation of its own assembly: helix-second hairpin, helix-first hairpin, and $\beta 1 - \beta 4$ sheet. All pathways appear to converge in the same folding nucleus.

Ubiquitin is a small protein (76 residues in length) that has a highly structured native state which is very stable. Its high stability may be linked with the function of ubiquitin. Ubiquitin covalently attaches to lysine side chains in proteins and marks them for degradation by a proteasome. The folding of ubiquitin is two-state under most conditions. However, an intermediate can be stabilized and become populated during folding using a number of methods (e.g., the use of a stabilizing salt such as sodium sulfate [220]).

4.1.3 Protein Benchmark to validate Experiment 2

A novel engineering approach has allowed for a set of proteins to be obtained with high sequence identity but different structure and function [209–211]. Two different wild-type protein domains from streptococcal protein G, called G_A and G_B, show an increasing degree of sequence identity (starting from 1% to 95%). G_A displays a three-helix bundle fold, and G_B is a $\alpha + \beta$ Protein G fold. Table 4.1 shows the percentage identity between different variants of the wild-type protein. This set of proteins represents a great opportunity to elucidate the relationships and dependency between `efold`, sequence information, and folding mechanisms. This comparison of sequence versus structure is also useful when analyzing the sensitivity of `efold` to changes in the AA sequence.

Two mutants of protein G (called N_uG_1 and N_uG_2) were created to alter the proteins folding behavior, while maintaining the same secondary and tertiary structure [212, 213]. A reversed folding pathway (when compared to the wild type) was obtained by increasing the intrinsic stability of the first hairpin and decreasing that of the second hairpin. Results show that the first β-hairpin is now formed first in mutants, and the second hairpin is disrupted by the rate limiting step. Mutants fold 100-fold faster, are more stable, and have a reversed-folding pathway when compared with the wild type, but all of them still maintain a high sequence and structure similarity. Table 4.1 and figure 4.2 show the percentage identity and point mutations between Protein G and different variants of the wild-type protein. These mutants represent an opportunity to study folding pathways within `efold` in cases where several different routes to the native state are equally consistent with the native state topology.

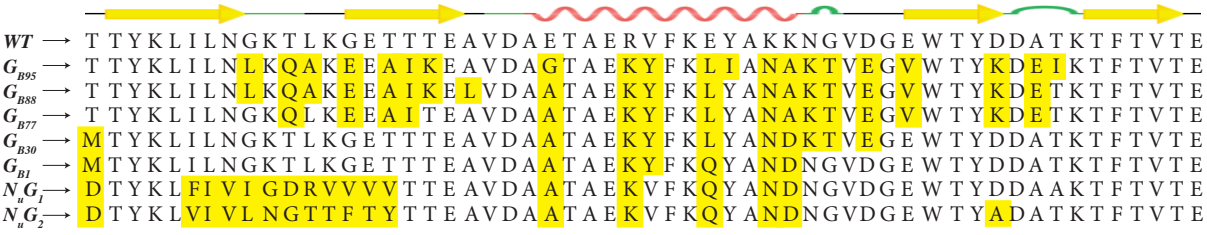

FIGURE 4.2: Folding variants of Protein G
Point mutations of the tested variants of Protein G

Protein	WT	G_{B95}	G_{A95}	G_{B88}	G_{A88}	G_{B77}	G_{A77}	G_{B30}	G_{A30}	G_{B1}	G_{A1}	N_UG_1	N_UG_2
WT	100	63	58	65	54	72	50	83	17	88	13	68	68
G_{B95}		100	95	93	92	90	88	74	54	67	45	54	55
G_{A95}			100	92	97	84	93	68	59	61	50	48	50
G_{B88}				100	88	93	84	77	50	70	42	55	59
G_{A88}					100	81	97	65	63	58	54	45	47
G_{B77}						100	77	84	43	77	34	59	61
G_{A77}							100	61	67	54	58	43	45
G_{B30}								100	31	93	24	68	68
G_{A30}									100	24	92	18	20
G_{B1}										100	17	75	75
G_{A1}											100	13	14
N_UG_1												100	80
N_UG_2													100

TABLE 4.1: Sequence identity between different variants of the wild-type Protein G. The variants identified as G_{A1}, G_{A30}, G_{A77}, G_{A88}, and G_{A95} correspond to the G_A fold, and G_{B1}, G_{B30}, G_{B77}, G_{B88}, and G_{B95} correspond to the G_B fold.

4.1.4 Protein Benchmark to validate Experiment 3

PF0018 - SH3 Domain:

Due to its small size and multiple homologues, SH3 domain has been widely studied to address various important aspects of protein folding, such as the synergistic relationship between experiments and simulations, the nature of protein folding transition state ensemble (TSE), the relationship between protein topology and the folding pathway [221]. SH3 is composed of two orthogonally packed β-sheets that form a single hydrophobic core [222]. The first sheet consists of three central strands of the protein ($\beta2-\beta3-\beta4$). The second sheet contains two terminal strands ($\beta1-\beta5$) and a portion of the RT loop. There is also a small 3_{10} helix between $\beta4$ and $\beta5$ [223]. It has been shown that the structure within the transition state ensemble is highly polarized with a hydrogen bonding network associated between two β-turns. The denaturation of the N and C termini, turns and loops, and a small amount of SS located in the central $\beta2-\beta3-\beta4$ are general features of the SH3 TSE [222] . The distal β-hairpin and diverging turn are formed in the transition state and all conformations in the TSE have the $\beta2-\beta3-\beta4$ formed [224]. Experimental results have also shown that $\beta2$, $\beta3$, and, to a lesser extent, $\beta4$ strands are the most ordered regions of the TSE.

Protein engineering studies suggest that the folding pathways of SH3 domain may be

evolutionarily conserved and that its topology may play an important role in determining the folding pathway of this structure. Furthermore, L24 has been shown experimentally to be involved in the TSE and a highly conserved position in the SH3 fold family [222, 225].

Kunitz Domain:

Kunitz domains are relatively small with a length of about 50 to 60 amino acids. Examples of Kunitz-type protease inhibitors are aprotinin (bovine pancreatic trypsin inhibitor, BPTI), Alzheimer's amyloid precursor protein (APP), and tissue factor pathway inhibitor (TFPI). From them, BPTI is one of the most extensively studied globular proteins and was the first well-documented case of the disulfide folding pathway. BPTI is a Kunitz-type protease inhibitor which comprises 58 amino acids and three disulfides-bonds in its native form. Its structure is a disulfide rich $\alpha + \beta$ fold. Disulfide-bonds occur between cysteine residues 5-55, 30-51 and 14-38.

The BPTI folding pathway is primarily a five-state system that includes the unfolded and native forms. In the second state, the formation of the native disulfide 30-51 predominates. In the third state, non-native disulfides 5-14 and 5-38 rapidly interconvert between each other and the native 14-38, with 30-51, remaining stable. In the fourth state, BPTI must pass through the intermediate, which contains the native disulfides 30-51 and 5-55 [226]. NMR exchange data indicates the formation of a fully folded sheet with subsequent helix formation during the folding process. The pathways involve the full association of the 3-stranded sheet (β1, β2 and β3), followed by the C and N terminal helices α2 and α1, respectively. The initial formation of the 30-51 disulfide is in agreement with the early formation of the $\beta1\beta2\beta3$ sheet and its association with α2. By incorporating α1 in the complex, the disulphide bonds $5-55$ can be formed. Finally, disulphides $14-38$ are formed when the loops form [226]. SS form early during the folding, which is then followed by docking and packing of preformed SS units to form the native tertiary structure [134].

LSm PF01423:

Two sequence motifs (named Sm1 and Sm2) have been identified through the comparison

of various LSm homologs. The size of the Sm1 and Sm2 motifs are 32 and 14 AA long, respectively. The Sm1 sequence motif corresponds to the $\beta 1$, $\beta 2$, $\beta 3$ strands, while Sm2's motif corresponds to $\beta 4$ and $\beta 5$ strands. These sequence motifs are conserved and separated by a non-conserved region of variable length. This fact suggest that all LSm protein genes evolved from a single ancestral gene [227].

4.1.5 Protein Benchmark to validate Experiment 4

BetaSheet916 benchmarking dataset was created from entries in the Protein Data Bank of May 2004 by Cheng [228]. This benchmark contains 916 chains (corresponding to 187,516 residues) determined by X-ray diffraction with a resolution of $2.5\mathring{A}$ or better. All protein chains contain standard AA with a length greater than 50. The sequence identity in the dataset is guaranteed to be $15-20\%$. 48,996 of the residues are β-residues participating in 31,638 interstrand residue pairs. The dataset has 10,745 β-strands with an average length of 4.6 residues and 8,172 β-strand pairs, including 4,519 antiparallel, 2,214 parallel and 1,439 pairs involved in isolated β-bridges. These strand pairs together form 2,533 β-sheets. The average sequence separation between residue pairs and strand pairs is 43 and 40 AA, respectively.

BetaSheet916 set is routinely adopted as a benchmark set for β-sheet prediction methods. `efold` is not a method designed for SS predictions alone. However, BetaSheet916 represents a considerable corpus of proteins with low identity to validate the accuracy of `efold` through a large folding space. The current version of `efold` can predict the folding pathways of proteins with up to 200 AA in length, and explore β-sheet architectures composed of up to 6 different β-sheet strands. A total of 125 proteins were selected out of the 916 data set to be modeled by `efold`.

The full protein benchmark can be found in Appendix A.3.

4.1.6 Metrics to validate Experiments 1, 2 and 3 (folding pathways predictions)

Clear structural information on the intermediate states that bridge between the unfolded and native states is required to acquire a clear identification of protein pathways. Experimentally determining these intermediate structures in real proteins has proven to be exceptionally difficult. Traditional methods, such as crystallography and NMR, are not able to define partial structures that form and/or decay in short time rates (i.e., less than one second). Modern methods use kinetic measurements to quantify the folding process providing time-scale based views of the folding process. Hydrogen exchange methods study folding intermediates as significantly populated forms during kinetic folding, conformationally excited forms present at equilibrium under native conditions, or equilibrium molten globule forms [229]. Experimental methods based on spectroscopic (such as fluorescence, circular dichroism [CD] and infrared [IR]) follow kinetic folding in real time but are blind to the specifics of structure. The protein folding process can be monitored in CD spectroscopy through the average formation of SS of the protein.

Given the level of pathway information that can be extracted from `efold`, we validate the prediction of folding pathways (in **Experiment 1,2 and 3**) by comparing the order of SS formation with known experimental results and MD simulations. Given that `efold` stochastically extracts the predicted pathways through a Boltzmann sampling technique (see section 3.5.5), we average the pathways of one hundred independent runs to be used as our prediction (examples of results for single runs can be found in Appendix C). Figure 4.3 shows a comparison of SS formation orders for proteins G, N_UG_1, N_UG_2, Ubiquitin, SH3 and LSm predicted by `efold` with known experimental results. Figure 4.3 only reports the most populated (and probable) pathway predicted by `efold`. However, subsequent sections will also analyze less populated pathways predicted by `efold`.

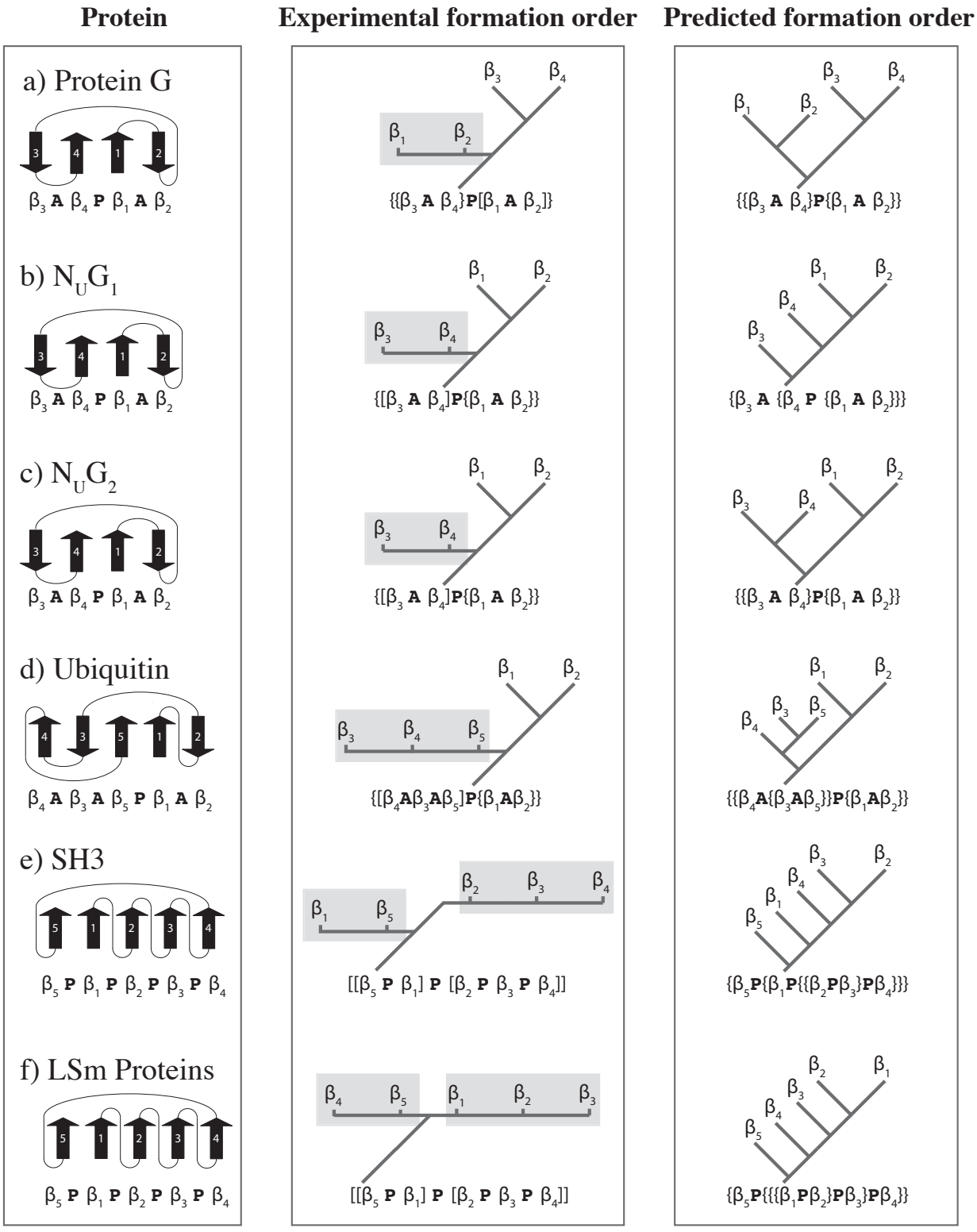

FIGURE 4.3: Comparison of SS formation orders predicted by `efold` with known experimental results.

SS formation orders for proteins a) Protein G, b) N_UG_1, c) N_UG_2, d) Ubiquitin, e) SH3 and f) LSm Proteins. Grey squares (and brackets) indicate no clear order. Only the most populated trajectories predicted by `efold` are shown. Each internal node in the tree represent a new pair of β interactions and nodes that are higher in the three indicate earlier interactions. The length (and angles) of branches do not have a specific meaning

4.1.7 Metrics to validate Experiment 4 (residue contact predictions)

Experiment 4 quantifies and evaluates the ability and capability of `efold` to predict protein topologies through the prediction of residue-residue contacts in the protein benchmark dataset of 125 proteins. The performance of `efold` is measured in terms of precision (i.e., $\frac{no.\ of\ correctly\ predicted\ contacts}{no.\ of\ predicted\ contacts}$), sensitivity (i.e., $\frac{no.\ of\ correctly\ predicted\ contacts}{no.\ of\ observed\ contacts}$) and the weight average of the precision and recall (called F-measure, i.e., $\frac{2 \times precision \times sensitivity}{precision + sensitivity}$). The number of *predicted contacts* is extracted from the stochastic contact map (that represents the probability of observing a given contact) predicted by `efold`. The number *of observed contacts* is extracted from the pdb file at three different levels: contact, strand and β/β-contacts. Residue contacts are defined as all pairs of AA that contains C_α atoms less than $8\mathring{A}$ apart in the pdb file. Strand contacts are residue contacts that are involved in SS. Finally, β/β-contacts are pairs of residue contacts that share an intra-backbone hydrogen bond as defined by the software `DSSP`. We classify the residue contacts (i, j) with their sequence separation $x = |i - j|$. Quality metrics are also estimated for short range ($x < 12$ i.e., less than 12 residues apart), medium range ($12 \leq x < 24$, between 12 and 23 residues apart), and long range contacts predictions ($x \geq 24$). Given that `efold` predicts distributions of structures rather than single structures, the evaluation is performed on exact and approximate contacts (i.e., a prediction ± 2 position apart of an observed contact) to take into account the neighbours of the true contact. In order to illustrate the quality of β-residue contact predictions achieved by `efold`, we compare its predictions with the ones obtained by state of the art methods.

In order to explore the impact of different sources of evolutionary sequence information, the proposed experimental framework was tested using as input the predictions computed by four different algorithms. `efold` was tested on secondary structure assignments (derived by the `DSSP` software), secondary structure predictions (derived by the `PSIPRED` software), residue contacts (derived by the `EVfold` software), and $\beta - \beta$ residue contacts (derived by the `bbcontacts` software) as sources of sequence information. In other words, the framework proposed in figure 4.1 (for experiment 4) was run for each

available evolutionary sequence information methodology (see section 3.2 for more details). It is important to stress that this analysis is performed to benchmark `efold` under different influences of evolutionary sequence information and is not meant to provide a performance-based ranking between the four algorithms. For simplicity, all results described in this chapter are shown for simulations that used `bbcontacts` software as input. However, the trends described also hold true for simulations using the other software. A complete set of results may be found in Appendix B.

4.2 Results

4.2.1 The predicted folding landscapes agree with pathways elucidated by experimental studies and / or MD simulations

4.2.1.1 Protein G

Figure 4.4a depicts the predicted folding transitions of Protein G. This figure reveals that the folding intermediates are consistent with previous literature reports [218, 219]. These results suggest an early formation of the second hairpin ($\beta 3 - turn - \beta 4$) and emphasize its fundamental role in the folding process. The second hairpin is known to be centered around nucleation points W43, Y50, and F54, where it is strongly stabilized by three hydrophobic residues W43, Y45, F52 [216]. Our results show a later formation of the first hairpin ($\beta 1 - turn - \beta 2$) in addition to the second hairpin. Overall, our simulations identified two folding pathways (blue and yellow pathways in figure 4.4a). These pathways passed through a $helix - hairpin2$ complex, a $helix - hairpin1$ complex and a $\beta 1 - \beta 4$ sheet complex. These two complexes are in agreement with previous studies suggesting that Protein G folds using three intermediates [216, 230].

The most probable pathway predicted by `efold` (see blue pathway in figure 4.4a) describes the folding process as the sequential assembly of elements from the super secondary structure. The first folding event is the formation of the ($\beta 3 - turn - \beta 4$) hairpin. The second event is the formation of the ($\beta 1 - turn - \beta 2$) hairpin, and, finally the nucleation

of the β-sheet residues between $\beta 1 - \beta 4$ completes the formation of the central part of the β-sheet completing the correct topology (See figures 4.3 and 4.4).

Figure 4.4b allows for a detailed understanding of the simulated folding dynamics of Protein G. It demonstrates how the probability of observing any of the reachable conformations changes over time. At the end of the folding process, the correct conformation topology $\beta 3\mathbf{A}\beta 4\mathbf{P}\beta 1\mathbf{A}\beta 2$ (see section 3.5.2 for a description of the used notation) emerges as the dominant topology in our simulations. The topology with the second best probability corresponds to $\beta 3\mathbf{A}\beta 4\mathbf{N}\beta 1\mathbf{A}\beta 2$, which is the topology obtained after creation of the first and second β-hairpins and just before the nucleation process to form the $\beta 1 - \beta 4$ assembly. The topology $\beta 3\mathbf{A}\beta 4$ presents the highest peak (around Log(Time) -1.5). This topology corresponds to the second hairpin, and has been reported to play a fundamental role during the folding [216]. The picture emerging from the dynamic simulations is in agreement with previous dynamics experiments and simulations of Protein G folding [215]. The most probable first folding event is the formation of the second β-hairpins, followed by $\beta 3\mathbf{A}\beta 4 \;\; \beta 1\mathbf{A}\beta 2$ (Log(Time) -0.5), and finally the nucleation of the β-sheets residues between $\beta 1$ and $\beta 4$ (Log(Time) 0), which results in the formation of the correct fold topology (i.e., $\beta 3\mathbf{A}\beta 4\mathbf{P}\beta 1\mathbf{A}\beta 2$).

4.2.1.2 Ubiquitin

Figure 4.5a shows the flow network prediction for the Ubiquitin protein resulting from `efold`. Our simulations are again in agreement with previous experimental results and in-silico simulations. In our simulations, the topology ($\beta 2\mathbf{A}\beta 1\mathbf{P}\beta 5\mathbf{A}\beta 3$) represents a fundamental transition state in folding trajectories because it is the state with the largest incoming and outgoing flows with direction to the native topology (i.e., $\beta 4\mathbf{A}\beta 3\mathbf{A}\beta 5\mathbf{P}\beta 1\mathbf{A}\beta 2$). The $\beta 2\mathbf{A}\beta 1\mathbf{P}\beta 5\mathbf{A}\beta 3$ sheet has been reported as a strongly organized transition state ensemble (TSE) on which Ubiquitin folds through [231]. The large incoming flow predicted by our simulations also agrees with the heterogeneity attributed for this transition state. It has been reported that this TSE may contain subpopulations with additional structure formation, where its structure can be spread in different directions (adding more

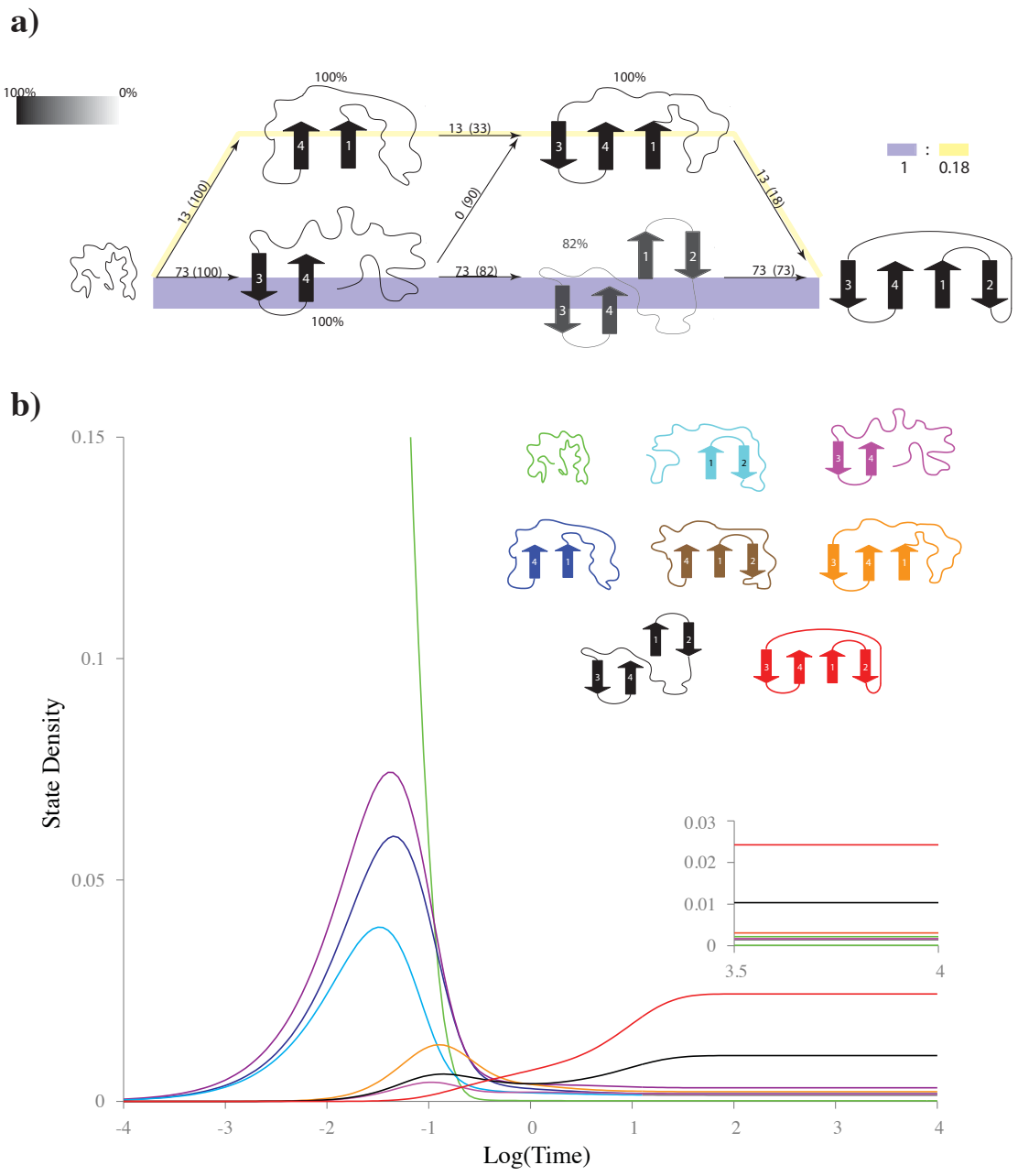

FIGURE 4.4: Predicted transition from a random coil to the native state of Protein G. **a)** Folding trajectories are represented as paths in a flow network of varyingly folded protein conformation states. Nodes represent energetically accessible conformation states. Edges represent the transition between two conformation states. Conformations connected by an edge are compatible topologies with structure similarity. Each node has its capacity and flow values labeled. The wide of the line represents how populated is a specific trajectory **b)** The predicted folding dynamics of Protein G, which shows how the probability of observing any of the reachable topologies changes over time the protein folds.

$\beta 1 - \beta 2$ or $\beta 3 - \beta 5$ structures). During the folding of Ubiquitin, the topology $\beta 1\mathbf{A}\beta 2$ acquires native-like conformations and is stabilized by tertiary interactions with the α-helix [232–234] (See figure 4.3). Next, strand $\beta 5$ and $\beta 3$ are joined to form the core structure [231, 235].

The dynamics of folding intermediates of Ubiquitin is shown in Figure 4.5b. Our simulations predict the correct fold topology with the highest probability at the end of the simulation. The precision of this topology for approximate contacts (i.e. ± 2 positions) averages around 58.81% and 69.53% for contact and inter-strand predictions, respectively. These results show that the predicted topology is an ensemble of conformations containing contact residues around the native conformation. The topology ($\beta 2\mathbf{A}\beta 1\mathbf{N}\beta 5\mathbf{A}\beta 3$) and ($\beta 2\mathbf{A}\beta 1\mathbf{P}\beta 5\mathbf{A}\beta 3$) have the second and third highest probabilities and they emphasize the importance of the four-stranded sheet complex during the folding process. The transition paths of Ubiquitin are also consistent with experimental and in-silico results [231, 235]. Our coarse grained simulations suggest the formation of $\beta 1\mathbf{A}\beta 2$ and $\beta 1\mathbf{P}\beta 5$ sheets during the first folding steps. Next, the topologies $\beta 2\mathbf{A}\beta 1\mathbf{P}\beta 5$, $\beta 3\mathbf{A}\beta 5\mathbf{P}\beta 1$ and $\beta 2\mathbf{A}\beta 1\mathbf{N}\beta 5\mathbf{A}\beta 3$ are formed. It is important to stress that all of these topologies have the central strand $\beta 5$ in common, which emphasizes the ability of the algorithm to identify the critical structural component through modelling non-local contacts. Once all elements are folded, the four-stranded sheet network ($\beta 2\mathbf{A}\beta 1\mathbf{P}\beta 5\mathbf{A}\beta 3$) starts to form around the Log(Time) -1. At Log(Time) 0, the native topology ($\beta 4\mathbf{A}\beta 3\mathbf{A}\beta 5\mathbf{P}\beta 1\mathbf{A}\beta 2$) emerges as the dominant conformation at the end of the folding process.

4.2.2 `efold` is able to predict the heterogeneity of folding pathways for proteins with high sequence identity.

4.2.2.1 Proteins G_B, N_UG_1 and N_UG_2

The predicted folding transitions for the G_B folds is shown in figure 4.6a. Focus is given to the G_B folds given that `efold` is an algorithm for modeling the folding process of large β-sheet proteins, such as the $\alpha + \beta$ fold reported by the G_B folds. The most populated

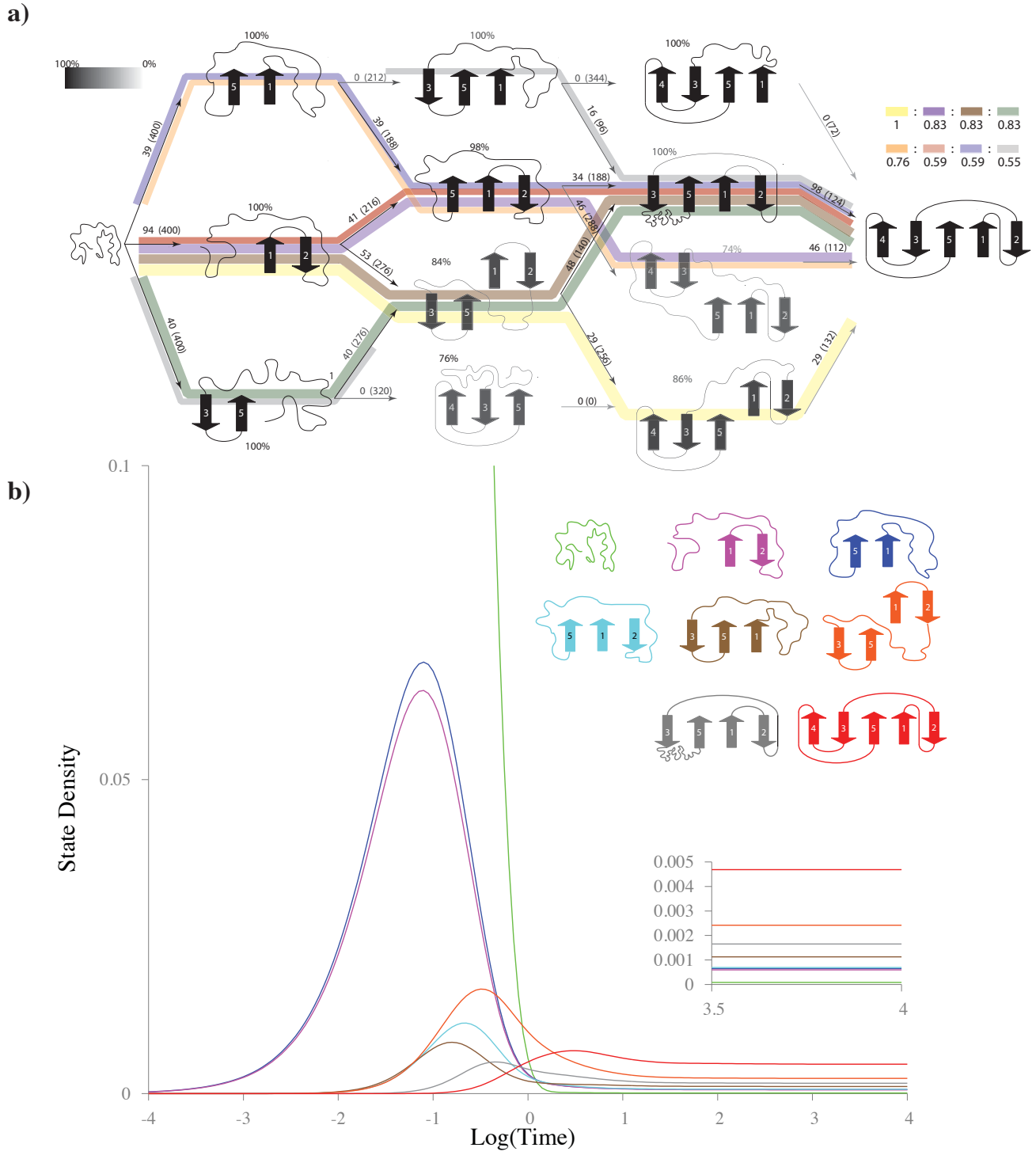

FIGURE 4.5: Predicted transition from a random coil to the native state of Ubiquitin. **a)** Folding trajectories are represented as paths in a flow network of varyingly folded protein conformation states. Nodes represent energetically accessible conformation states. Vertices represent the transition between two conformation states. Conformations connected by an edge are compatible topologies with structure similarity. Each node has its capacity and flow values labeled. **b)** The predicted folding dynamics of Ubiquitin, which shows how the probability of observing any of the reachable topologies changes over time the protein folds.

pathway predicted by our simulations is shown in figure 4.6. This path implies a first folding event corresponding to the creation of the $\beta 1 - \beta 4$ sheet followed by the formation of the second and first hairpin. This path is completely different to the paths predicted for the wild type of Protein G (See section 4.2.1). Then, our results proposes a different pathway route of the Protein G variants and our results agree with the evidence that Protein G variants exhibit distinct folding routes, where the main difference between them is the order of formation for the β-strands [236]. Analyzing the point mutations between the Protein G variants and the wildtype forms (see figure 4.2), it can be stated that the selection of the folding pathway is highly influenced by the mutations located between the start (E24) and end (K36) of the $\alpha-$helix. These mutations affect the folding of the helix and its early interaction with the second hairpin ($\beta 3 - turn - \beta 4$). In our simulations, the formation of the first hairpin ($\beta 1 - turn - \beta 2$) is delayed and it is only present at the end of the simulation. This results is in agreement with previous studies that establish that the second β-hairpin is stable in isolation and it can stabilize itself to some extent independently of the rest of the protein, whereas the first hairpin cannot [237].

Figure 4.6b shows the predicted folding transitions for the $N_U G_1$ variant. Compared with the trajectories predicted for the wild type version of Protein G, it is important to notice that the hairpin $\beta 1 - \beta 2$ plays an important folding role in $N_U G_1$. Furthermore, based on our simulations, the topology $\beta 1 - \beta 2$ is the topology with the maximum flow and it represents an early event during the folding process. This result is consistent with the experimental results in [213] and the predictions in [238, 239] (see figure 4.3). In our predictions, the hairpin $\beta 1 - \beta 2$ presents a parallel interaction with the strand $\beta 4$. Based on our predictions, this is the most populated route and it agrees with experimental results that suggest that in the folding process, the second hairpin (i.e., $\beta 3 - \beta 4$) disrupts at the rate limiting step in folding. Finally, there are other two less populated pathways (i.e., yellow and red pathways in figure 4.6b) in our predictions. These two pathways have not been previously found by experimental results, but they could be present given that the redesigned first hairpin of $N_U G_1$ is less rigid than the wild type and $N_U G_2$, counterpart [212].

Figure 4.6c shows the predicted folding transitions for the N_UG_2 variant. Similar to the N_UG_1 simulations, the hairpin $\beta1-\beta2$ is present as an important and early folding event. In our simulations, the hairpin $\beta1-\beta2$ is formed in the transition state of the folding pathway and serves as the starting point on which the rest of the protein can fold (see figure 4.3). This results agrees with experimental and in-silico results [213, 238, 239]. After the creation of the first hairpin, $\beta1-\beta2$ interacts with either $\beta1-\beta4$ or $\beta3-\beta4$. In the former's case, the hairpin $\beta1-\beta2$ is formed along the entire transition path, and is soon associated with $\beta4$, while $\beta3$ remains unfolded. In the latter's case, both hairpins are formed near the middle of the transition paths, but the level of hairpin-hairpin contact is lower (as shown by topology $\beta3A\beta4\text{N}\beta1A\beta2$). This order of structure formation has been observer previously by all-atom MD simulations in a triple mutant of N_UG_2 [240].

4.2.3 `efold` reveals the conservation of folding intermediates in evolutionary related proteins

4.2.3.1 PF00014 family

Figure 4.7a reports the predicted folding transitions for proteins belonging to Pfam PF00014. Proteins 1D0D, 1BUN, 1BIK and 5PTI were used in our simulations. This figure demonstrates that there are two pathways that are recurrent in all our simulations. From them, the pathway that conducts to the topology $\beta2\mathbf{A}\beta1\mathbf{A}\beta3$ is the most representative and it was found in 92% of our simulations. The topology $\beta1\mathbf{A}\beta2$ was traversed for all our simulations.

The folding pathways of disulphide proteins are known to vary substantially [187]. It has been shown that two of the structurally homologous kunitz-type protease inhibitors, bovine pancreatic trypsin inhibitor and tick anticoagulant peptide, are heterogeneitic in their folding intermediates and kinetics [134]. The simulated proteins in our benchmark represent three different kunitz-type protease inhibitors. Proteins 1D0D and 5PTI are bovine pancreatic trypsin inhibitors (BPTI), 1BUN is a serine protease inhibitor homolog

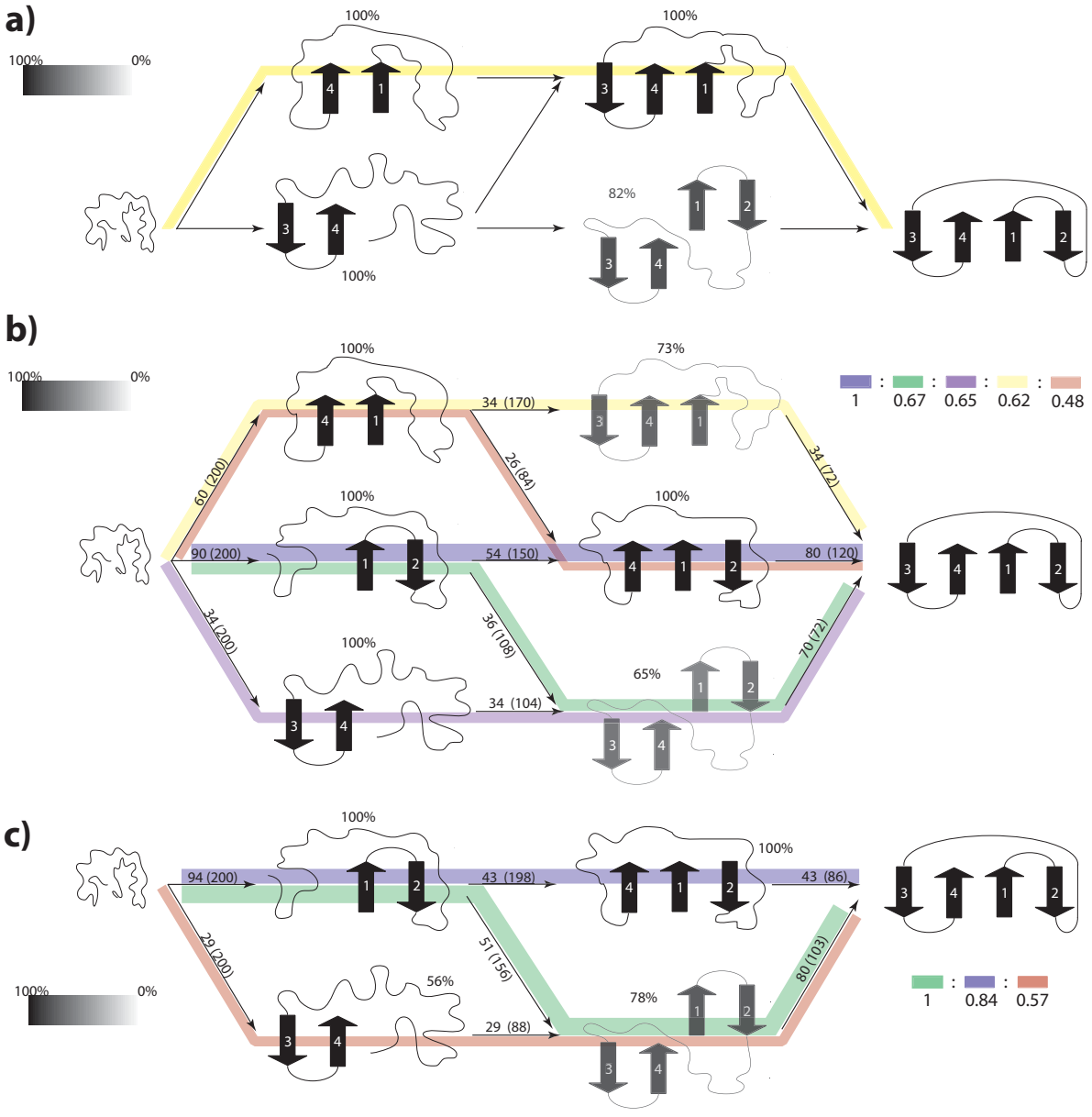

FIGURE 4.6: Predicted folding transition of the variants of protein G. **a)** Predicted folding trajectories for the G_B folds. **b)** Predicted folding trajectories for the N_UG_1 fold, and **c)** Predicted folding trajectories for the N_UG_2 fold.

beta-bungarotoxin B2 chain, and 1BIK is a Alpha-1-Microglobulin/Bikunin Precursor (AMBP) protein.

The BPTI simulated proteins traverse its folding pathway through the $\beta 2\mathbf{A}\beta 1\mathbf{A}\beta 3$ in 82% of our simulations. On the other hand, the topology $\beta 1\mathbf{A}\beta 3\mathbf{A}\beta 2$ was traversed in only 30%. Then, the BPTI simulated proteins show a dominant folding pathway with an early formation of their secondary structures. These results are consistent with experiment results that suggest BPTI as the stable subdomain structure dictating the formation of native-like intermediates and limiting the heterogeneity of folding intermediates [226]. The pathway traversing the topology $\beta 2\mathbf{A}\beta 1\mathbf{A}\beta 3$ is also the most probable for the AMBP protein (100%). However, topology $\beta 1\mathbf{A}\beta 3\mathbf{A}\beta 2$ is found in a majority of the simulations (90%) as well. The latter's pathway provides an interesting suggestion for an alternate folding route of disulphide proteins and could represent heterogeneitic folding intermediates for the PF00014 family.

4.2.3.2 SH3 - Domain

The predicted flow network for proteins with the SH3 domain are shown in figure 4.7b. Proteins 1OOT, 1I0C, 1NEG and 2HDA are used to simulate the transitions from a random coil to the native state. The four stranded $\beta 1\mathbf{P}\beta 2\mathbf{P}\beta 3\mathbf{P}\beta 4$ conformation (yellow path in figure 4.7b) is present in 81% of the simulations. SH3 domains were shown to agree with the formation of the $\beta 1\mathbf{P}\beta 2$, $\beta 2\mathbf{P}\beta 3$, and $\beta 3\mathbf{P}\beta 4$ motifs. Previous findings [241] suggested this topology as a metastable folding intermediate. The intermediate conformation has been shown to be highly aggregation prone, as it exposes strand $\beta 1$ [242]. Thus, the formation of the native strand $\beta 5$ (i.e., the last folding step in the path) is critical in preventing aggregates during folding.

The two predicted pathways by `efold` cross through the $\beta 2\mathbf{P}\beta 3\mathbf{P}\beta 4$ topology. This structure is common to all the domains simulated and represents a central (hydrophobic) sheet. Experimental results suggest that the second, third, and fourth β-strands are the most ordered regions of the TSE [222] (see figures 4.3 and 4.7). Transitioning from the $\beta 2\mathbf{P}\beta 3\mathbf{P}\beta 4$ topology, the folding trajectories branch into two distinct pathways (i.e., red

and yellow paths in figure 4.7b). The reported pathways complete their folding process with the formation of the second sheet (a less structured topology [223]) by the addition of two terminal strands $\beta 1$ and $\beta 5$. It is important to stress that this protein does not contain α-helix motifs, showing that `efold` is able to correctly model β-strands motifs in the absence of other secondary structures.

4.2.3.3 Ubiquitin-like - Domain

Figure 4.7c portrays the predicted folding transitions for proteins belonging to the Pfam family with index PF00240. Proteins 1CMX, 1EUV and 1UBQ were used in the simulations. The topology $\beta 2\mathbf{A}\beta 1\mathbf{P}\beta 4\mathbf{A}\beta 3$ was shown to be the most conserved topology for this family. This topology corresponds to the organized TSE that consist of a four-stranded sheet network (i.e,. $\beta 2\mathbf{A}\beta 1\mathbf{P}\beta 5\mathbf{A}\beta 3$) in the folding pathway of the Ubiquitin protein (see section 4.2.1.2). Contrary to the simulations targeting the ubiquitin protein, simulations for the PF00240 do not show $\beta 3\mathbf{P}\beta 1\mathbf{P}\beta 2$ as a populated topology. The two most populated paths (blue and yellow in figure 4.7c) cross through topologies $\beta 3\mathbf{A}\beta 4$, showing concordance with the Ubiquitin folding suggested from previous experiments (see section 4.2.1.2). The strand $\beta 4$ is present in all paths that transition to the final topology. $\beta 4$ is the central strand and a critical structural component in the Ubiquitin topology [232].

The $\beta 2\mathbf{A}\beta 1\mathbf{P}\beta 4\mathbf{A}\beta 3$ topology is topographically related to Protein G because the order, positions, and stretches of their secondary structures are identical. This similarity has been shown to be present in the Ubiquitin family and in other proteins with biologically distinct functions, such as the (Ig)-binding protein G [243, 244]. This common fold is termed the β-grasp and has been suggested to be a multi-functional scaffold in diverse biological contexts [245]. Studies have proposed that these proteins belong to the same superfamily (i.e., identical folding, but highly diverged sequences) and retain the identity of sequence positions that participate in the folding nucleus [246]. Monte Carlo simulations have previously identified nucleus positions that are conserved among structures with homologous folds for Protein G and Ubiquitin [230]. Shimada et.al.(2002) identified nucleus residues in hairpin 1 (Y3 and L5), the helix (F30), and the hairpin 2 (W43, Y45,

and F52). All of these residues show a low sequence entropy over aligned sequences in the Ubiquitin superfamily [246]. With respect to our simulations, these residues were reported to participate in most of the predicted pathways, which highlights the importance that `efold` confers to these residues. The aligned sequences in the Ubiquitin superfamily (with respect to the protein G) report that residues L5:K6, F30:V26, W43:L43, and Y45:F45 have a presence of 99.52%, 82.54%, 85.37% and 42.45% in the reported pathways, respectively.

4.2.3.4 PF01423 family

The predicted folding transitions for proteins belonging to the Pfam PF01423 are shown in figure 4.7d. Proteins 1KQ1, 1HK9 and 1H64 were used in the simulations. From this figure, it is clear that `efold` predicts only one populated path that corresponds to the early formation of β1, β2 and β3 strands, followed by the formation of the β4 and β5 strands. These two sets of strands correspond with two sequence motifs (32 and 14 AA long, respectively) that have been identified between various LSm homologs [227] (See figure 4.3).

4.2.4 `efold` on average has greater precision than state of the art contact residue algorithms for proteins without homology-based templates.

4.2.4.1 Contact prediction

Figure 4.8 (column 'Contact Prediction') reports the results obtained by `efold` on 125 proteins extracted from the BetaSheet916 benchmark data set (see section 4.1.5 for further details regarding the benchmark). The proposed method predicts residue contacts with a high precision for ± 2 for all contact separations in the complete benchmark (the precision value averages around 85%). The precision of exact prediction (i.e., columns ± 0) is also high and it averages around 46% for the short and medium contacts and around 33% for

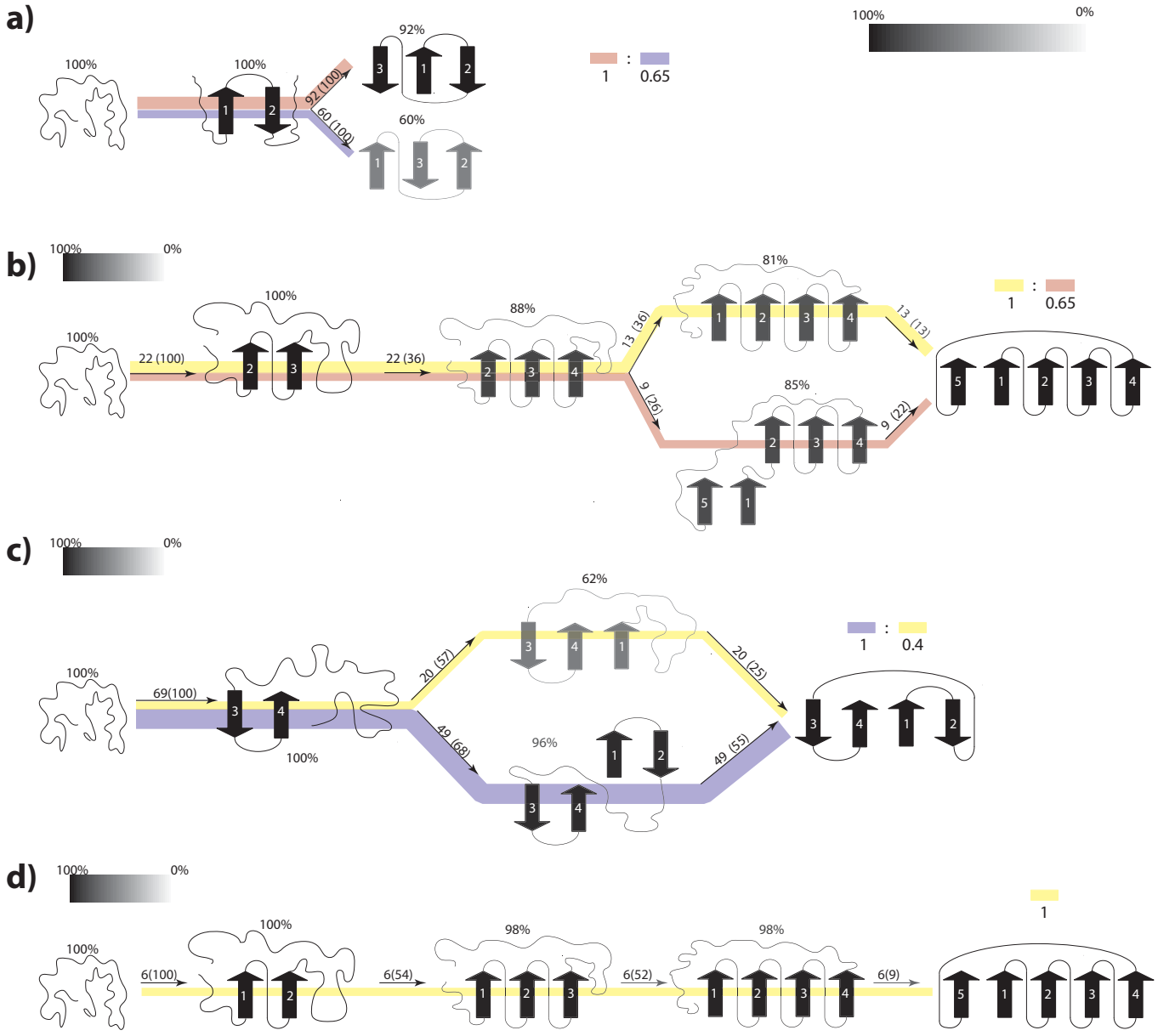

FIGURE 4.7: Predicted transition from a random coil to the native state of the Pfam families: a) PF00014, b) PF00018, c)PF00240, d)PF01423.
Folding trajectories are represented as paths in a flow network of varyingly folded protein conformation states. Nodes represent energetically accessible conformation states. Vertices represent the transition between two conformation states. Conformations connected by an edge are compatible topologies with structure similarity. Each node has its capacity and flow values labeled.

long-range contacts. This result is significant because the precision achieved by state-of-the-art residue contact prediction algorithms without homology-based templates averages around 20% [113]. The difference of performance between the experiments for exact and approximated predicted contacts can be understood by the fact that `efold` predicts distributions of structures rather than single structures as measured by the precision metrics. The sensitivity obtained by `efold` averages around 15% and 35% for exact contacts and approximate contacts within ±2 positions, respectively. The performance measured by the precision versus the sensitivity metrics can be understood by the fact that `efold` aims to predict a subset of contacts between residues within SS, while the benchmark includes all possible types of contacts. The number of false negative values produced by `efold` is high compared with the number of false positives.

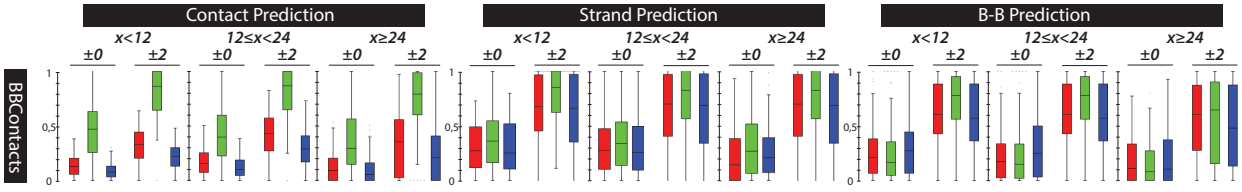

FIGURE 4.8: Contact, Strand and β/β predictions performed by `efold` for the complete protein benchmark.
Performance is evaluated on precision (green), sensitivity (blue) and F-measure (red) of experimentally observed contacts. These metrics are reported for contacts which are 0, 12 and 24 residues apart, and when predicted contacts are within ±2 residues of an observed contact.

4.2.4.2 Strand prediction

Given that `efold` is focused on the prediction of residue contacts involved in $\beta-$strands, we also calculate the power of `efold` at predicting inter-strand and β/β residue contacts. The second column in figure 4.8 reports the results of each experiment for contact prediction of residue-residue contacts involved in β-sheet structures. `efold` achieves an excellent F-measure above 70% for ±2 of all contact separations in the complete benchmark. Typically, the F-measure for experiment performed using SS averages around 90%. The average of `efold` performance metrics are similar to the ones obtained for any type of short, medium, or long-range contacts in this experiment. As expected, the performance metrics reported for `efold` are more homogeneous than those obtained in the

unrestrained contact prediction case. The results reported by figure 4.8 confirm the good performance of `efold` at predicting residue-residue contacts and incidentally β-sheets. In average, `efold` performs better in predicting inter-strand residues than contact residues.

4.2.4.3 $\beta - \beta$ Contact residues

In order to obtain a solid basis for comparison and to be able to compare `efold` results with other general contact predictors, a residue level comparison was performed. The third column in figure 4.8 reports the results of our experiments regarding pairs of residues involved in β/β contacts. `efold` achieves F-measures above 63% for ± 2. The results for long-range contacts are more spread out (i.e., a bigger $25 - 75$ interquartile range) than short and medium-range contacts. Figure 4.9 reports a comparison of residue-level performance on the BetaSheet916 benchmark dataset of previous contact and β/β contact prediction methods, respectively. Analyzing these figures, it can be noted that `efold` exhibits better results than those obtained by contact prediction methods. However, `efold` is outperformed by β/β contact prediction methods. These results are explained for the following three reasons: *i*) `efold` predicts distributions of structures rather than single structures as measured by the compared metrics. `efold` performance is highly penalized by the increase of a false positive prediction that corresponds to the neighbourhood of the true residue contact. When analyzing an ensemble of predictions rather than a single prediction, it is observed that `efold` outperforms β/β contact prediction methods (see column ± 2 in figure 4.8 and the label '`efold` (± 2)' in figure 4.9). *ii*) `efold` was not built to detect β-bridges, these structures represent 17% of the $\beta-$strand pairs in the BetaSheet916 data set. Most of the state of the art methods (e.g., CMM[247], PhyCMAP [248], MLN and MLN-2S [249], BetaPro [228], BCov [250]) code information about β-bridges during the prediction. *iii*) The state of the art methods use `DSSP` assignments for evaluation (with the exception of `bbcontacts`). `DSSP` assignments use known secondary structures, which increase prediction power. When `DSSP` assignments are compared for all the methods (see label '`efold` (`DSSP`)' in figure 4.9), we observe that `efold` compares favourably for precision/recall (i.e., in the same range averaging around 40%) compared to

β/β contact prediction methods (with the exception of `bbcontacts`, which outperforms all other tested methods).

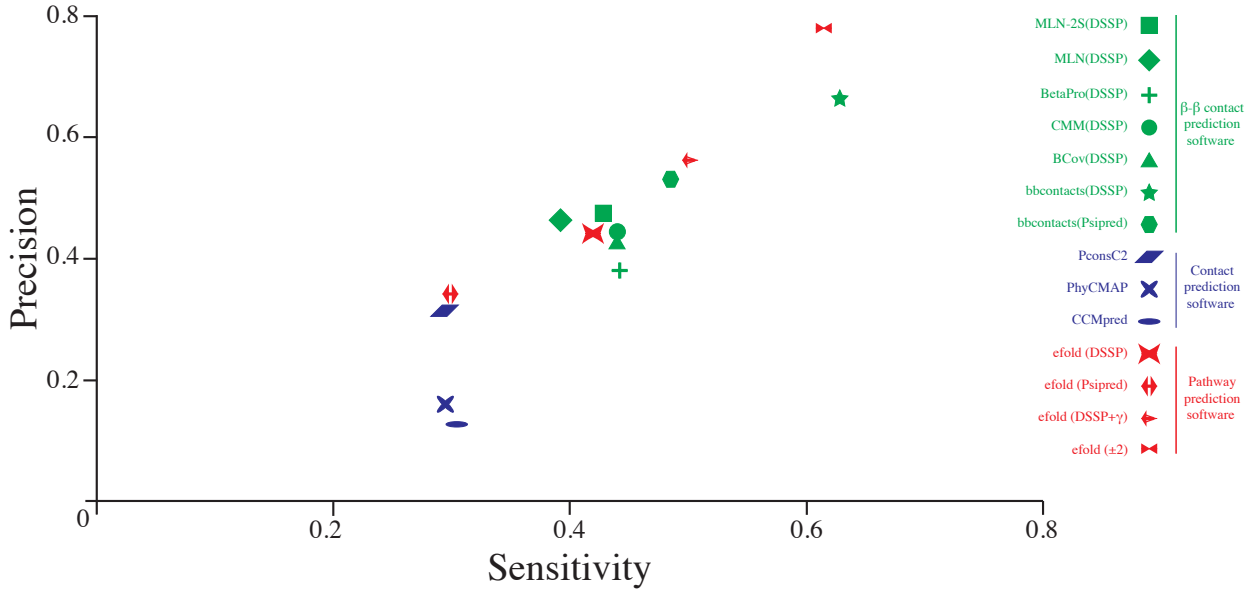

FIGURE 4.9: Performance of `efold` compared to state of the art software
Performance of `efold` compared to previous β/β (CMM[247], MLN and MLN-2S [249], BetaPro [228], BCov [250], `bbcontacts` [183]) and residue contact prediction (PhyCMAP [248], CCMpred [251] and PconsC2 [252]) methods on the BetaSheet916 dataset. This figure is based on data reported in [183].

4.2.4.4 Further analysis of `efold`

From the previous analysis, it is clear that `efold` is able to compute excellent predictions in terms of protein residue contacts. Moreover, `efold` is able to improve its performance up to an additional 22% if the right prediction decisions regarding the contributions of its different parameters are made. Figure 4.9 shows, that based on perfect decisions, `efold` clearly outperforms the results obtained by contact prediction methods and shows improved prediction over β/β contact prediction methods. Hyperparameter optimization was not performed because `efold` is not a contact residue predictor, instead `efold` is a protein folding pathways predictor. Our main interest is not increasing the quantity of correctly predicted residues, but increasing the quality of predictions. We performed a large-scale validation of `efold` structure prediction capacities as a milestone to predicting folding pathways.

Given that `EVfold` and `bbcontacts` perform predictions of residue contacts, we can extend the analysis of the `efold`'s results to understanding the varying performance between methods. Figure 4.10 answers the following questions: *i*) What percentage of true `efold` predictions were not correctly predicted by `bbcontacts/EVfold`? *ii*) What percentage of true `bbcontacts/EVfold` predictions were not correctly predicted by `efold`? *iii*) What percentage of wrong `bbcontacts/EVfold` predictions were correctly discarded by `efold`?. From figure 4.10, it can be understood that the number of `efold` predictions that were not discovered by the other two methods is very high when the contact and strand level metrics are used. With respect to $\beta - \beta$ predictions, `efold` outperforms the other two methods in proteins with low number of detected homologous sequences. Regarding question ii), as expected, the number of true contact predictions that were not predicted by `efold` is high when compared with the predictions of `EVfold`. This difference is due to the fact that `efold` aims to predict β/β pairs and ignores most residue contacts not involved in SS. This conclusion can be verified when the strand and residue levels are analyzed. In those levels, the average of missed residue contacts notably decreased. Finally, it is important to note that `efold` was able to discard most erroneous predictions that entered ensemble modeling as inputs. This result is shared by all the performance measure metrics (i.e., contact, strand and residue level) for both programs (i.e., `EVfold` and `bbcontacts`).

The schema followed by `efold` during the modeling of ensembles focuses on β-strands folds (see section 3.5). However, `efold` is able to include α-helices in the modeling thorough the incorporation of evolutionary sequence information. An α-helix prediction can be achieved from direct SS or 2D contact map predictions. These predictions are generally very accurate (and dramatically more successful than β-sheet predictions [253]) given that they are comprised by local patterns that can be quite readily detected. The current accuracy of state of the art methods to predict helices averages around 86% [254]. Studies regarding the misclassification rates between different SS types establish that protein SS predictions are more probable to wobble between $\beta-$strands - coil and α-helix - coil [255, 256]. In other words, one can expect that the misclassification of helical as sheet (or viceversa) residues is lower than the misclassification involving coils [257]. These expectations are also shared by the benchmark used by `efold`, where the

accuracies for α-helix prediction of `PSIPRED` and `bbcontacts` are 73.23% and 82.84%, respectively. The misclassifications of true α-helices as β-strands correspond to 1.56% and 3.82% for the `PSIPRED` and `bbcontacts` methods, respectively. The inverse analysis (i.e., misclassification of true β-strands as α-helices) shows also very low values where `PSIPRED` and `bbcontacts` mistakenly classified only 1.12% and 0.54% of the true β-strand residues, respectively. On the other hand, the misclassification of true α-helices as coils is higher and it corresponds to 13.55% and 13.33% for the `PSIPRED` and `bbcontacts` methods, respectively. Figure 4.10 shows that `efold` was able to discard most erroneous predictions that entered ensemble modeling as inputs. Then, the previous results support the idea that `efold` can assume helix predictions as accurate (for its purposes). Furthermore, `efold` managed (and corrected) the misclassification errors during the ensemble modeling. Particularly, the noise added to our energy function (i.e., misclassification between α-helices and β-strands) is very low, and the missing information (i.e., misclassification between α-helices [or β-strands] and coils) is complemented by `efold` during the ensemble modeling process.

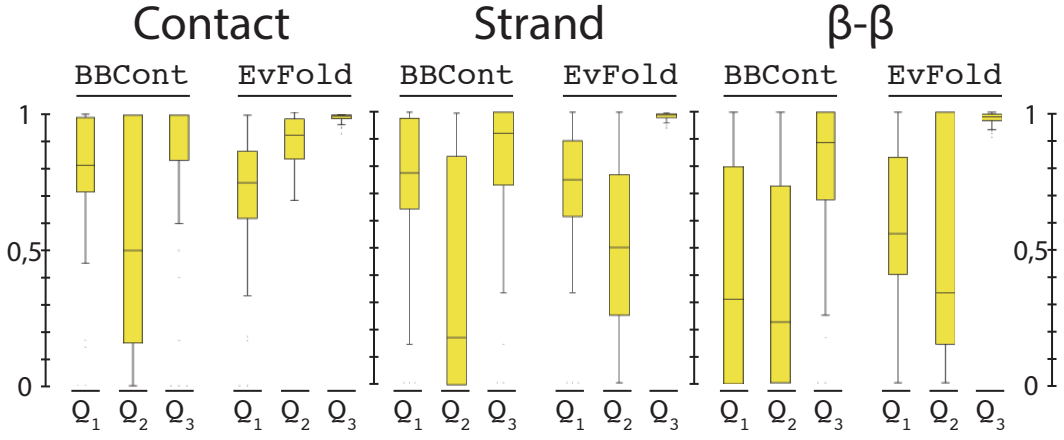

FIGURE 4.10: Performance of `efold` compared with the input predictions Q_1 = Percentage of true `efold` predictions that are not correctly predicted by `bbcontacts`/`EVfold` Q_2 = Percentage of true `bbcontacts`/`EVfold` predictions that are not correctly predicted by `efold` and Q_3 = Percentage of wrong `bbcontacts`/`EVfold` predictions correctly discarded by `efold`.

To validate the energy consistency within the clusters of protein conformations created by `efold`, the silhouette metric was computed. Particularly, a silhouette analysis is used to study the energy separation distance between clusters of protein conformations with compatible topologies and structural similarity (i.e., nodes connected by an edge in the

graph that represents varyingly folded protein conformation states). In our study, the silhouette analysis relates dissimilarities of energies of conformations of different clusters to dissimilarities of energies of the same cluster (see equation 4.1). The silhouette metric ranges from -1 to 1, where a value close to one indicates an excellent match between the conformations that belong to the same cluster and a very poorly match with respect to conformations belonging to neighbouring clusters. In other words, a high value indicates that the clustering configuration protocol is appropriate. Negative values in the silhouette metric indicates that the conformations have been assigned to the wrong clusters because the intra-clustering similarity is in average lower than the inter-clustering differences. Finally, a silhouette value near zero means that in average the conformations are on the border of two or more natural clusters indicating the presence of overlapping clusters. It is important to stress that the clustering procedure performed by `efold` is based on structural similarities of samples (see section for further information). However, the silhouette analysis is performed taking the energetic values of the protein conformations as reference. We are interested in studying the consistency of the clusters based on their energy because the rates of interconversion between proteins states during the folding dynamics simulations are estimated from the gradient $\Delta G_{i,j}$ of energy differences between two macro states (see equations 3.14 and 3.15). Having consistent energy clusters of structural similar protein conformations is important to guarantee a logic study of protein folding dynamics via the master equation method used by `efold`. The following results are computed using the full set of proteins tested in this thesis (i.e.,125 proteins).

$$
\begin{aligned}
S = \frac{1}{n}\sum_{i} s_i = \frac{1}{n} \times \frac{b(i) - a(i)}{max[a(i), b(i)]} \\
a(i) = \frac{1}{|c_i|}\sum_{j \in c_i} d(i,j) \\
b(i) = \frac{1}{n - |c_i|}\sum_{j \in C\ c_i} d(i,j)
\end{aligned}
\tag{4.1}
$$

The (average) silhouette value measured based on the potential energy of its protein conformations (see equation 3.3) is 0.7835. The standard deviation of this measure is equal to

0.083. On the other hand, the (average) silhouette value using `efold`'s objective function (see equation 3.9) as distance is 0.8513. The standard deviation for this value is equal to 0.127. These values show that the compositions of the clusters computed by `efold` are consistent and they are suitable to model the folding dynamics of the system. The good consistency of `efold`'s clusters is supported by the mechanism used to construct the clusters. Particularly, clusters joined by an edge in the graph are identical to each other, modulo the addition or removal of a single β-strand pairing. Figure 4.11 shows the histogram of the inter- and intra- ΔG energy distributions from the centroids of the clusters. This plot shows a clear separation between the intra and inter histograms with a small overlap area. Then, it is possible to state that the conceptual framework used by `efold` for analyzing the dynamics of folding process is convenient because the modelled transitions allowed secondary structural changes (addition or deletion of a β-strand) favoring the formation of stabilizing interactions i.e., hydrogen bonds of SS. `efold` is able to accurately guide the folding process by correctly coupling the formation of SS during folding. In other words, the consistency of `efold`'s clusters allows for the characterization of low energy pathways that have been determined by the stepwise addition of SS.

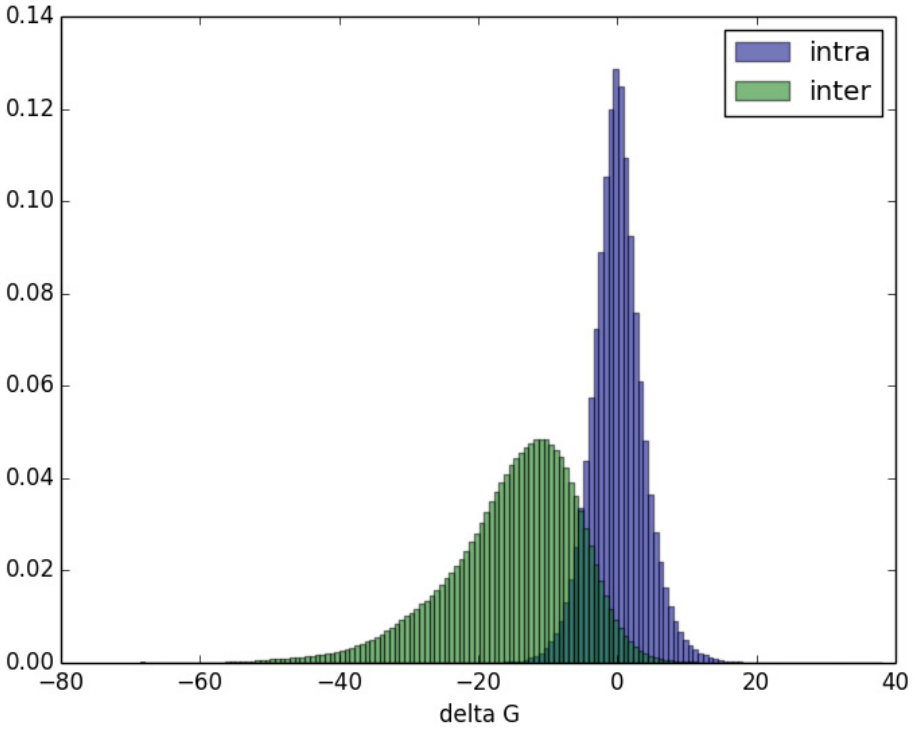

FIGURE 4.11: Intra and Inter ΔG distributions
The histogram of the Intra and Inter ΔG distributions calculated from the centroids of the clusters

Chapter 5

Visualizer of Protein Folding Pathways

5.1 Introduction

Statistical ensemble methods replace the idea of a single folding pathway by considering a multitude of folding routes that traverse a multidimensional energy landscape of a folding funnel model [258]. The intermediate conformations are therefore not considered discrete states, but an ensemble of structures, where the transition between consecutive ensembles on the folding pathway happens by parallel routes [186]. Simulating protein folding through statistical ensemble methods is also a means to gain new insight into experimental problems compared with classic methods. Ideally, these simulations shed new insight into how proteins fold, unravel and suggest hypotheses that are hard to explore experimentally, construct more focused experimental designs, and suggest new interpretation of experiments.

Despite its relative young age, ensemble statistical mechanic methods have already begun to influence our view of protein folding. Applying ensemble-based approaches to folding prediction requires a visually appealing information system that allows users to easily interact with predictions. Gaining the wealth of information contained in folding pathways will also advance our understanding of protein folding mechanisms and its implication in

human health and disease. However, this advance would never be realized without the appropriate visualization techniques to interact with protein pathways. `efold` represents an alternative to PF high computational cost approaches and allows for the inclusion of pathway prediction during the study of the folding process. These features of `efold` may cross traditional borders of structure prediction software, but we still require a visual tool that allows for the easy integration of structure and pathways information in order to obtain meaningful biological inferences. In particular, we believe that `efold` will allow for large scale studies of folding dynamics annotations in proteomes.

Protein structures and pathways derived from large-scale protein predictions significantly increase the annotation coverage of proteomes. The most critical limitation of structure prediction pipelines is the computational and time costs associated with the prediction, storage, analysis, data curation, interaction and visualization of predictions; which makes full proteome analysis impossible [259]. This computational cost is related not only to the 'high-performance' computing infrastructure (i.e., hardware in supercomputers, grids, or dedicated machines) needed to calculate the predictions, but also with the 'average user' hardware (i.e., available network bandwidth and the memory of commercial computers, tablets or mobiles) and the time of manual curation and analysis (i.e., by humans) that are essential for the interactive visualization of large and complex predictions.

There are several internet services that are available to predict and visualize protein structures and functions (see [260] for an up-to-date list of PSP methods, see [261–267] for examples of presentation and manipulation of protein structure data, and see [268, 269] for examples of approaches to predict protein function from structure). Recently, novel graphical interfaces that present functionalities to interact with multiple protein structure predictions have been developed (see [270, 271], [272] and [273] for interactive platforms for PSP, co-evolution analysis and protein assembly, respectively). A few graphical interfaces and servers have also been created to perform web-based analyses of protein folding dynamics [274]. For example, servers to calculate the folding nucleus for proteins [275], folding rates [276], discrete molecular dynamics [277], analysis of specific pathways [278, 279] and unfolding pathways [93] have been reported in the literature. However,

options to access, visualize, and interact with outputs (representing networks of folding/unfolding pathways) remain very limited. We believe there is a lack of online servers that allow for the prediction and interaction of protein pathways. This lack of appropriate tools, poor data accessibility, and insufficient benchmark data are major impediments for the PPP in PF. Structure visualization of macromolecules techniques may be a good starting point to develop novel protein pathway visualizers. These techniques have been fundamental for the study of 3D protein structures and understanding of biological processes [280]. However, we need complementary visualizing tools in the field because the number, size, and complexity of macromolecular structures are increasing dramatically and large-scale structural analyses have become a Big Data challenge[1]. The ability to efficiently analyze, store and interact with the growing flood of protein structure data is fundamental to allow new insights in protein folding mechanisms [281].

The few pathway prediction algorithms present in the literature [93, 278, 279], are built on complex implementations that do not allow for the creation of easy-to-use interactive networks of folding pathways. Attempts by structural biologists or experimentalists to generate and analyze pathways predictions, before beginning time-consuming experiments and simulations, are often hampered by the lack of a system to generate, manage, store, and study predictions. The potential users are faced with the daunting task of generating, evaluating, and validating predicted pathways. Currently, there is a tangible need in structural biology research to develop a pipeline that can present data to the user community in both a human (to support single experimental designs such as those in early stages of protein engineering and drug discovery processes [282]) and machine (to support large-scale protein simulations such as the study of kinetic stability of proteomes [283]) readable manner.

`efold` predicts folding energy landscapes in the form of graphs. These graphs contain thousands of topologies (vertices in a graph) with many relations between them (edges in a graph). We believe the raw data will be useless if protein topologies cannot be properly analyzed, annotated, stored, and displayed. Therefore we have created a landscape visualizer to present data to the wider biology and computer science communities. We

[1]The term Big Data challenge makes reference to the increasing amount of complex biological data produced thanks to improvements in biomedical tools and technologies

developed a tool to convey the content of the landscape graph through an interactive exploration of networked data. This tool aims to provide the means to disseminate data created by `efold` and comprehend their biological importance. This tool is available at http://cs.mcgill.ca/∼dbecer/efold/.

Our system uses graph theory (e.g., graph traversal, shortest path and flow networks algorithms) in order to search for properties, inspecting nodes and edges, and exploring their relationships within a web browser. We decided to display predicted pathways on the web because they will be more accessible to scientists and non-experts without the need of dedicated hardware or software. Additionally, a web-based solution avoids the need for custom software installation, which could represent a non-trivial barrier to the adoption of software applications [280]. The proposed system uses JavaScript as the main programming language for pathway visualization and analysis. We expect this pathway visualizer will enable users to analyze large and different protein pathways, carry out large-scale simulations, and visualize the results in an intuitive manner. The next sections will briefly describe the main features of the proposed visualizer in analyzing protein structure (see section 5.1.1), dynamics (see section 5.1.2) and pathway (see section 5.1.2.1) ensemble landscapes.

5.1.1 A visualizer of conformational landscapes

`efold` predicts folding transitions from a random coil to the native state as a path through varyingly folded protein conformation states. This transition through conformation states is represented as a graph, where vertices are the most energetically accessible and favourable conformation states for a given topology (previously generated by the Boltzmann ensemble sampling method). Each vertex represents an ensemble of similar conformations states that have been grouped together using a hierarchical clustering algorithm (see section 4.2.4.4). Each conformation state represents protein SS elements through a coarse-grained representation. Graph edges represent the transition between two conformation states, where conformations connected by an edge are compatible topologies with similar structure. `efold` enumerates and ranks all β-sheet topologies

to build a coarse-grained energy landscape (see section 3.5 for a complete description of the methodology).

Figure 5.1 shows an example of the conformational energy landscape predicted by `efold`. The visualizer displays a graph (network) of transition conformational states. Each state is represented by a circle of variant size and color. The size of each node represents its associated occurrence likelihood, where a bigger circle (higher likelihood) indicates a more energetically accessible set (ensemble) of structurally related low-energy conformations. The color of each node shows the number of β-strands in the conformational ensemble. There are two display modes in the visualizer to show the conformational landscape, where the user selects a 'network' or 'ring' view of the landscape. The 'network' mode represent the standard interaction network visualization in which entities are displayed as round nodes and edges represent the relationships between them. In the 'ring' mode, the localization of the nodes is not arbitrary and nodes are localized in the perimeter of a circle (i.e., ring) in which the radius depends on the number of β-strands. In the 'ring' mode, there are as many rings as number of $\beta-$strands and each node will be located in the perimeter of the circle that match the number of β-strands of the topology represented by the node. These two displays of network nodes can highlight non-trivial data relations that may otherwise be overlooked.

In figure 5.1, each node in the graph represents an ensemble of protein SS conformations. If the user selects one circle, all the information regarding the current ensemble is displayed. This metadata is essential to the user for understanding the relationships between protein conformations and the network structure. The metadata is showed aside the network to preserve the visual simplicity of the graph. This simplicity is important because it allows users to visually correlate entire populations of topologies based on how they relate to one another. A box containing information about topological labeling (see section 3.5.2 for details of the labeling) of the ensemble is depicted just beside the clicked node. Labeling is accompanied by a graphical representation of the SS conformation, where an array of directional arrows (each arrow representing a β-strand) is used to indicate the topology of the protein ensemble. The arrows are colored using a 'blue to purple' color scale that represents the position of each β-strand with respect to the AA sequence.

In the top-left part of the visualizer, a box showing additional protein information is also depicted. Particularly, this box shows information about the shape (i.e., topological labeling), ID (unique identifier of the node), probability (i.e., occurrence likelihood) and residues (extracted from one example of the ensemble) involved in β-strands. Finally, a pathway connecting the unfolded state, intermediate states and the selected state (i.e., the node clicked by the user) is depicted. The visualizer will list all paths that connect the unfolded state with the selected node sorted by the sum of the current likelihoods of nodes that make up the path.

5.1.2 A visualizer of protein dynamics

`efold` predicts the folding dynamic of a protein system as motions on the coarse-grained conformational energy landscape (i.e., the coarse-grained energy landscape already modeled in section 5.1.1). This landscape is suitable to model protein dynamics because it provides a structural definition of key conformational states of a protein (i.e., nodes in a graph of varyingly folded protein conformation states), the mechanism of protein conformational change (i.e., addition of single strands), and the height of barriers connecting these key conformations (i.e., the difference between the free energies of the states) [121, 195–198]. `efold` defines the time-evolution of the protein folding process by considering the protein system as being in exactly one of many countable number of states at any given time, and where switching between states is treated probabilistically. `efold` models the time-evolution by a transition matrix using the symmetric Kawasaki rule (see section 3.6 for more details). The time units used by `efold` are not related to real time. Ideally, model time should be proportional to physical time, but this is unlikely for models following the master equation formalism (such as `efold`) [284].

In our visualizer, the user may access the folding dynamic results by clicking 'Display Pathway' (see figure 5.1). The user will find a subgraph of the complete graph (i.e., the conformational landscape graph) composed by all the conformational states for which a folding trajectory (i.e., a path connecting the unfolded and the 'user selected' conformation) intersects. Given that the size (i.e., number of nodes and edges) of a subgraph is less

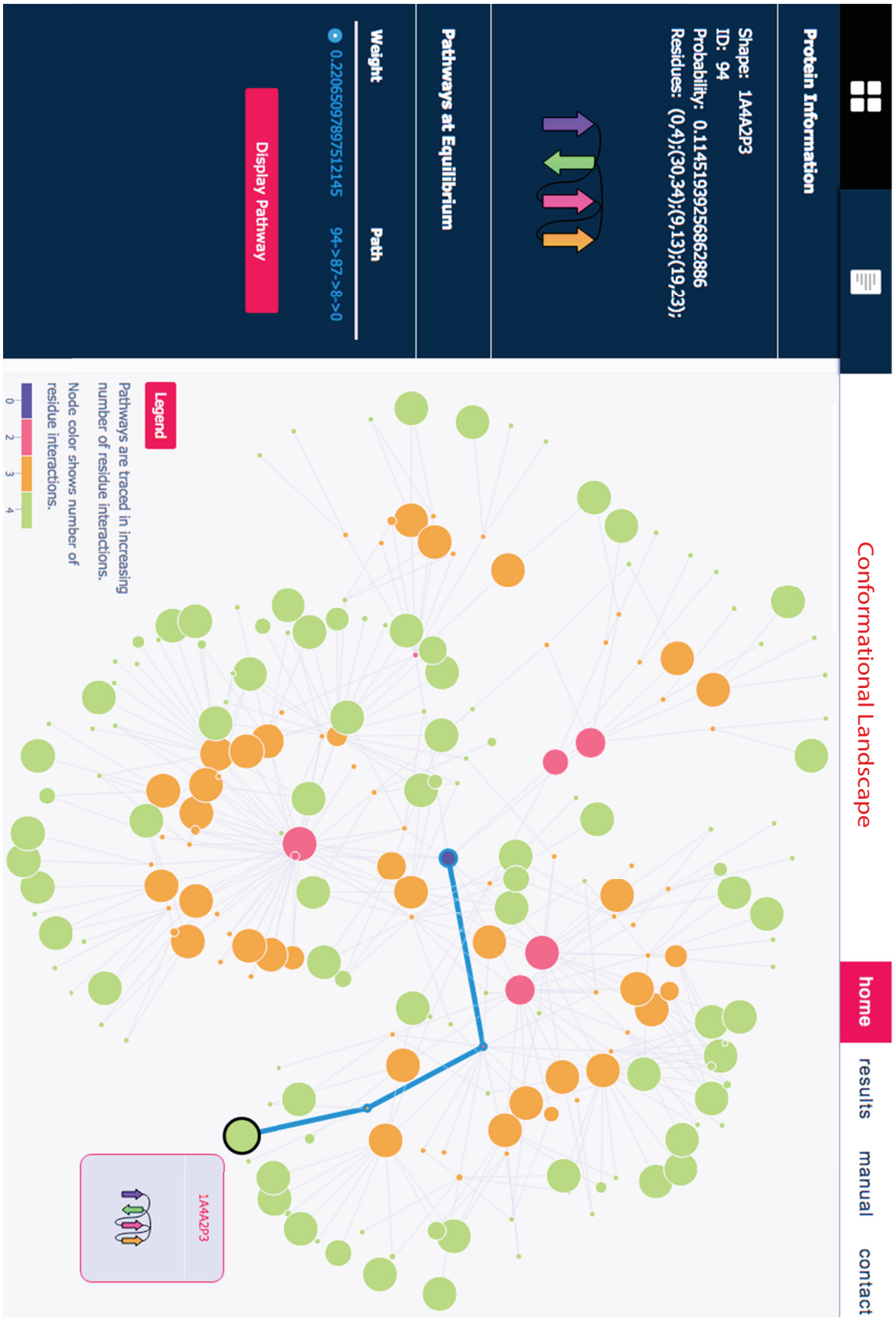

FIGURE 5.1: A visualizer to model protein structures
A screenshot of the visualizer to model the structure landscape predicted by `efold`

than the one of the original graph, nodes are represented by a graphical representation of the topology of the protein ensemble (see figure 5.2). The user may see the predicted folding dynamics as a movie of the folding time evolution for the selected protein by selecting 'Dynamics Movie'.

Figure 5.2 shows an example of how the probability of observing any of the reachable topologies in a protein changes over time that a protein folds. The time has been discretized by default in 241 different time-steps[2] (from time -6 to $+6$ with step variations of 0.05), and the occurrence likelihood has been computed for each conformational state. The weight of a connection between two conformational states i and j (i.e., an edge in the graph) represents the amount of flow that pass through this specific edge. The weights for each time step ℓ are computed by equation 5.1. In this equation $P^j_{t_\ell}$ is the occurrence likelihood of the conformational state j in time ℓ as computed by equation 3.13. Q is the finite set of possible energy states and $Q_{i,j}$ is the subset of nodes that have an incident edge with j. Equation 5.1 can be understood as the normalization of the transition probabilities (represented by equation 3.16) with respect to the occurrence likelihood of all states j in the set Q.

$$weight^{i,j}_{t_\ell} = \frac{P^j_{t_\ell}}{\sum_{s \in Q} P^s_{t_\ell}} \times \frac{P^i_{t_\ell}}{\sum_{s \in Q_{i,j}} P^s_{t_\ell}} \tag{5.1}$$

Finally, the proposed visualizer displays a movie by flashing each of the 241 time frames on the screen for a short time and then immediately replaces it by the next one. This movie gives the user an idea of how the folding flow changes on time based on the occurrence likelihood for each conformational state.

5.1.2.1 Visualizing folding pathways in a flow network

`efold` defines a folding pathway as a time ordered sequence of folding events in which the unfolded protein is able to assume its native state. `efold` codes these folding events as the ensemble of fully folded structures containing a set of interacting SS. The transition

[2]this is a parameter that may be customized

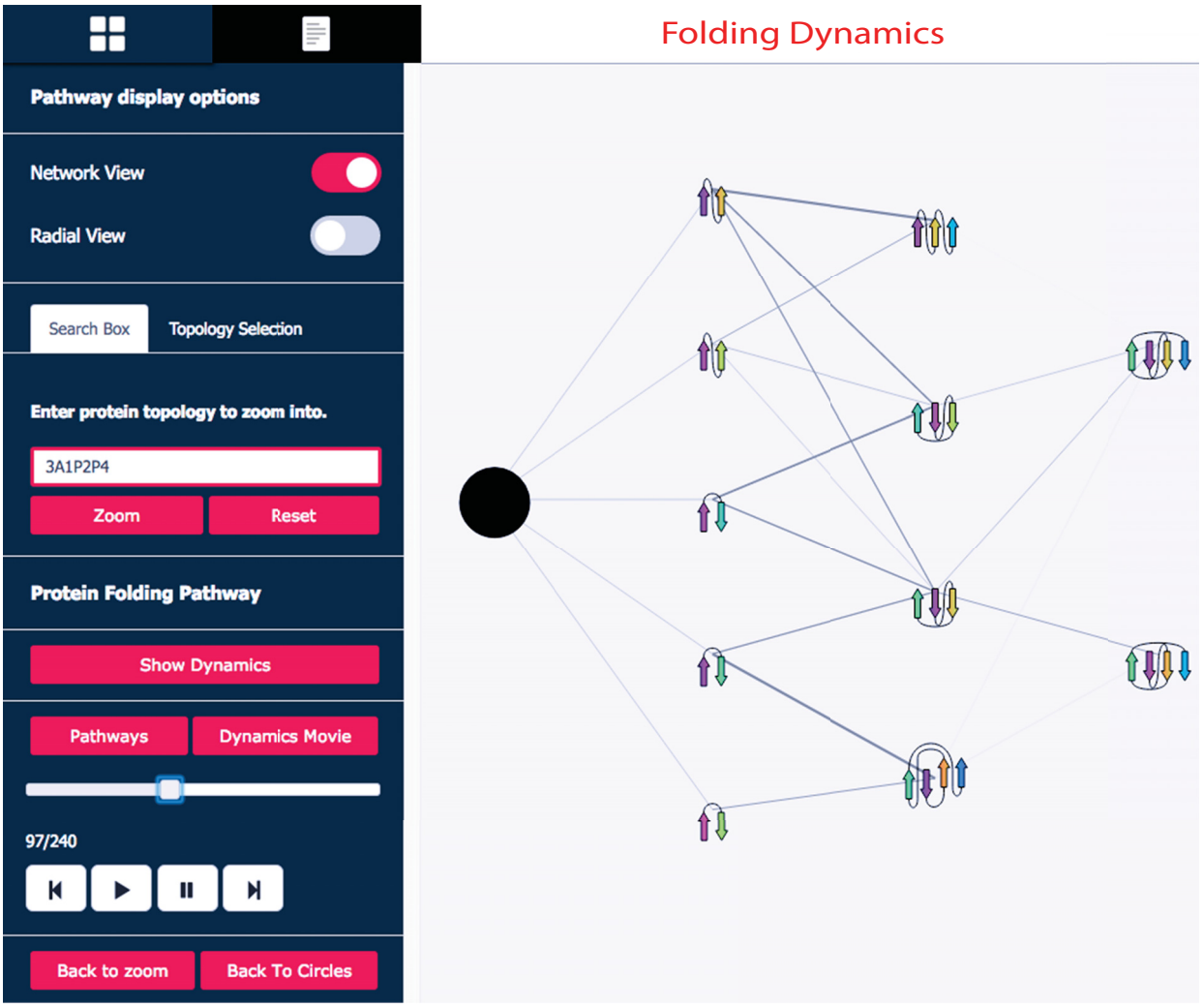

FIGURE 5.2: A visualizer to model protein dynamics

between these events is modeled as a path in a graph (that represent an energy landscape) of different intermediate structures. The order of visiting these transition states is assembled through the use of dynamic folding information (which is also modeled by `efold`). `efold` combines folding dynamics and conformational energy landscape information to model folding pathways as weighted paths that connect the unfolded, intermediate, and native protein states. `efold` uses the analogy of considering protein folding as a flow arising in a network of (un)folding pathways at a coarse grained free energy landscape (see section 3.6.2 for details).

`efold` performs a DFS traversal of the subgraph (which contains all the transition states involved in trajectories connecting the unfolded state with the current state selected by the user) to extract the predicted pathways. Each predicted pathway is associated with

a flow value computed as the minimum edge's weight present in the trajectory between the unfolded and native states (where the weight of the edges is computed by equation 3.16). Following the analogy of PF as a flux in a flow network, the weight of each reported pathway represents the probability that this specific trajectory arises from the network of folding pathways [205–207]. The visualizer uses the width of a line to model the weight (a.k.a flow) of each path. If two or more paths share an edge, the visualizer will plot each path on top of the other to avoid the overlapping of lines. The user can obtain the predicted paths by clicking 'Pathways'. Figure 5.3 shows an example of four pathways found by `efold` (each plotted using a different color).

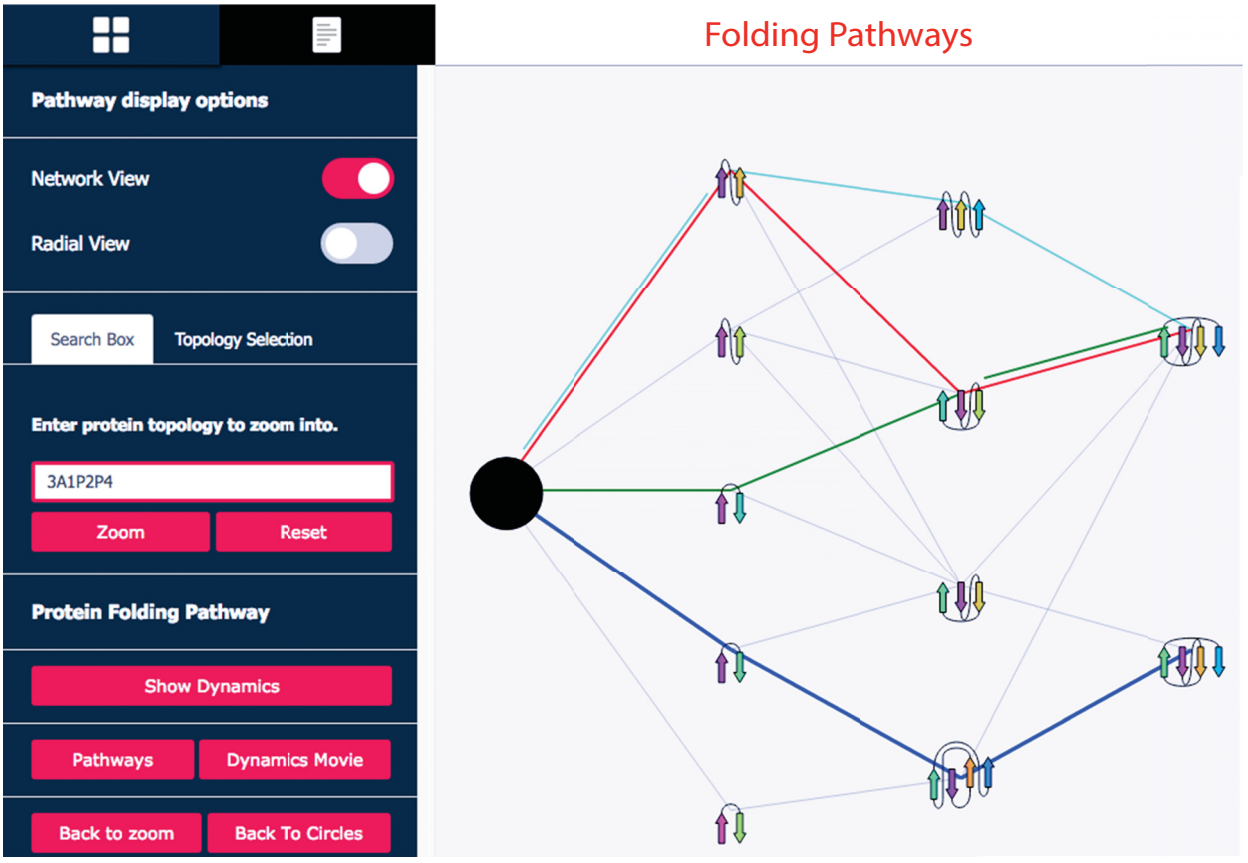

FIGURE 5.3: A visualizer to model protein pathways
A screenshot of the visualizer to model the pathway landscape predicted by `efold`

Chapter 6

Conclusions

6.1 Contributions

An enormous challenge for PF prediction methods has been to predict 3D native structures and folding pathways for the broad range of proteins that are currently known. This set of proteins contains thousands of different folds, different structural families, and unique folding mechanisms. Additionally, the PF problem is an NP-complete problem in lattice models [71, 72] that require significant running time. Reliable predictions for critical features (such as intermediate states, folding rates and folding pathways) of protein foldings have been produced through custom-designed hardware. However, these state of the art methods are currently unable to compute, or even approximate, the complete 3D conformational landscape of all protein targets. Therefore, there is a tangible need to develop efficient and effective protein folding methods in structural biology.

In this thesis, we propose a new and effective coarse grained methodology for the PPP that requires minimal computational resources when compared with classical approaches. This new methodology combines ensemble modeling and evolutionary based sequence information to provide accurate folding pathway predictions. Residue contact information is integrated into a Boltzmann sampling process to circumvent the limitations of potential energy scoring schemes and to narrow the conformational search space (the two most important bottlenecks in PF prediction). The proposed method expands the scope

of previous ensembles prediction techniques and differs from state of the art works on the following features: *i*) improvement in speed, accuracy and flexibility (input requires protein sequence alone and does not require any *a priori* knowledge of the native protein structure. Additionally, `efold` has the ability to model β, α/β and $\alpha+\beta$ interactions); *ii*) implementation of novel approaches to model the folding process (implementation of new energy model mimicking a framework-based folding process and the ability of proteins to adopt different conformational states); *iii*) exploits evolutionary data of proteins adding information about evolutionary constrained interactions into the protein folding prediction; and *iv*) a PPP visualizer that allows for anyone to observe the analysis and interaction between pathway predictions.

6.1.1 Improvement in performance (speed, accuracy and flexibility)

Simulating protein folding has been a very difficult task performed on small structures ($\leq$ 50 AA) through computationally expensive methods. To-date, the classical computational approaches to obtain pathway information rely on time-consuming high-resolution MD, MCM, or fragment assembly methods that are primarily limited to relatively small molecules [32, 48, 49, 101, 285]. These detailed models (typically working on full atomic detail) have the obvious benefit of potentially greater accuracy. However, the computational demands and the increasing complexity of simulations restrict their applicability to most protein systems. Coarse grained models offer an alternative approach that has allowed for the study of folding pathways on larger proteins, but these methods usually introduce constrains that limit the biological significance of simulations [74, 87, 107, 286, 287]. The right balance between the level of detail of a simulation versus the resources conferred to obtain this detail (i.e., an inexpensive and global coarse view versus a detailed and expensive view) is still an open discussion. This thesis contributes to this discussion by embracing an alternate, yet complementary strategy (`efold`), which offers a better trade-off between resolution (i.e., modeling secondary structures and residue contacts) and efficiency. `efold` complements existing PF techniques by addressing the

simulation complexity barrier by computing coarse grained representations of complete energy landscapes at a large scale.

`efold` is methodologically different to previous PPP algorithms and contributes to the field by improving upon the main drawbacks found in state of the art methods. Compared to state of the art PPP methods, `efold` offers a different trade-off between representation detail and computation tractability. `efold`'s prediction of structures from AA sequence alone allows for application to a large corpus of data. Based on our experiments, we believe `efold` is a reliable protein structure and pathway predictor. The ability of `efold` to formulate quick, coarse-grained predictions in a matter of minutes or hours, rather than weeks of atomistic-detail simulation, is invaluable when used to support the initial stages of more complex and detailed models.

Unlike MD/MC simulations, `efold` does not need custom-hardware supercomputer to calculate simulations and is able to recognize the optimal (and overlapping) substructures during runtime. In other words, `efold` has 'memory' and it is able to determine whether protein conformations have already been visited. This attribute allows `efold` to embrace a divide and conquer approach, which assures a much faster running time than MD/MC techniques. `efold` is able to predict more than one trajectory in each run of the algorithm and given that `efold` computes the complete energy landscape, it has less difficulties escaping local minima.

Unlike PRM methods, `efold` does not assume *a priori* knowledge with respect to the protein's native conformation. `efold` has, in general, a better algorithmic efficiency when the configuration space is large. `efold` statistically samples the conformation space while considering biological significance (conformation with SS structure).

Unlike fragment assembly methods, `efold` does not need *a priori* information of structural elements. `efold` can be run on any AA sequence, while generating new insights about the folding process. `efold` statistically characterizes the complete conformational landscape. If an optimization algorithm is used during the assembly simulation, `efold` may have a better algorithmic efficiency.

Unlike unfolding strategies, `efold` does not assume *a priori* knowledge with respect to the protein native conformation. `efold` is not based on the folding microscopic reversibility hypothesis to guarantee biologically significant results.

6.1.2 Novel views to model the folding process

The PF problem is considered among the most compelling scientific challenges facing researchers today. The PF problem could be understood as the conjunction of two related subproblems: *i*) the problem of predicting the 3D structure for biologically active proteins (PSP) and *ii*) the problem of predicting the time ordered folding steps that allow an unfolded protein conformation to be transformed into a functional 3D structure (PPP). Both subproblems have been widely acknowledged as open problems; but due to its inherent complexity, the PPP has received less attention than the PSP. As a consequence, the information that is embedded within folding pathways have remained largely unexploited and there is a need to develop novel computational biology tools and benchmarks to model and validate protein pathways. The work described in this thesis addresses this problem by proposing an algorithm that accurately describes folding pathways by predicting both the protein pathway and structure. In particular, this thesis contributes to closing the gap between PSP and PPP methods.

One central question in the modeling of the protein folding process is whether a native conformation (or ensemble of conformations) corresponds to the most stable (thermodynamic control) or kinetically most accessible (kinetic control) conformation [288–290]. The kinetics and thermodynamics perspective of protein folding is conceptually synthesized through the computation of an energy landscape model. This model shows the evolution of the folding process as a function of folding energy and represents different thermodynamic and kinetic variations in the landscape. `efold` addresses the disjunction between thermodynamics and kinetics by modeling the stability of native states (by statistically characterizing the native conformation) and ensuring accessibility (by modeling the folding dynamics). This algorithm combines folding dynamics and conformational energy landscape information to model folding pathways as weighted paths that connect the

unfolded, intermediate and native protein states. This novel approach to model pathways represents a structured and more inclusive way to model pathways.

There is compelling evidence that suggests that proteins have the ability to adopt different conformational states *in vivo*, where multiple optimal structures could exist with different functional properties. The folded state can then be understood as a small ensemble of conformational structures (that also have the potential to misfold), instead of a unique conformation. Furthermore, the intermediate conformations are not considered discrete states, but an ensemble of structures where the transition between consecutive ensembles on the folding pathway happens on parallel routes (some routes are expected to be more populated than others). It is important to recognize the statistical implications of the protein folding process and to consider that protein ensembles may mimic the ability of proteins to adopt different conformational states in vivo. `efold` is an ensemble algorithm that follows this research line by predicting a statistical distribution of topologically allowed pathways. `efold` describes a realistic energy landscape of conformational variants by predicting an ensemble of protein conformations and pathways (instead of an individual lowest energy structure obtained by a single pathway). Folding pathways are fully described by the complete conformational energy landscape, where proteins fold through distinct intermediate ensemble conformations using multiple routes. The algorithms used by `efold` have been inspired by ensemble prediction algorithms that allow for the accurate computation of RNA secondary structure energy landscapes [80, 118, 119]. Those RNA-based algorithms have been successfully mapped by `efold` to the different domain of predicting protein structure, dynamics, and pathways.

`efold` represents intermediate conformations as ensembles of fully folded structures containing a set of interacting SS, where the intermediate conformations are modeled as obligatory sequential states of the overall folding process. The sequentiality of folding events (assumed by `efold`) implies that certain transitions occur before others. `efold` builds the conformational landscape through the enumeration of all SS (in terms of β-strands) a protein can attain. This enumeration is computed using a stepwise permutation algorithm that adds a single β-strand pairing at a time. Contrary to previous approaches [121], the addition of a new β-strand could be perform at any of the extremes (i.e., left

or right) of a permutation. This new approach is very important because it allows `efold` the flexibility to adopt a more general approach to compute energy values for the SS.

6.1.3 Exploits evolutionary records

In this thesis we propose a novel algorithm that integrates protein sequence information into a Boltzmann sampling process performed by ensemble methods to predict protein pathways. This method takes advantage of available protein sequence information to incorporate evolutionary information (in the form of SS or co-evolutionary residue contacts) into a statistical energy function to improve `efold`'s capabilities to represent key AA interactions.

There is no clear consensus about what the required accuracy, coverage, and distribution of evolutionary information along an AA sequence are needed to improve the prediction of protein structures and / or pathways. Furthermore, there is no consensus about exactly what information in a protein sequence is sufficient and necessary for folding. Our results support the hypothesis that, in general, the incorporation of sequence information into protein folding programs leads to an improvement in the prediction accuracy of protein structures/pathways [108–110, 113, 177–179]. Our results show an increase in accuracy of pathway predictions (by increasing the signal of more populated trajectories) and a decrease in the running time of the algorithm (by constraining the search space of conformations). To the best of our knowledge, this is the first time that evolutionary information is enclosed in an ensemble method to model folding pathways.

One of the main limitations when determining the amount of sequence information needed to define the fold is the vast number of potential cooperative interactions between AAs (i.e., the free energy contribution of one AA depends on those of other AAs) [291]. Proteins are able to create complex networks of AA interactions that require a great number of mutual constraints between AA positions to define the fold [292]. The interaction between AAs are generally modeled by statistical potential energy functions, which compute the interactions responsible for protein structure. These functions generally limit the size and resolution of protein simulations due to the unfeasibility of an adequate and

accurate conformational sampling to provide a complete landscape. There is no consensus in the field about the suitability of using accurate (and computationally complex) functions that make impractical the modeling of large systems versus the practicability of using imprecise (and uncomplicated) scoring schemes that make tractable the modeling of complete landscapes. Our results show that `efold` is able to correctly balance between the amount of needed sequence information and the complexity of scoring functions. `efold` improves the accuracy of its potential energy function by discarding false and unfundamental co-evolved AA interactions. Therefore, `efold` demonstrates that the synergy between statistical analysis of physical predictions (i.e., low free-energy models) and evolutionary based predictions (i.e., sequence variation methods) is useful and practical in the definition of true folding landscapes. `efold` compares favourably to previous studies [293] and shows that an uncomplicated statistical energy function captures the coevolution between AAs that is necessary to correctly model the folding process. The results of `efold` also suggest that folding pathways of proteins might acquire a certain degree of conservation during evolution [294–300].

Long-range interactions play a fundamental role in the stability of proteins [301]. However, they are considered one of the main bottlenecks for protein folding predictions given the difficulty of being accurately predicted. Regarding β-strands, it has been shown that the prediction accuracy has a negative correlation with respect to the separation in sequence of contacting residues (i.e., prediction accuracy drops as AA have contacts with more sequentially distant AA) [302]. `efold` asseses a higher precision in predicting long-range contacts than state of the art template-free algorithms by including evolutionary information in a free-energy model. The correct prediction of long-range contacts allows `efold` to reconstruct folding pathways more accuratelly and to narrow the search space of possible conformations by imposing strong constraints on the 3D structure.

6.1.3.1 Improvements in the MSA problem

MSAs encodes evolutionary and structural relationships in the forms of protein sequence information. Several protein structure prediction techniques and procedures (including

`efold`) are dependent on MSAs to extract sequence, structural and/or functional relationships. The quality of information extracted from MSAs highly depends on the accuracy of MSA and sequencing results, where errors in the alignments could limit the accuracy of downstream analysis. In order to analyze MSA's uncertainty (i.e., number of errors, bias, and range of improvement) we examined the capacity of humans and algorithms to provide insights that can not be entirely replicated by the cost functions used in heuristics-based algorithms. As a result, we discovered that crowdsourcing (for humans) and MOOP (for algorithms) are suitable methodologies to improve the accuracy of pre-computed MSAs. In the case of crowdsourcing, the harnessing of intelligence and processing power generated by crowds of online gamers allow for the use of human's visual pattern recognition skills to improve local inaccuracies that are neglected by the heuristics implemented in alignment software. On the other hand, the proposed MOEA algorithm proved to be less dependent on specific features of sequences, while remaining very stable and robust when used on diverse biological sequences.

6.1.4 Interaction with pathway predictions

Despite many *in-silico* and *in-vitro* experiments that have tried to deduce principles for PF. Currently, there is no general consensus on the folding routes or transition states for arbitrary protein sequences [303]. Substantial improvements have been developed for protein structure methods that have led to accurately predict structures [70]. For example, the best predictions in critical assessments of PSP methods (CASP) have been shown on average to be accurate enough to interpret biological mechanisms, guide biochemical studies, and initiate drug discovery programs. In spite of these improvements, there are still many difficult challenges to achieve in terms of determining folding mechanisms; making *ab initio* predictions consistent enough to decrease the current dependency on knowledge of existing structure; and studying folding diseases, drug affinities, membrane proteins, and disorder proteins; to name a few.

Visualization techniques are needed for the study of protein pathways to increase our understanding of biological processes that have been neglected by PPP (when compared

to PSP). As the number, size, and complexity of macromolecular structures increases, the need for new and novel visualizers grows. In this work, we developed a visualizer to interact with pathway predictions performed by `efold`. This visualizer is essential to guarantee the impact and dissemination of `efold` to the wider biology and computer science communities. By developing a PPP prediction pipeline that can present data to the user community in both a human and machine readable manner, we aim to support large-scale protein simulations as well as single experimental designs.

6.2 Perspectives on future work

6.2.1 Generation of 3D models

The β-contact interactions predicted by `efold` can be used to reconstruct protein 3D structures. The probabilistic contact map reported by `efold` can be used as a starting point for 3D structure reconstruction. Current reconstruction algorithms infer structures with a proportion (around 25%) of the real contacts, but these methods have difficulty reconstructing the 3D structures predicted from contact maps that contain false contacts (a small percentage of false contacts are sufficient enough to hamper 3D reconstruction). The generation of the 3D reconstruction (based on `efold` predictions) is a non trivial task that will face the following challenges: *i*) Currently, there are only a few distinct and successful approaches to the reconstruction of a protein's 3D structure from a map of AA contacts [304–309]. There are even fewer methods able to perform the reconstruction based only on β-residues [310]. *ii*) The overall contact prediction quality achieved by state of the art contact residue methods is still lower than the required levels of accuracy to satisfy 3D structure retrieval protocols. *iii*) Information about AA chirality is lost in a 2D contact map representation and 3D reconstruction protocols applying external forces can introduce chirality errors into the reconstructed structure. *iv*) The general problem of recovering a set of 3D coordinates consistent with some given contact map has been proven to be NP-hard[1] [308, 311]. The 3D structure reconstruction protocols

[1] using a reduction of the unit-disk-graph realization problem

often involves heuristics extracted from 'the novo' prediction that can be time consuming. Thus, an efficient and fast procedure must be found to not hamper the speed benefits of `efold`.

6.2.2 Determinants of protein folding

Anfinsen's experiments suggested that the primary AA sequence of proteins defines their structure and the rate at which a structure is formed. AA sequence is the main determinant of folding mechanisms because it defines the size, stability, structure, and folding kinetics. Several sequence-folding relationship studies have shown that different sequences can adopt similar structures, but each one encodes for a unique free energy surface that may lead to distinct folding behaviors [240]. The question about whether a protein sequence directly defines the distinct folding behaviors (such as different folding pathways) or if it establishes these distinct behaviors by defining other properties of a protein (such as global equilibrium properties, topology) is still an open question. The rational redesign of protein folding pathways (by mutating AA sequences) has been a fundamental tool to improve our understanding of distinct folding behaviors. This protein redesign is based on performing AA mutations that slightly shift the folding barriers of proteins to provide indirect access to transition states by changing the topology, stability and / or folding rates of the protein. Given that `efold` is able to compute complete energy landscapes that contain folding pathways to the known native state, our method enables the study and analysis of landscape behavior properties. `efold` may also be used to determine how sensitive the folding kinetics of proteins are to fine sequence details (modeled for AA mutations).

The development of quantitative theoretical models for the protein folding process is still a necessity in the protein folding field. It is important to be able to determine simple stability, length, folding rates, or topology-based rules for the prediction of protein folding. This type of work will provide insights into the existence of common folding mechanism(s) that underly the diverse kinetic properties of proteins, or if it is the contrary and folding kinetics are sequence-dependant. We have already developed a quantitative theoretical

model for the prediction of folding dynamics driven by local topological features in an iterative manner[2] [312]. This work will allow for the study of energy independent models, but a thorough investigation has yet to be done to clearly establish the determinants of protein folding.

6.2.3 Ability to model all fold classes

The classification of a protein in a specific 'folding class' is based on the arrangement of its SS elements in space. These classes of group structures have similar SS composition, but different overall tertiary structures and evolutionary origins. 'β' (domains consisting of β-sheets), 'α/β' ($\beta-\alpha-\beta$ units) and '$\alpha+\beta$' (segregated α and β regions) are three of the main classes defined by the Structural Classification Of Proteins database (SCOP). These classes represent 59% of the total number of reported folds. `efold` addresses modeling techniques for these three families. `efold` focuses on β-sheet proteins because they represent the vast majority of folds (more than half of the reported folds [and proteins] belong to any of these three classes), they have a high influence in amyloid fibrils and they are difficult to characterize by experimental methods. However, `efold` is able to include α-helices in the modeling thorough the incorporation of evolutionary sequence information. The inclusion of the remaining folding classes (all α proteins, multi-domain proteins, membrane and cell surface proteins, and small proteins) will increase the coverage, and, possibly, the accuracy of `efold`. The inclusion of new interactions (needed to model all the folding classes) is not a straightforward task and will require the redefinition of the DP algorithm as well as the construction of a more inclusive, complex interaction energetic model.

6.2.4 Co-evolving methods

MSAs are in the core of most successful methods for recognizing evolutionary related protein sequences and/or residues. As shown in this chapter, MSAs can be improved by using crowdsourcing and/or multi-objective methods. However, how these improvements

[2]This work is in peer-review phase of a publication process

can increase the power of homology-based methods to recognize remote homologs and model dependencies between AA (that are close in space but far apart in sequence) has to be further studied. Future versions of `efold` could use improvements in MSA and homology detection methods for β-proteins (such as the model proposed in [145]) to study the impact of these improvements in the prediction accuracy of protein pathways.

Computational co-evolution methods are a promising complementary technique to extract and exploit structure and function information from rapidly growing sequence databases. Particularly, `efold` uses co-evolution methods to identify conservation patterns, which are evidence of structural and functional constraints exerted on proteins. Co-evolution algorithms are currently limited by the amount of evolutionary sequence data needed to obtain reliable models. State of the art methods require large numbers of homologous sequences using accurate MSA sufficiently diverse to reveal accurate co-evolution patterns. To circumvent this challenge, new and novel methods to building reliable models from incomplete/inaccurate evolutionary data are needed. This set of new algorithms will help `efold` to increase its spectrum of applicability and versatility for PPP.

Functional proteins are known to undergo natural selection processes preserving their function and hence their structure. The protein energy landscape for a protein sequence is molded by evolution such that its native protein structure is conserved. However, point mutations can reshape the energy landscape of a protein populating certain (un)folding pathways. Evolutionary information, encapsulate within multiple sequence alignments (MSAs), can be used to identify conservation and mutation patterns, which are evidence of structural constraints plus mutational drift. The main hypothesis is that residues in physical contact coevolve (i.e. they show correlated mutational behavior) and that some of them may be both structurally or functionally important positions within protein folds and consequently could be targets for disease-associated point mutations. In conjuction with `efold`, we plan to develop a co-evolutionary algorithm to characterize co-evolving residues leading to disease when disrupted by a point mutation. This novel algorithm will aim the identification of co-variate residues e interactions that significantly (de)stabilize the folding pathways (as predicted by `efold`) of these proteins.

Appendix A

Materials

A.1 MSA: `Open-Phylo` Approach

Table A.1: Data Set MULTIZ
AncOr = Ancestor Original; AncCl = Ancestor Classic; AncEx = Ancestor Expert; GuiOr = Guidance Original; GuiCl = Guidance Classic; GuiEx = Guidance Expert; MusOr = Muscle Original; MusCl = Muscle Classic; MusEx = Muscle Expert; TcoOr = T-Coffee Original; TcoCl = T-Coffee Classic; TcoEx = T-Coffee Expert

BlockName	**AncOr**	**AncCl**	**AncEx**	**GuiOr**	**GuiCl**	**GuiEx**	**MusOr**	**MusCl**	**MusEx**	**TcoOr**	**TcoCl**	**TcoEx**
BRCA1.01	1314.1	1330.4	1349.9	0.5371	0.7136	0.5758	2952	3101	3264	49	51	52
BRCA1.11	1447.3	1475	1455.3	0.1352	0.6458	0.6425	2509	2694	2509	48	48	48
BRCA1.21	2859.9	2918	2903.8	0.3192	0.3816	0.3733	7896	7956	8041	73	73	73
BRCA1.31	3175.4	3190	3178	0.4053	0.4457	0.4374	9183	9205	9186	65	65	65
BRCA1.41	2911.3	2933	2933	0.5188	0.6363	0.5559	8303	8384	8412	82	82	82
BRCA1.51	3304.3	3337.3	3310.3	0.6590	0.6774	0.6714	9569	9651	9592	76	76	77
BRCA1.61	3821.5	3821.5	3821.5	0.7555	0.7555	0.7555	13040	13040	13040	88	88	88
BRCA1.71	1307.8	1307.8	1311.8	0.5065	0.5065	0.6111	3472	3472	3502	74	74	76
BRCA1.s.01	1805	1833.8	1826.9	0.2066	0.3252	0.3099	5037	5168	5121	53	53	54
BRCA1.s.11	2162.7	2195.6	2199.3	0.1445	0.2563	0.1778	4544	4719	4566	56	57	57
BRCA1.s.21	2853.1	2915.3	2885.1	0.3766	0.3983	0.3952	8311	8318	8381	70	70	71
BRCA1.s.31	2917.8	2942.8	2949.5	0.4890	0.5481	0.5383	8524	8612	8702	72	72	72
BRCA1.s.41	3352.1	3373.5	3368.4	0.6353	0.6588	0.6067	9368	9439	9439	80	80	80
BRCA1.s.51	3569.9	3590	3586.5	0.6875	0.6875	0.7117	11690	11710	11800	83	83	84
BRCA1.s.61	3274.3	3274.3	3287.9	0.5012	0.5012	0.5944	10190	10190	10310	82	82	83
P531.01	2401.1	2401.1	2415.2	0.3036	0.3062	0.3368	4679	4813	4851	62	62	62
P531.11	2224.6	2224.6	2225.5	0.5718	0.5718	0.5976	3631	3631	3631	75	75	75
P531.21	2054.3	2058.6	2059	0.5512	0.5602	0.5864	4097	4142	4481	63	63	63
P531.31	3544.9	3544.9	3549.9	0.4624	0.4624	0.5056	7031	7031	7180	63	63	64
P531.41	3188.2	3222.7	3191.7	0.5063	0.6342	0.5808	8528	8632	8582	80	80	80
P531.51	2880.4	2886.4	2891.6	0.4414	0.4618	0.4500	6071	6091	6276	68	68	68
P531.61	2669.4	2728.6	2712	0.3889	0.4386	0.4453	7383	7531	7645	71	71	71
P531.71	3302.5	3302.5	3305.1	0.7736	0.7736	0.7660	9326	9326	9337	84	84	84
P531.81	2391.6	2424.1	2413.9	0.4552	0.6757	0.6021	6612	6717	6869	76	76	76
P531.91	2771	2774.1	2779.3	0.6119	0.6767	0.6518	8513	8535	8561	82	82	83

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
P531.101	1167.3	1187.3	1176.3	0.5531	0.6602	0.5475	3189	3241	3240	59	59	60
P531.111	491	491	491	0.4932	0.4932	0.5024	1483	1483	1483	65	65	65
P531.s.01	2849.8	2849.8	2864.5	0.5488	0.5516	0.5520	7096	7197	7432	76	76	76
P531.s.11	2313.6	2317.9	2318.6	0.5029	0.5062	0.6113	4174	4199	4384	69	69	69
P531.s.21	2070.8	2070.8	2087.2	0.3417	0.3417	0.3162	3604	3604	3855	54	54	54
P531.s.31	3634.5	3634.5	3647.7	0.4323	0.4323	0.4639	7498	7498	7516	82	82	82
P531.s.41	2802.9	2820.7	2811.4	0.4085	0.6570	-1.0000	6790	6907	6839	73	73	73
P531.s.51	2448.2	2495.8	2486.4	0.2853	0.3185	0.3000	5913	6093	6130	62	62	63
P531.s.61	3324.6	3324.6	3327.8	0.6624	0.6624	0.6680	7981	7981	8010	84	84	85
P531.s.71	3068.2	3068.2	3091	0.6577	0.6577	0.7108	8333	8333	8729	83	83	84
P531.s.81	2168.3	2172.3	2173.6	0.5721	0.6327	0.5989	6500	6534	6550	78	78	78
P531.s.91	1943.4	1949.2	1954.1	0.5320	0.5366	0.5438	5236	5270	5418	73	73	73
P531.s.101	1274.6	1294	1282.4	0.4353	0.4951	0.4355	3736	3773	3746	65	65	65
P532.01	3228	3231.6	3249.8	0.5080	0.5315	0.5202	8357	8385	8495	82	82	83
P532.11	2780.5	2815.7	2792.6	0.7477	0.7529	0.7748	8836	8865	8875	90	90	90
P532.21	2752.7	2786.5	2789.1	0.6802	0.7844	0.7470	8261	8301	8643	88	88	88
P532.31	824.3	825.3	824.3	0.6371	0.6400	0.6422	641.9	765.3	641.9	26	26	26
P532.41	1782.4	1786.1	1798.7	0.6159	0.6387	0.6497	3358	3469	3532	55	55	55
P532.51	1688.8	1710.6	1717.1	0.1065	0.1400	0.1198	3276	3367	3499	51	51	51
P532.61	2699.8	2726.1	2705.5	0.7794	0.8772	0.8503	9489	9513	9534	90	90	90
P532.s.01	3115.9	3129.5	3126.5	0.7698	0.8223	0.8012	9908	9937	9968	91	91	91
P532.s.11	3452.1	3455.8	3463.4	0.8879	0.9145	0.9157	11830	11840	11940	97	97	97
P532.s.21	1495.5	1537.5	1500.1	0.2305	0.2812	0.2665	1594	1625	1665	36	36	36
P532.s.31	9.99999	14.2	23.5	-1.0000	-1.0000	-1.0000	399.3	399.3	501.8	36	36	42
P532.s.41	2259.5	2282.6	2277.5	0.4809	0.5895	0.5138	5303	5360	5576	63	63	63
P532.s.51	2644.2	2669.6	2661.4	0.4428	0.4581	0.4595	6084	6205	6333	77	77	77
P532.s.61	967.3	978.9	967.3	0.6781	0.7531	0.7759	3449	3449	3488	91	91	91
P533.01	1346.6	1356.1	1421.3	0.1888	0.2618	0.3618	3307	3309	3356	68	68	68
P533.11	2754.1	2754.1	2762.9	0.6592	0.6761	0.6157	6549	6602	6601	80	80	80

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
P533.21	1497.9	1512.7	1514.3	0.3176	0.3778	0.3814	2830	2856	3043	52	52	54
P533.31	1376.6	1376.6	1376.6	0.4979	0.6126	0.5872	1899	1954	1995	51	51	51
P533.41	1337.8	1379.5	1339.8	0.2645	0.2685	0.2645	2638	2676	2638	28	28	28
P533.51	942.8	948	1005.1	0.1204	0.1517	0.1178	1495	1495	1732	33	33	33
P533.61	420.1	435.1	442.5	0.0642	0.1051	0.1087	968.6	968.6	1130	33	33	33
P533.71	1448	1473.6	1452.3	0.2730	0.3388	0.2466	5102	5109	5102	49	49	49
P533.81	146.8	165.7	173	0.6972	0.7659	0.7189	617.4	642.5	661.6	69	69	69
P533.s.01	2038.5	2046	2061.4	0.3737	0.3924	0.5534	5663	5663	5704	78	78	78
P533.s.11	1307.1	1311.4	1398.2	0.2694	0.2705	0.2801	2348	2348	3200	46	46	46
P533.s.21	2099.7	2118.5	2099.7	0.3926	0.4401	0.4529	4082	4085	4196	42	42	43
P533.s.31	986.6	986.6	986.6	0.4665	0.4665	0.6104	721.1	721.1	747.2	35	35	35
P533.s.41	1466.7	1466.7	1483.6	0.1609	0.2633	0.1676	2252	2313	2362	38	38	38
P533.s.51	413.2	414.3	434.7	0.1192	0.1583	0.1127	603.5	681.9	785.6	32	32	32
P533.s.61	1041.8	1043.8	1045.8	0.2460	0.2559	0.2715	3242	3242	3251	64	64	64
P533.s.71	830.4	865.9	880.8	0.5003	0.6125	0.5159	3297	3356	3398	43	43	44
RB1.01	811.9	830.6	879.6	0.3569	0.3664	0.4208	2261	2290	2496	56	56	57
RB1.11	757.3	761.6	791.9	0.0560	0.1351	0.1101	1931	2043	2021	32	32	33
RB1.21	1037.9	1038.2	1052.5	0.3462	0.3827	0.3961	4060	4201	4363	69	69	69
RB1.s.31	1171.1	1174	1186.5	0.5716	0.6091	0.6781	4685	4785	4944	74	74	74
RB1.31	989.5	1006.7	1007.6	0.6818	0.7644	0.7172	5759	5953	6116	84	84	85
RB1.41	1551.9	1558.1	1578.2	0.1275	0.6327	0.6425	7399	7443	7425	76	76	76
RB1.51	1709.4	1712.1	1713.7	0.4980	0.4980	0.5340	5831	5831	5864	70	70	70
RB1.61	1429.1	1476.7	1443.4	0.5233	0.6899	0.5517	6016	6268	6276	66	66	67
RB1.71	1611.1	1611.1	1627	0.6954	0.7487	0.7445	7279	7300	7582	82	82	83
RB1.81	487.2	493.8	495.8	0.7128	0.9726	0.8419	2412	2462	2495	92	92	93
RB1.s.01	874.5	898	933.1	0.1196	0.2560	0.1850	3162	3429	3386	45	45	45
RB1.s.11	735.8	752.6	746.1	0.3004	0.5047	0.3932	3439	3570	3591	64	64	64
RB1.s.41	1451.1	1458.6	1480.8	0.3982	0.6514	0.3727	6168	6239	6428	69	69	69
RB1.s.51	2008.9	2016	2041.8	0.5359	0.6280	0.6448	6807	6841	7032	68	68	68

Continued...

BlockName	**AncOr**	**AncCl**	**AncEx**	**GuiOr**	**GuiCl**	**GuiEx**	**MusOr**	**MusCl**	**MusEx**	**TcoOr**	**TcoCl**	**TcoEx**
RB1.s.21	1294.1	1301.3	1307.1	0.2318	0.7381	0.6314	7367	7509	7569	84	84	85
RB1.s.61	1299.5	1319.8	1315.5	0.6662	0.6688	0.7403	6158	6160	6264	82	82	83
RB1.s.71	1368.8	1375.4	1382.6	0.6197	0.6376	0.6349	6058	6119	6291	81	81	82

Table A.2: Data Set PRANK

AncOr = Ancestor Original; AncCl = Ancestor Classic; AncEx = Ancestor Expert; GuiOr = Guidance Original; GuiCl = Guidance Classic; GuiEx = Guidance Expert; MusOr = Muscle Original; MusCl = Muscle Classic; MusEx = Muscle Expert; TcoOr = T-Coffee Original; TcoCl = T-Coffee Classic; TcoEx = T-Coffee Expert

BlockName	**AncOr**	**AncCl**	**AncEx**	**GuiOr**	**GuiCl**	**GuiEx**	**MusOr**	**MusCl**	**MusEx**	**TcoOr**	**TcoCl**	**TcoEx**
BRCA1.01	1528.4	1534.5	1532	0.5693	0.6293	0.6055	2983	2983	3029	45	45	45
BRCA1.11	1780.3	1780.3	1782.3	0.2714	0.2955	0.2480	3036	3301	3036	46	46	47
BRCA1.21	3364.2	3386.1	3365.2	0.4756	0.4807	0.4601	8126	8154	8214	73	73	74
BRCA1.31	3683.2	3685.2	3683.2	0.7300	0.7893	0.6937	9508	9540	9588	65	65	65
BRCA1.41	3322.5	3339.1	3325.4	0.7814	0.8377	0.7814	8838	8929	8848	82	82	82
BRCA1.51	3828.2	3828.2	3831.5	0.6263	0.6753	0.6282	9656	9656	9731	75	75	75
BRCA1.61	4335.7	4335.7	4344.7	0.8197	0.8197	0.8191	13070	13070	13230	88	88	88
BRCA1.71	1505.7	1505.7	1506	0.2320	0.2587	0.2320	3753	3753	3788	71	71	71
BRCA1.s.01	2144.2	2156	2148.5	0.4460	0.5600	0.5158	5258	5311	5307	48	48	49
BRCA1.s.11	2511.6	2512.9	2537.1	0.2943	0.2943	0.2725	4633	4667	4896	53	53	55
BRCA1.s.21	3392	3392	3402.9	0.5672	0.6189	0.5762	8532	8572	8607	74	74	74
BRCA1.s.31	3354.8	3355.4	3354.8	0.6363	0.6887	0.6418	8595	8630	8724	72	72	73
BRCA1.s.41	3859.8	3872.9	3859.8	0.6664	0.6782	0.6694	9568	9612	9577	80	80	80
BRCA1.s.51	4110.6	4110.6	4122.2	0.8185	0.8185	0.7912	11910	11910	12050	85	85	85
BRCA1.s.61	3710.3	3710.3	3719.2	0.8181	0.8181	0.8578	10420	10420	10470	82	82	82
P531.01	2698.9	2702.3	2713.3	0.5012	0.5123	0.4880	4900	5003	4948	62	62	62
P531.11	2511.6	2511.6	2527.2	0.4990	0.4990	0.5030	4752	4752	4774	74	74	74
P531.21	2343.8	2352.5	2343.8	0.6959	0.6972	0.7094	4574	4579	4722	63	63	63
P531.31	3868.3	3868.3	3877.9	0.3695	0.3695	0.4041	7068	7068	7102	64	64	65
P531.41	3547.1	3547.1	3547.1	0.5589	0.5589	0.5772	8515	8515	8665	79	79	79
P531.51	3221.7	3227.2	3227.7	0.5243	0.6180	0.6011	6199	6259	6436	68	68	69
P531.61	3076.4	3076.4	3076.4	0.5792	0.6145	0.5626	7870	7894	7919	72	72	72
P531.81	2765.5	2767.5	2767.5	0.5663	0.5937	0.5930	6595	6649	6748	71	71	71
P531.71	3698.9	3698.9	3698.9	0.7042	0.7042	0.7340	9504	9504	9574	84	84	84
P531.91	3047.1	3050.5	3049.1	-1.0000	0.5367	0.6853	7842	7869	7857	80	80	80

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
P531.101	1291.6	1291.6	1339.7	0.6597	0.6814	0.6494	2603	2611	2680	48	48	48
P531.111	556	556	556	0.6902	0.6902	0.6902	1498	1498	1500	70	70	70
P531.s.01	3217.9	3219.7	3223.9	0.6914	0.7035	0.7116	7388	7446	7418	76	76	76
P531.s.11	2589.3	2599.1	2596.2	0.5823	0.5823	0.6757	4487	4487	4559	68	68	68
P531.s.21	2670.5	2670.5	2683.4	0.2386	0.2386	0.2178	4158	4158	4223	53	53	53
P531.s.31	3979	3979	3982.3	0.5293	0.5293	0.5120	7297	7297	7357	83	83	83
P531.s.41	3143.2	3143.2	3143.2	0.5537	0.6267	0.5944	6854	6859	6907	71	71	71
P531.s.51	2803.7	2816.1	2821.1	0.4205	0.4869	0.5550	6389	6476	6416	64	64	64
P531.s.61	3720.9	3720.9	3726.8	0.7140	0.7140	0.6798	8101	8101	8145	84	84	85
P531.s.71	3456.2	3456.2	3457.8	0.7258	0.7258	0.8049	8694	8694	8787	83	83	83
P531.s.81	2513.8	2522.6	2513.8	0.5917	0.5917	0.4386	6368	6391	6368	74	74	75
P531.s.91	1892.7	1892.7	2038.7	0.2582	0.5241	0.1531	4087	4191	4087	68	68	68
P531.s.101	1435.9	1435.9	1438.7	0.6018	0.6226	0.6018	3438	3528	3494	63	63	63
P532.01	3664.7	3691.7	3669	0.4968	0.5253	0.5367	8958	8978	8958	81	81	82
P532.11	3115.4	3115.4	3129.1	0.7649	0.7649	0.8179	8902	8902	8994	89	89	90
P532.21	2948.5	2948.5	3002.3	0.7969	0.7969	0.6892	8541	8541	8638	89	89	89
P532.31	833.8	833.8	833.8	0.4200	0.4200	0.4155	545.5	545.5	609.6	18	18	18
P532.41	1955.1	1955.1	1955.1	0.6307	0.6307	0.6464	3404	3404	3408	53	53	53
P532.51	1947.8	1947.8	1950.6	0.3700	0.4319	0.4489	3744	3746	3891	52	52	53
P532.61	3035.4	3035.4	3035.4	0.8288	0.8288	0.8288	9711	9711	9711	90	90	90
P532.s.01	3478.8	3478.8	3485.4	0.7736	0.7736	0.8518	10080	10080	10190	90	90	91
P532.s.11	3868.6	3868.6	3868.6	0.9047	0.9047	0.9109	11930	11930	12020	96	96	96
P532.s.21	1525.9	1525.9	1572.5	0.3738	0.3738	0.3116	1778	1778	1854	28	28	28
P532.s.31	502.3	517.1	539.4	0.4678	0.8771	0.8521	1694	2126	2793	66	66	66
P532.s.41	2504	2504	2506.7	0.3797	0.4319	0.3914	4902	4902	4950	59	59	59
P532.s.51	2983.1	2990.2	2993.9	0.5381	0.5485	0.3526	6570	6601	6644	78	78	78
P532.s.61	1087.3	1087.3	1087.3	0.7443	0.7443	0.7443	3509	3509	3543	91	91	92
P533.01	1583.8	1583.8	1632.9	0.3074	0.3074	0.3213	2450	2450	3072	68	68	68
P533.11	3161.1	3162.6	3161.8	0.2609	0.5840	0.5595	7047	7113	7119	79	79	79

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
P533.21	1807.8	1810.5	1808.6	0.4479	0.5077	0.3670	2855	2855	2855	49	49	49
P533.41	1313	1329.1	1350.8	0.5487	0.6569	0.4179	2605	2679	2754	26	26	26
P533.51	1138.2	1138.2	1138.2	0.2397	0.3981	0.2662	1556	2037	1556	30	30	30
P533.61	709.4	710.1	709.4	0.5158	0.8294	0.5898	1845	1908	1845	39	39	42
P533.71	1916.6	1916.6	1921	0.3939	0.5074	0.5474	5177	5177	5177	56	56	56
P533.81	298.9	298.9	310.1	0.4378	0.4378	0.5321	729.2	729.2	729.2	66	66	66
P533.s.01	2244.7	2254.2	2351.8	0.3828	0.4675	0.4773	4994	5054	5609	77	78	77
P533.s.11	1687.9	1689.9	1693.1	0.3257	0.4869	0.4152	2747	2884	2747	51	51	51
P533.s.21	2276.7	2278.7	2455.3	0.5101	0.5662	0.5342	3814	3820	3913	41	41	41
P533.s.41	1805.5	1805.5	1805.5	0.2235	0.3412	0.2486	2751	2786	2751	37	37	37
P533.s.51	681.8	681.8	685.8	0.3553	0.3569	0.4300	1189	1189	1235	28	28	28
P533.s.61	1441	1443.8	1461.2	0.5648	0.7443	0.4950	3628	3628	3631	67	67	67
P533.s.71	1216	1235	1236	0.6643	0.6643	0.6134	3462	3511	3482	47	47	47
RB1.01	1235.4	1258.4	1239.7	0.5150	0.8462	0.7641	3050	3991	3050	61	61	62
RB1.11	1161.1	1161.1	1161.1	0.1832	0.8584	0.8116	2497	2564	2497	38	38	38
RB1.21	1477.8	1477.8	1477.8	0.5708	0.6240	0.5233	5630	5630	5630	70	70	70
RB1.31	1324	1324	1328.9	0.7113	0.7113	0.7389	6506	6506	6579	85	85	86
RB1.41	1947.5	1953.7	1974.9	0.6262	0.6471	0.7143	7996	8017	7996	80	80	80
RB1.51	2064.9	2070.3	2073.6	0.5823	0.6470	0.5940	6251	6276	6372	69	69	69
RB1.61	1833.9	1833.9	1833.9	0.6243	0.6523	0.6621	6558	6561	6558	68	68	68
RB1.71	2012	2016	2019.7	0.8345	0.8345	0.4190	8032	8032	8032	83	83	83
RB1.81	610.8	610.8	611.1	0.9095	0.9095	0.9079	2574	2574	2604	93	93	94
RB1.s.01	1311.8	1312.1	1311.8	0.3455	0.4718	0.3253	4172	4172	4172	53	53	53
RB1.s.11	920.9	944.6	935.9	0.2839	0.6226	0.2804	3111	3182	3111	45	45	45
RB1.s.21	1691.9	1691.9	1691.9	0.8093	0.8093	0.8093	8414	8414	8440	86	86	86
RB1.s.31	1485.8	1490	1499.2	0.5071	0.8049	0.5917	5391	5439	5577	72	72	74
RB1.s.41	1822.8	1857.9	1838.1	0.4077	0.5124	0.4546	6634	6659	6634	71	71	71
RB1.s.51	2467.3	2472.5	2470.2	0.6882	0.8239	0.6905	7473	7473	7473	70	70	70
RB1.s.61	1619.7	1625	1627.4	0.7823	0.7982	0.7857	6515	6523	6515	82	82	82

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
RB1.s.71	1730.6	1732.6	1730.9	0.8117	0.8117	0.8215	6665	6665	6674	84	84	84

Table A.3: Data Set MUSCLE
AncOr = Ancestor Original; AncCl = Ancestor Classic; AncEx = Ancestor Expert; GuiOr = Guidance Original; GuiCl = Guidance Classic; GuiEx = Guidance Expert; MusOr = Muscle Original; MusCl = Muscle Classic; MusEx = Muscle Expert; TcoOr = T-Coffee Original; TcoCl = T-Coffee Classic; TcoEx = T-Coffee Expert

BlockName	**AncOr**	**AncCl**	**AncEx**	**GuiOr**	**GuiCl**	**GuiEx**	**MusOr**	**MusCl**	**MusEx**	**TcoOr**	**TcoCl**	**TcoEx**
BRCA1.01	1512.1	1541.6	1518.1	0.6757	0.6859	0.6754	-1	-1	3520	48	48	48
BRCA1.11	1790.4	1818.3	1804	0.2513	0.2725	0.2277	3594	3608	3594	50	50	50
BRCA1.21	3328.3	3329.4	3373.2	0.5602	0.5685	0.3980	8568	8568	8568	74	74	74
BRCA1.31	3697.9	3709.6	3700.2	0.5325	0.5529	0.5692	9827	9830	9831	65	65	65
BRCA1.41	3279.5	3281.8	3289.1	0.7434	0.7660	0.7736	8778	8778	8778	82	83	82
BRCA1.51	3826.3	3843.3	3837	0.6862	0.7157	0.7130	9985	9985	9985	78	78	78
BRCA1.61	4346.7	4346.7	4354.7	0.7599	0.7599	0.6896	13180	13180	13180	89	89	89
BRCA1.71	1510.5	1510.5	1512.5	0.4995	0.4995	0.5842	3671	3671	3689	77	77	77
BRCA1.s.01	2136.8	2160.2	2139.1	0.5408	0.5835	0.5506	5868	5868	5868	50	50	50
BRCA1.s.11	2519.2	2526.7	2519.7	0.2190	0.2238	0.2203	5519	5519	5519	56	56	56
BRCA1.s.21	3359.4	3370.9	3361.4	0.4599	0.4715	0.4674	8840	8883	8857	72	72	72
BRCA1.s.31	3321.6	3335.6	3321.6	0.4701	0.5923	0.5252	9193	9230	9193	70	71	70
BRCA1.s.41	3863.2	3880.2	3895.1	0.6537	0.6779	0.6888	9758	9758	9758	81	81	81
BRCA1.s.51	4109.2	4109.2	4132.8	0.8019	0.8019	0.8003	12060	12060	12110	85	85	85
BRCA1.s.61	3737.6	3737.6	3739.6	0.5053	0.5053	0.5337	10520	10520	10520	83	83	83
P531.01	2737.2	2737.2	2760.9	0.3394	0.3431	0.3582	5357	5357	5388	62	62	62
P531.11	2495.4	2495.4	2519.3	0.5533	0.5533	0.5496	4879	4879	4949	76	76	76
P531.21	2338.5	2338.5	2339.9	0.6459	0.6544	0.6685	4809	4809	4809	61	61	61
P531.31	3895.6	3895.6	3895.6	0.4922	0.4922	0.4783	7124	7124	7124	66	66	66
P531.41	3546.1	3564	3546.7	0.6324	0.6473	0.6296	8796	8849	8805	81	81	81
P531.51	3244.5	3244.5	3264.7	0.5641	0.5641	0.5739	6647	6647	6647	69	69	69
P531.61	3086.9	3086.9	3087.1	0.5896	0.5896	0.6234	8118	8118	8118	75	75	75
P531.71	3679.7	3679.7	3692	0.7488	0.7488	0.7628	9703	9703	9703	84	84	85
P531.81	2715.9	2715.9	2750.2	0.5397	0.5616	0.5806	7070	7070	7070	72	72	72
P531.91	3042.7	3043.1	3047	0.6945	0.7009	0.7187	8040	8040	8044	81	81	81

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
P531.101	1294.1	1324.3	1309.7	0.5300	0.5373	0.5037	2798	2804	2802	49	49	49
P531.111	547.7	547.7	561	0.3717	0.3717	0.6200	1484	1484	1523	70	70	70
P531.s.01	3208.3	3210.3	3228	0.5378	0.5378	0.5722	7660	7660	7660	76	76	76
P531.s.11	2584.1	2584.1	2591.6	0.5950	0.6052	0.6109	4687	4687	4722	68	68	68
P531.s.21	2338.4	2338.4	2349.8	0.3779	0.3779	0.3850	4385	4385	4385	55	55	55
P531.s.31	3895.6	3895.6	3895.6	0.4731	0.4731	0.4930	7124	7124	7124	66	66	66
P531.s.41	3151.6	3151.6	3155.6	0.5130	0.5391	0.5293	7119	7119	7123	72	72	72
P531.s.51	2832.6	2837.1	2832.6	0.5791	0.5791	0.5832	6766	6766	6766	66	66	66
P531.s.61	3739	3739	3740.3	0.6899	0.6899	0.7184	8278	8278	8369	86	86	87
P531.s.71	3453.1	3453.1	3457.4	0.6652	0.6652	0.6602	8994	8994	8994	83	83	83
P531.s.81	2477.6	2481.3	2499.8	0.5290	0.5323	0.5358	6594	6653	6594	75	75	75
P531.s.91	2053.7	2057	2053.7	0.4359	0.4778	0.4680	3533	3550	3906	72	72	72
P531.s.101	1451.1	1458.6	1453.1	0.3987	0.5565	0.4293	3733	3733	3747	69	69	69
P532.01	3688.1	3690.6	3698.9	0.6289	0.6307	0.6105	9308	9308	9308	81	81	82
P532.11	3130	3130	3132	0.8908	0.8908	0.8945	9149	9149	9149	90	90	90
P532.21	3042.3	3043.1	3046.3	0.7050	0.7070	0.7263	8764	8766	8764	89	89	89
P532.31	878.6	878.6	878.6	0.5457	0.5955	0.5916	873.5	873.5	906.9	22	22	22
P532.41	1891.7	1891.7	1898.7	0.6739	0.6739	0.6545	3569	3569	3569	55	55	55
P532.51	1972	1972	1977.8	0.2919	0.2919	0.3096	4256	4256	4256	54	54	54
P532.61	3028.6	3028.6	3036.4	0.6738	0.7221	0.6939	9839	9839	9839	90	90	90
P532.s.01	3482.8	3482.8	3489.1	0.8057	0.8057	0.8146	10200	10200	10200	91	91	91
P532.s.11	3869.8	3869.8	3871.1	0.9108	0.9108	0.9150	12120	12120	12120	97	97	97
P532.s.21	1594.4	1594.4	1622.5	0.1962	0.2236	0.2144	2124	2124	2124	37	37	37
P532.s.31	518.4	544.4	589.6	0.8085	0.9227	0.7958	1979	1979	3152	65	65	65
P532.s.41	2551.9	2551.9	2560.2	0.4540	0.4564	0.4555	5716	5716	5716	63	63	63
P532.s.51	3006.9	3006.9	3025.2	0.4669	0.4945	0.4713	7127	7127	7127	79	79	79
P532.s.61	1081.3	1081.3	1092.5	0.8254	0.8254	0.7738	3560	3560	3560	91	91	92
P533.01	1584.4	1595.4	1652.5	0.2254	0.2361	0.3643	3548	3548	3548	69	69	69
P533.11	3176.4	3178.4	3176.4	0.6354	0.6354	0.6635	7262	7262	7262	80	80	80

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
P533.21	1854.5	1861.8	1854.8	0.2117	0.2252	0.2356	3379	3379	3389	49	49	49
P533.31	1343.2	1343.2	1343.2	0.5744	0.6641	0.6041	1900	1961	1900	49	49	49
P533.41	1427.9	1440.4	1438.8	0.5152	0.5299	0.5244	3338	3338	3338	23	23	24
P533.51	1044.2	1045.5	1060.6	0.1997	0.1997	0.1699	1979	1979	1979	37	37	37
P533.61	652.7	654.4	652.7	0.2763	0.3476	0.3532	2067	2067	2067	42	42	42
P533.71	1901.2	1943.9	1926.7	0.4598	0.4667	0.4539	5578	5578	5588	57	57	57
P533.81	310.5	310.8	310.5	0.3500	0.6305	0.4464	882.3	895.4	882.3	66	66	67
P533.s.01	2355.5	2355.5	2368.1	0.3881	0.3881	0.5742	6066	6066	6066	77	77	77
P533.s.11	1786.9	1786.9	1787.6	0.2261	0.2261	0.2337	3183	3701	3183	52	52	52
P533.s.21	2348.3	2361.5	2348.3	0.4215	0.4565	0.4435	4476	4476	4493	42	42	42
P533.s.31	820.4	825.5	820.4	0.6491	0.7052	0.6827	883.3	927.3	883.3	31	31	31
P533.s.41	1814.5	1843	1814.5	0.1777	0.1777	0.1583	3129	3129	3129	36	36	36
P533.s.51	648.7	673.1	648.7	0.1692	0.2431	0.2168	1545	1545	1545	32	32	32
P533.s.61	1443.6	1447.7	1445.6	0.4854	0.5002	0.4997	3683	3683	3701	69	69	69
P533.s.71	1192.3	1234.9	1238.4	0.3232	0.4488	0.3290	3784	3784	3784	48	48	48
RB1.01	1284.8	1297.2	1284.8	0.4559	0.4760	0.4419	3456	3456	3456	61	61	62
RB1.11	1115.6	1120.9	1115.6	0.2250	0.3647	0.2388	2797	2797	2797	34	35	34
RB1.21	1457	1457	1459	0.3833	0.4132	0.4297	5722	5722	5722	70	70	70
RB1.31	1320.6	1320.6	1320.6	0.6899	0.6899	0.7024	6719	6719	6719	86	86	86
RB1.41	1979.4	1981.4	1985.7	0.8375	0.8375	0.8226	8169	8169	8169	80	80	80
RB1.51	2065.4	2065.4	2069.4	0.4258	0.4341	0.4215	6469	6469	6469	71	71	71
RB1.61	1810.5	1810.5	1813.4	0.5092	0.5548	0.5489	6758	6758	6758	69	69	69
RB1.71	2002.6	2008.3	2012.7	0.7473	0.7640	0.7847	8001	8018	8001	83	83	83
RB1.81	610.8	610.8	611.1	0.8778	0.8778	0.8730	2612	2612	2612	93	93	94
RB1.s.01	1286.9	1290	1290.2	0.3590	0.4212	0.3441	4358	4365	4358	52	52	53
RB1.s.11	936.3	952.8	942.8	0.3757	0.4480	0.4181	3487	3503	3487	46	46	46
RB1.s.21	1689.6	1702.8	1696.7	0.7214	0.7499	0.7968	8421	8421	8421	86	86	87
RB1.s.31	1492.8	1492.8	1492.8	0.4620	0.4819	0.5842	5603	5603	5603	70	70	75
RB1.s.41	1845	1858	1847.7	0.5314	0.5314	0.4555	7147	7157	7147	74	74	74

Continued...

BlockName	**AncOr**	**AncCl**	**AncEx**	**GuiOr**	**GuiCl**	**GuiEx**	**MusOr**	**MusCl**	**MusEx**	**TcoOr**	**TcoCl**	**TcoEx**
RB1.s.51	2456.8	2468.6	2460	0.6084	0.6431	0.6174	7369	7369	7369	69	69	69
RB1.s.61	1600.7	1609.3	1602.7	0.7546	0.8087	0.7874	6577	6585	6610	82	82	82
RB1.s.71	1718.5	1723.5	1718.8	0.7715	0.8068	0.7585	6611	6611	6618	83	83	83

Table A.4: Data Set T-Coffee
AncOr = Ancestor Original; AncCl = Ancestor Classic; AncEx = Ancestor Expert; GuiOr = Guidance Original; GuiCl = Guidance Classic; GuiEx = Guidance Expert; MusOr = Muscle Original; MusCl = Muscle Classic; MusEx = Muscle Expert; TcoOr = T-Coffee Original; TcoCl = T-Coffee Classic; TcoEx = T-Coffee Expert

BlockName	**AncOr**	**AncCl**	**AncEx**	**GuiOr**	**GuiCl**	**GuiEx**	**MusOr**	**MusCl**	**MusEx**	**TcoOr**	**TcoCl**	**TcoEx**
BRCA1.01	1485.1	1487.6	1528.7	0.5198	0.5402	0.5747	3106	3135	3294	45	45	46
BRCA1.11	1692.9	1698	1702.3	0.1418	0.1506	0.1718	2462	2661	2595	45	45	45
BRCA1.21	3297	3307.4	3347.8	0.3339	0.3942	0.4271	8000	8044	8346	72	72	74
BRCA1.31	3660.8	3671.4	3700	0.4181	0.4370	0.5458	9363	9403	9593	65	65	66
BRCA1.41	3319.9	3350.9	3354.8	0.5456	0.5456	0.5456	8538	8670	8538	82	83	83
BRCA1.51	3853.5	3854.8	3855.8	0.6438	0.6595	0.6801	9697	9840	9727	77	77	77
BRCA1.61	4342.9	4342.9	4355.5	0.7668	0.7668	0.7836	13110	13110	13200	88	88	88
BRCA1.71	1501.8	1501.8	1513.4	0.4697	0.4697	0.6291	3500	3500	3663	74	74	77
BRCA1.s.01	2089.3	2110.3	2111.2	0.2629	0.2629	0.2709	5066	5167	5232	48	48	49
BRCA1.s.11	2485.5	2495.1	2490.8	0.1466	0.1509	0.1580	4686	4769	4686	54	55	55
BRCA1.s.21	3317.1	3337.5	3357.9	0.3756	0.4066	0.4493	8496	8525	8620	71	71	72
BRCA1.s.31	3327.4	3353.6	3338.4	0.5107	0.5239	0.5301	8700	8756	8795	72	72	72
BRCA1.s.41	3884.7	3898.3	3889	0.6492	0.6912	0.6556	9644	9721	9688	80	80	80
BRCA1.s.51	4121.5	4121.5	4121.5	0.6803	0.6803	0.7010	11820	11880	12010	84	84	84
BRCA1.s.61	3729.6	3729.6	3740.2	0.4892	0.4892	0.5774	10220	10220	10390	82	82	83
P531.01	2736.5	2736.5	2745.4	0.3124	0.3132	0.3285	4871	4871	4954	62	62	62
P531.11	2504.4	2504.4	2516.9	0.5487	0.5487	0.6038	3830	3830	4943	76	76	76
P531.21	2278.4	2301.7	2318.9	0.5300	0.5420	0.5995	4133	4150	4226	61	61	62
P531.31	3849.8	3849.8	3867	0.4213	0.4213	0.4373	7070	7070	7121	65	65	65
P531.41	3512.8	3526.7	3556.2	0.5114	0.5164	0.5578	8399	8419	8744	80	80	80
P531.51	3232.6	3233.6	3240.9	0.4450	0.4450	0.4922	6371	6371	6473	68	68	68
P531.61	3006.2	3054.1	3055.9	0.7628	0.4254	0.4923	7375	7545	7772	71	71	73
P531.71	3684.7	3684.7	3692.3	0.3345	0.3345	0.7913	9578	9578	9694	85	85	85
P531.81	2714.7	2714.7	2767.3	0.5357	0.5357	0.5357	6502	6502	6644	72	72	72
P531.91	3013.4	3030.2	3017.7	0.3921	0.5026	0.3921	7383	7433	7524	80	80	80

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
P531.101	1289.5	1321.3	1295.7	0.4537	0.4708	0.4702	2510	2539	2609	49	49	49
P531.111	561	561	561	0.5332	0.5332	0.5332	1526	1526	1526	67	67	67
P531.s.01	3213.8	3216.1	3225.3	0.5625	0.5823	0.5604	7278	7281	7395	76	76	76
P531.s.11	2584.3	2595.4	2593.8	0.4684	0.4758	0.5567	4444	4460	4458	69	69	69
P531.s.21	2266.3	2266.3	2293.7	0.2913	0.2913	0.3480	3800	3800	3920	54	54	54
P531.s.31	3956.1	3956.1	3983.6	0.4475	0.4475	0.5181	7338	7338	7422	83	83	83
P531.s.41	3133	3133	3141.6	0.4482	0.5253	0.4925	6958	6975	7018	72	72	72
P531.s.51	2762.9	2804.6	2807.6	0.2813	0.3272	0.4371	5957	6171	6457	63	63	65
P531.s.61	3729.5	3729.5	3738.1	0.6861	0.6861	0.6896	8187	8187	8238	85	85	86
P531.s.71	3439.5	3439.5	3461.4	0.6423	0.6423	0.6746	8463	8463	8758	83	83	84
P531.s.81	2491.7	2507.3	2508.4	0.4456	0.5445	0.4739	6459	6534	6590	76	76	76
P531.s.91	2079.1	2105.4	2083.8	0.4497	0.4775	0.4394	3398	4267	3765	72	72	72
P531.s.101	1437.5	1450.4	1445.2	0.3683	0.3683	0.3759	3562	3588	3570	65	65	65
P532.01	3654.1	3655.8	3669.8	0.5220	0.5365	0.5458	8799	8810	8914	81	81	82
P532.31	866.1	885	890.7	0.6091	0.6165	0.6143	735.7	774.1	749.6	27	27	27
P532.21	3049.4	3049.4	3054.8	0.6734	0.6734	0.7150	8410	8410	8776	89	89	89
P532.41	1896.2	1896.2	1908.4	0.5766	0.5766	0.6326	3204	3204	3419	53	53	54
P532.51	1925.4	1925.4	1955.6	0.1242	0.1242	0.1838	3622	3622	3976	53	53	55
P532.61	3039.8	3046.4	3045.3	0.7847	0.8345	0.7911	9746	9767	9761	90	90	91
P532.s.01	3476.9	3476.9	3490.5	0.7648	0.7648	0.8530	10080	10080	10190	90	90	91
P532.s.11	3864.1	3864.1	3872.4	0.8972	0.8972	0.9171	12030	12030	12070	97	97	97
P532.s.31	537.8	541	547.1	0.7649	0.8198	0.7722	1601	2403	1735	63	63	63
P532.s.41	2524.2	2524.2	2552.8	0.4511	0.4511	0.5162	5261	5261	5601	62	62	63
P532.s.51	2979.1	2979.1	3011.3	0.4331	0.4367	0.4841	6519	6519	6998	78	78	79
P532.s.61	1087.3	1087.3	1092.5	0.7100	0.7100	0.7889	3514	3514	3557	91	91	92
P533.01	1550.6	1550.6	1602	0.2263	0.2263	0.3592	3433	3433	3457	68	68	68
P533.11	3125.8	3135.8	3151.3	0.4670	0.6074	0.4757	6797	6850	7014	80	80	80
P533.31	1412.7	1412.7	1412.7	0.5193	0.5857	0.5942	2079	2132	2201	52	52	52
P533.21	1775	1790.5	1814.7	0.2277	0.2925	0.3042	2975	2981	3046	49	49	49

Continued...

BlockName	AncOr	AncCl	AncEx	GuiOr	GuiCl	GuiEx	MusOr	MusCl	MusEx	TcoOr	TcoCl	TcoEx
P533.41	1568.4	1593.7	1583.2	0.2996	0.3365	0.2807	2940	3115	3023	25	25	25
P533.51	1018.7	1018.7	1018.7	0.1528	0.1773	0.1528	1563	1563	1566	34	34	35
P533.61	624.7	641.2	632.4	0.0666	0.1003	0.0803	1259	1259	1419	42	42	42
P533.71	1924	1932.3	1934.6	0.3190	0.3542	0.3543	5436	5471	5537	57	57	58
P533.81	308.5	312.3	310.5	0.4693	0.5160	0.5460	757.5	835.3	771.3	67	67	67
P533.s.11	1641.5	1647.8	1664.5	0.2165	0.2165	0.2129	2937	3501	2985	49	49	50
P533.s.21	2411.6	2412.1	2411.6	0.3139	0.3674	0.4248	4176	4188	4274	42	42	42
P533.s.31	954.4	954.4	954.4	0.4729	0.4729	0.5683	594.3	594.3	650.1	31	31	31
P533.s.41	1759.2	1780	1804	0.0487	0.1156	0.0974	2362	2448	2441	35	35	35
P533.s.51	598.9	658.5	652.2	0.0889	0.2608	0.1346	890.9	1012	1105	35	35	36
P533.s.61	1407.8	1408.1	1409.8	0.2332	0.2530	0.2697	3549	3574	3549	68	68	68
P533.s.71	1202.1	1208.9	1219.3	0.5014	0.5622	0.5477	3622	3671	3653	48	48	48
RB1.01	1185.6	1185.9	1228.9	0.3764	0.4074	0.4011	2711	2887	3176	60	60	62
RB1.11	1130.4	1149	1146	0.0705	0.1186	0.0854	2453	2480	2462	40	40	40
RB1.21	1443.8	1446.1	1443.8	0.3767	0.4386	0.4147	4771	4816	5097	70	70	70
RB1.31	1305.5	1322.4	1321.7	0.6672	0.8046	0.6954	6261	6459	6569	84	84	85
RB1.41	1945.9	1964.8	1979.6	0.6360	0.8268	0.7070	7804	7909	8056	79	79	80
RB1.51	2061.4	2067.6	2079.7	0.4307	0.4562	0.4751	6215	6215	6328	70	70	70
RB1.61	1785.5	1811.5	1814.2	0.4897	0.5655	0.5329	6312	6505	6683	67	67	68
RB1.71	1992.3	2001.5	2014	0.6902	0.7229	0.7725	7640	7700	8027	81	81	82
RB1.81	603.8	603.8	611.8	0.7263	0.7263	0.8381	2562	2562	2607	93	93	94
RB1.s.01	1245	1271.1	1287	0.2502	0.3102	0.2737	3590	3640	3932	52	52	53
RB1.s.11	898.4	930.3	902.8	0.1435	0.1639	0.1859	2474	2665	2594	46	46	47
RB1.s.21	1696.8	1699.1	1701.1	0.5827	0.7155	0.6047	7777	7866	8089	86	86	86
RB1.s.31	1491.1	1491.1	1507.7	0.5528	0.6231	0.6787	5102	5102	5581	77	77	77
RB1.s.41	1819.1	1862.6	1846.5	0.3554	0.6005	0.3737	6469	6677	6623	70	70	70
RB1.s.51	2416.1	2426.8	2457.2	0.5520	0.6211	0.6260	7143	7212	7349	69	69	70
RB1.s.61	1594.6	1630.5	1608	0.6723	0.6800	0.7115	6345	6422	6572	81	81	82

A.2 MSA: MOEA Approach

Table A.5: Data Set BAliBase RV11
SP = Sum of Pairs; TC = Total Column

GROUP	Metal	SP	TC
BB11001	0.0647343	0.974	0.94
BB11002	0.630466472	0.523	0
BB11003	0.420567376	0.665	0.48
BB11004	0.473871734	0.605	0.43
BB11005	0.604277485	0.476	0.11
BB11006	0.665912739	0.448	0.25
BB11007	0.377058433	0.688	0.4
BB11008	0.393442623	0.507	0.44
BB11009	0.545524691	0.671	0.6
BB11010	0.713801289	0.348	0.17
BB11011	0.765422886	0.298	0.09
BB11012	0.164835165	0.918	0.86
BB11013	0.88547486	0.161	0
BB11014	0.27265987	0.799	0.65
BB11015	0.24743382	0.792	0.67
BB11016	0.712388037	0.31	0
BB11017	0.336630399	0.749	0.65
BB11018	0.581952007	0.628	0.32
BB11019	0.395745631	0.673	0.21
BB11020	0.400215408	0.672	0.35
BB11021	0.343994314	0.74	0.58
BB11022	0.740112994	0.152	0
BB11023	0.548793607	0.53	0.23
BB11024	0.761670762	0.28	0
BB11025	0.929506546	0.105	0
BB11026	0.845039683	0.122	0
BB11027	0.710380835	0.29	0
BB11028	0.556884971	0.615	0
BB11029	0.517156863	0.545	0.48
BB11030	0.599717541	0.251	0
BB11031	0.620681324	0.572	0.18
BB11032	0.395189628	0.657	0.33
BB11033	0.687259615	0.361	0
BB11034	0.631143075	0.511	0.11
BB11035	0.525667351	0.503	0.3
BB11036	0.557480695	0.559	0.24
BB11037	0.419277799	0.48	0.27
BB11038	0.311403079	0.775	0.62

Table A.6: Data Set BAliBase RV12
SP = Sum of Pairs; TC = Total Column

GROUP	Metal	SP	TC
BB12001	0.248978377	0.836	0.63
BB12002	0.073557692	0.925	0.79
BB12003	0.107518797	0.951	0.88
BB12004	0.093418381	0.932	0.74
BB12005	0.109449761	0.914	0.73
BB12006	0.091594828	0.94	0.91
BB12007	0.199289564	0.874	0.74
BB12008	0.11007397	0.92	0.72
BB12009	0.075044405	0.915	0.78
BB12010	0.166051408	0.899	0.75
BB12011	0.256740383	0.735	0.43
BB12012	0.402721999	0.579	0.43
BB12013	0.098156114	0.942	0.86
BB12014	0.077857143	0.995	0.97
BB12015	0.180122264	0.915	0.82
BB12016	0.070922703	0.811	0.61
BB12017	0.156193353	0.926	0.82
BB12018	0.150624688	0.919	0.88
BB12019	0.320613311	0.868	0.76
BB12020	0.171314741	0.861	0.76
BB12021	0.133480176	0.902	0.84
BB12022	0.147763578	0.938	0.88
BB12023	0.192629935	0.873	0.78
BB12024	0.102774923	0.937	0.88
BB12025	0.336221122	0.793	0.67
BB12026	0.140294057	0.882	0.63
BB12027	0.123828593	0.915	0.73
BB12028	0.255699733	0.85	0.74
BB12029	0.211871477	0.862	0.92
BB12030	0.26651202	0.911	0.81
BB12031	0.24989121	0.83	0.68
BB12032	0.137815126	0.922	0.77
BB12033	0.391297518	0.678	0.45
BB12034	0.158974359	0.889	0.8
BB12035	0.141859421	0.925	0.63
BB12036	0.089525108	0.952	0.91
BB12037	0.170810217	0.852	0.64
BB12038	0.12738063	0.862	0.61
BB12039	0.156277436	0.879	0.7
BB12040	0.099848714	0.961	0.94
BB12041	0.38490566	0.678	0.45
BB12042	0.350215161	0.726	0.57

Continued...

GROUP	Metal	SP	TC
BB12043	0.186645932	0.908	0.52
BB12044	0.190547185	0.878	0.71

TABLE A.7: Data Set BAliBase RV20
SP = Sum of Pairs; TC = Total Column

GROUP	Metal	SP	TC
BB20001	0.428599647	0.444	0
BB20002	0.645660683	0.082	0
BB20003	0.05836958	0.953	0.28
BB20004	0.13537186	0.894	0.6
BB20005	0.271342145	0.777	0.07
BB20006	0.076326023	0.959	0.54
BB20007	0.274010069	0.789	0.34
BB20008	0.491632111	0.702	0
BB20009	0.115206725	0.918	0.65
BB20010	0.195015399	0.794	0.28
BB20011	0.370794078	0.719	0.02
BB20012	0.368658935	0.693	0.19
BB20013	0.402535677	0.651	0.22
BB20014	0.247722349	0.805	0.13
BB20015	0.502617112	0.628	0
BB20016	0.298916121	0.659	0
BB20017	0.079124207	0.943	0.55
BB20018	0.05012664	0.963	0.68
BB20019	0.241520581	0.778	0.03
BB20020	0.168531367	0.871	0.62
BB20021	0.123597025	0.856	0.02
BB20022	0.14835866	0.887	0.36
BB20023	0.190046434	0.871	0.3
BB20024	0.527361319	0.526	0.38
BB20025	0.092377077	0.933	0.41
BB20026	0.215528719	0.837	0.34
BB20027	0.439853698	0.833	0.47
BB20028	0.068791561	0.976	0.69
BB20029	0.368444998	0.739	0
BB20030	0.114507481	0.914	0.24
BB20031	0.235581181	0.366	0
BB20032	0.108904865	0.922	0.1
BB20033	0.080496321	0.938	0.33
BB20034	0.175656202	0.144	0
BB20035	0.167942967	0.023	0
BB20036	0.261217328	0.749	0
BB20037	0.189485843	0.85	0.16
BB20038	0.102563331	0.932	0.4

Continued...

GROUP	Metal	SP	TC
BB20039	0.101113204	0.923	0.15
BB20040	0.136318165	0.896	0.57
BB20041	0.364598672	0.725	0.39

TABLE A.8: Data Set BAliBase RV30
SP = Sum of Pairs; TC = Total Column

GROUP	Metal	SP	TC
BB30001	0.178569604	0.854	0.38
BB30002	0.313825746	0.803	0.37
BB30003	0.480782507	0.564	0.09
BB30004	0.184260359	0.864	0.6
BB30005	0.215663825	0.842	0.4
BB30006	0.199403255	0.667	0.39
BB30007	0.209010731	0.787	0.5
BB30008	0.346587323	0.708	0.25
BB30009	0.65232001	0.465	0
BB30010	0.110047302	0.919	0.55
BB30011	0.067976495	0.946	0.73
BB30012	0.270637534	0.71	0.36
BB30013	0.342556642	0.687	0.32
BB30014	0.144206963	0.888	0.54
BB30015	0.161179577	0.801	0.57
BB30016	0.478647347	0.519	0
BB30017	0.384086414	0.709	0.38
BB30018	0.106232703	0.915	0.48
BB30019	0.344607601	0.674	0.33
BB30020	0.362516492	0.535	0
BB30021	0.406128115	0.626	0.14
BB30022	0.199267399	0.824	0.27
BB30023	0.231673087	0.798	0.31
BB30024	0.232150302	0.753	0.27
BB30025	0.529501016	0.702	0
BB30026	0.339260526	0.75	0.35
BB30027	0.444649539	0.637	0.15
BB30028	0.163341102	0.872	0.23
BB30029	0.214852679	0.832	0.51
BB30030	0.44828616	0.573	0

TABLE A.9: Data Set BAliBase RV40
SP = Sum of Pairs; TC = Total Column

GROUP	Metal	SP	TC
BB40001	0.560969874	0.765	0
BB40002	0.67351214	0.585	0

Continued...

GROUP	Metal	SP	TC
BB40003	0.175839023	0.901	0.76
BB40004	0.109060805	0.912	0.4
BB40005	0.195813076	0.899	0.78
BB40006	0.326310632	0.733	0.34
BB40007	0.456078083	0.716	0.33
BB40008	0.182988083	0.868	0.66
BB40009	0.270969965	0.746	0.43
BB40010	0.152522936	0.82	0.53
BB40011	0.374066574	0.686	0
BB40012	0.177789872	0.85	0.58
BB40013	0.20557954	0.754	0.38
BB40014	0.337378338	0.741	0.46
BB40015	0.552407789	0.445	0
BB40016	0.265996704	0.866	0
BB40017	0.266204016	0.866	0
BB40018	0.099724116	0.867	0.7
BB40019	0.140580317	0.903	0.75
BB40020	0.207057146	0.88	0.69
BB40021	0.239595948	0.857	0.48
BB40022	0.445564267	0.605	0
BB40023	0.266791951	0.718	0
BB40024	0.406017974	0.589	0
BB40025	0.163606729	0.879	0.72
BB40026	0.491159437	0.532	0
BB40027	0.652249867	0.431	0
BB40028	0.107695634	0.909	0.61
BB40029	0.152439294	0.891	0.59
BB40030	0.265340685	0.804	0.58
BB40031	0.464574158	0.637	0.99
BB40032	0.347360126	0.91	0.84
BB40033	0.277196444	0.79	0.53
BB40034	0.338133021	0.693	0.31
BB40035	0.695920698	0.582	0
BB40036	0.106513418	0.933	0.63
BB40037	0.550783513	0.248	0
BB40038	0.628854139	0.59	0
BB40039	0.179451343	0.866	0.52
BB40040	0.222960599	0.782	0.45
BB40041	0.33623975	0.682	0
BB40042	0.253969413	0.78	0
BB40043	0.277216539	0.696	0.34
BB40044	0.462905069	0.678	0
BB40045	0.265130985	0.605	0.25
BB40046	0.439239494	0.609	0.17
BB40047	0.322234001	0.886	0.49

Continued...

GROUP	Metal	SP	TC
BB40048	0.127824519	0.907	0.7
BB40049	0.46502176	0.77	0.25

Table A.10: Data Set BAliBase RV50
SP = Sum of Pairs; TC = Total Column

GROUP	Metal	SP	TC
BB50001	0.375019245	0.794	0.39
BB50002	0.641637438	0.441	0.02
BB50003	0.4995148	0.601	0.29
BB50004	0.116108381	0.961	0.88
BB50005	0.11994335	0.942	0.77
BB50006	0.47332213	0.597	0
BB50007	0.414677887	0.616	0.02
BB50008	0.247516511	0.876	0.63
BB50009	0.397203112	0.697	0
BB50010	0.434067832	0.794	0.36
BB50011	0.527964222	0.593	0
BB50012	0.470385789	0.616	0
BB50013	0.148084146	0.928	0.71
BB50014	0.24181047	0.821	0.52
BB50015	0.599248225	0.451	0
BB50016	0.536709542	0.735	0.33

A.3 PPP: `efold` Approach

Table A.11: Benchmark of Standard Proteins

Pdb	Ch	Len	Str	MS	mS	Gap	Topology	Pfam	Msa1	Msa2
1EM7	A	56	4	7	5	4	2A1P4A3	PF01378	50	17
1UBQ	A	76	5	9	3	2	2A1P5A3A4	PF00240	11560	5266

Table A.12: Benchmark of Heteromorphic Proteins

Pdb	Ch	Len	Str	MS	mS	Gap	Topology	Pfam	Msa1	Msa2
1EM7	A	56	4	7	5	4	2A1P4A3	PF01378	50	17
GA1	A	56	4	7	5	4	N/A	N/A	N/A	33
GB1	A	56	4	7	5	4	2A1P4A3	N/A	N/A	16
GA30	A	56	4	7	5	4	N/A	N/A	N/A	32
GB30	A	56	4	7	5	4	2A1P4A3	N/A	N/A	16
GA77	A	56	4	7	5	4	N/A	N/A	N/A	33
GB77	A	56	4	7	5	4	2A1P4A3	N/A	N/A	11
GA88	A	56	4	7	5	4	N/A	N/A	N/A	24

Continued...

Pdb	Ch	Len	Str	MS	mS	Gap	Topology	Pfam	Msa1	Msa2
GB88	A	56	4	7	5	4	2A1P4A3	N/A	N/A	11
GA95	A	56	4	7	5	4	N/A	N/A	N/A	16
GB95	A	56	4	7	5	4	2A1P4A3	N/A	N/A	14
1MHX	A	57	4	9	5	2	2A1P4A3	PF01378	50	16
1MI0	A	57	4	8	4	4	2A1P4A3	PF01378	50	16

Table A.13: Benchmark of Pfam proteins extracted from BetaSheet916 data set

Pdb	Ch	Len	Str	MS	mS	Gap	Topology	Pfam	Msa1	Msa2
1D0D	A	60	2	6	6	5	1A2	PF00014	4915	2
1BUN	B	61	2	7	7	4	1A2	PF00014	4915	2262
5PTI	A	58	3	7	1	4	2A1A3	PF00014	4915	2274
1BIK	A	110	4	7	7	4	1A2N3A4	PF00014	4915	2227
1OOT	A	58	5	6	3	2	4A3A2A1A5	PF00018	10749	7299
1I0C	A	59	5	6	4	3	4A3A2A1A5	PF00018	10749	7392
2HDA	A	64	5	6	3	3	4A3A2A1A5	PF00018	10749	7506
1NEG	A	65	5	6	3	2	4A3A2A1A5	PF00018	10749	7363
1CMX	B	76	4	6	4	5	2A1P4A3	PF00240	11560	5273
1UBQ	A	76	5	9	3	2	2A1P5A3A4	PF00240	11560	5266
1EUV	B	79	5	7	2	2	2A1P5A3A4	PF00240	11560	704
1KQ1	H	66	5	9	6	2	5A1A2A3A4	PF01423	8102	442
1HK9	E	68	5	9	5	3	5A1A2A3A4	PF01423	8102	476
1H64	K	71	5	11	4	2	5A1A2A3A4	PF01423	8102	2445

Table A.14: Benchmark of proteins extracted from BetaSheet916 data set

Pdb	Ch	Len	Str	MS	mS	Gap	Topology	Pfam	Msa1	Msa2
1PTQ	A	50	2	4	4	21	1A2	PF00130	6123	1409
1BX7	A	51	2	4	4	2	1A2	PF02822	460	11
1JJ2	T	53	2	4	4	5	1A2	PF01246	1060	354
1D0D	A	60	2	6	6	5	1A2	PF00014	4915	2
1BUN	B	61	2	7	7	4	1A2	PF00014	4915	2262
1BJP	A	62	2	7	7	30	1P2	PF01361	2760	2149
1ICF	I	65	2	4	4	4	1A2	PF00086	2274	710
1GYJ	A	76	2	6	6	32	2P1	PF01361	2760	1752
1QGW	A	76	2	9	9	22	1A2	PF02972	25	26
1YCQ	A	88	2	3	3	13	1A2	PF02201	1464	53
1T3U	B	93	2	7	7	2	1A2	PF05164	2881	1026
1A5K	A	100	2	8	8	2	1A2	PF00547	1774	576
1KRL	A	44	3	12	7	2	3A2A1	PF09200	2	1
1H59	B	45	3	4	2	11	1A3A2	PF00219	1051	134
1B13	A	54	3	3	2	5	2A1A3	PF00301	2943	1775
5PTI	A	58	3	7	1	4	2A1A3	PF00014	4915	2274
1BXY	A	60	3	6	5	15	2A1A3	PF00327	5033	1342
1O7Z	A	61	3	7	4	5	3A2A1	PF00048	2117	890

Continued...

Pdb	Ch	Len	Str	MS	mS	Gap	Topology	Pfam	Msa1	Msa2
1M8A	A	61	3	7	4	4	3A2A1	PF00048	2117	1011
1NR2	A	62	3	6	4	4	1A2A3	PF00048	2117	998
1JZA	A	66	3	4	2	4	1A3A2	PF00537	487	267
1A15	A	67	3	9	4	3	3A2A1	PF00048	2117	1017
1MOG	A	67	3	12	10	7	1A3A2	PF07311	611	399
1B33	N	67	3	7	7	16	2A1A3	PF01383	448	277
1DI2	A	69	3	8	7	2	1A2A3	PF00035	8915	3159
1EAY	D	69	3	5	5	21	2A1A3	PF09078	622	536
1WVN	A	74	3	8	4	8	1A3A2	PF00013	19558	3278
1UCR	B	75	3	6	2	2	1A3A2	PF08679	56	48
1ESR	A	76	3	6	4	4	1A2A3	PF00048	2117	1020
1F9P	A	81	3	7	4	4	1A2A3	PF00048	2117	993
1EC6	A	87	3	7	4	11	1A3A2	PF00013	19558	3319
1R8H	A	86	3	9	6	20	2A1A3	PF00511	565	266
1JS2	A	89	3	4	2	7	1A2A3	PF01355	231	155
1LZW	A	91	3	8	5	23	2A1A3	PF02617	2895	974
1PUG	A	94	3	6	6	4	1A2A3	PF02575	4505	1567
1G2R	A	94	3	6	4	5	2A1A3	PF04296	2403	988
1PYT	A	92	3	6	6	17	2A1A3	PF02244	1016	988
1MK0	A	97	3	8	6	5	2A1A3	PF01541	9798	554
1DCO	A	99	3	6	4	3	1A3A2	PF01329	2229	1342
1JOS	A	100	3	8	6	5	1A2P3	PF02033	4543	1627
1NN7	A	105	3	5	3	2	2A1P3	PF02214	3855	1447
1I4J	B	110	3	10	8	22	1A3A2	PF00237	6446	1447
1OCU	A	134	3	11	7	8	1A2A3	PF00787	7076	2478
1H6H	A	143	3	12	7	4	1A2A3	PF00787	7076	2133
1MGT	A	169	3	10	8	1	1A2A3	PF01035	7975	2133
1FD4	A	41	4	5	2	5	1P3A4A2	PF00711	464	191
1DUR	A	55	4	3	2	5	1A4N2A3	PF00037	8206	16202
1MHN	A	59	4	10	5	4	1A2A3A4	PF06003	460	920
1MI0	A	61	4	8	5	4	2A1P4A3	PF01378	50	17
1S0Y	A	62	4	7	2	2	1P2N4A3	PF01361	2760	2127
1D1M	B	65	4	7	4	2	1A2A3A4	PF09048	146	2237
1J2L	A	68	4	3	2	2	1A2N3A4	PF00200	2414	1030
1CC7	A	72	4	7	6	4	4A1A3A2	PF00403	19664	8841
1HZ5	A	72	4	8	6	4	2A1P4A3	PF02246	19	5
1XXA	C	73	4	6	5	3	1A2A4A3	PF02863	4296	855
1UB4	C	75	4	4	2	2	1A2N3A4	PF04014	4979	1719
1CMX	B	76	4	6	4	5	2A1P4A3	PF00240	11560	5273
1H75	A	76	4	5	4	2	2P1A3A4	PF00462	11246	14902
1JB0	C	80	4	5	4	7	4A1N2A3	PF12838	18155	15923
1IQZ	A	81	4	4	3	5	1A4N2A3	PF13370	1093	10671
1Q5Y	A	84	4	11	8	4	4A1A3A2	PF08753	1066	515
1E44	A	84	4	9	3	5	3A4A1A2	PF03513	22	32
1NFJ	A	87	4	11	4	9	1P2A4A3	PF01918	745	285

Continued...

Pdb	Ch	Len	Str	MS	mS	Gap	Topology	Pfam	Msa1	Msa2
1PCH	A	88	4	7	4	2	1A4A2A3	PF00381	8181	2262
1TIG	A	88	4	8	5	7	1P2A4A3	PF00707	4680	1292
1UDV	A	88	4	11	3	6	1P2A4A3	PF01918	745	199
1FBQ	B	90	4	5	2	6	1A2A4A3	PF00447	1989	1102
1G7C	B	90	4	12	9	2	4A1A3A2	PF00736	874	413
1BC7	C	93	4	4	3	5	1N2N4N3	PF00178	2375	330
1URN	A	96	4	6	4	9	4A1A3A2	PF00076	50512	21110
1LN4	A	98	4	6	3	2	1P2A4A3	PF01985	3330	862
1J27	A	98	4	13	6	6	2A3A1A4	PF04456	654	357
1NO5	B	102	4	6	4	13	3P2A1A4	PF01909	8895	3591
1IIB	A	103	4	7	4	12	2P1P3P4	PF02302	18949	1485
1Q4R	A	103	4	9	6	12	2A3A1A4	PF07876	1481	1100
1KTE	A	105	4	5	4	2	2P1A3A4	PF00462	11246	9885
1RQM	A	105	4	7	6	2	2P1A3A4	PF00085	24231	23677
1MFW	A	106	4	7	4	8	2A1P4A3	PF03607	1063	303
1OAP	A	108	4	5	3	20	1A4A2P3	PF00691	17270	8639
1BIK	A	110	4	7	7	4	1A2N3A4	PF00014	4915	2227
1EM8	B	110	4	5	4	11	2P1P3P4	PF03603	745	156
1NZ0	D	111	4	7	4	3	1A2A4P3	PF00825	4169	1744
1GNK	B	112	4	10	7	20	4A1A3A2	PF00543	4823	2440
1RMD	A	116	4	3	3	5	1A2N3A4	PF00097	9094	10670
1RLK	A	116	4	10	5	11	2P1A4A3	PF01981	1006	684
1P9Y	A	117	4	8	6	4	1A2A4A3	PF05697	4472	1639
1GNU	A	117	4	8	3	12	2A1P4A3	PF02991	1011	436
1DPT	A	117	4	8	5	14	2P1A3P4	PF01187	669	943
1PXW	B	120	4	6	4	15	1A4A2P3	PF01248	5260	1958
1HUF	A	123	4	7	3	2	1A2N3A4	PF09013	68	6
1GPQ	B	128	4	9	6	4	4A3A2A1	PF08816	506	90
1JL3	A	137	4	7	4	16	2P1P3P4	PF01451	7884	4201
1B66	A	138	4	14	5	2	1A2A3A4	PF01242	3812	1766
1IR2	I	140	4	9	2	4	2A1A3A4	PF00101	1831	592
1FX3	B	149	4	16	14	2	1A4A3A2	PF02556	1987	561
1BE4	A	151	4	9	3	22	4A1A3A2	PF00334	5820	2277
1QTN	A	152	4	9	7	22	2P1P3P4	PF00656	4033	1038
1J98	A	154	4	10	8	3	1A2A4A3	PF02664	2731	477
1IHK	A	157	4	4	4	5	1A2N3A4	PF00167	1638	627
1KLO	A	162	4	4	3	4	1A2N3A4	PF00053	16492	3796
1K8K	F	167	4	7	4	3	1A2A3A4	PF05856	366	111
1EJB	B	168	4	9	5	26	2P1P3P4	PF00885	4443	1709
1OOT	A	58	5	6	3	2	4A3A2A1A5	PF00018	10749	7299
1I0C	A	59	5	6	4	3	4A3A2A1A5	PF00018	10749	7392
1IGU	B	60	5	7	5	2	4A3A2A1A5	PF06613	67	29
1VIE	A	60	5	8	3	3	5A1A2A3A4	PF06442	33	4
1KU6	B	61	5	6	4	5	1A2N4A3A5	PF00087	594	287
1ONJ	A	61	5	8	3	2	1A2N4A3A5	PF00087	594	319

Continued...

Pdb	Ch	Len	Str	MS	mS	Gap	Topology	Pfam	Msa1	Msa2
2HDA	A	64	5	6	3	3	4A3A2A1A5	PF00018	10749	7506
1NEG	A	65	5	6	3	2	4A3A2A1A5	PF00018	10749	7363
1KQ1	H	66	5	9	6	2	5A1A2A3A4	PF01423	8102	442
1AZP	A	66	5	8	4	2	1A2N3A4A5	PF02294	33	4
1HZA	A	67	5	9	4	2	3A2A1A4A5	PF00313	14886	4062
1HK9	E	68	5	9	5	3	5A1A2A3A4	PF01423	8102	476
1GCQ	C	69	5	6	3	3	4A3A2A1A5	PF07653	4617	5122
1H64	K	71	5	11	4	2	5P1P2P3P4	PF01423	8102	2445
1LR7	A	73	5	5	2	4	1A2N4A3A5	PF09289	363	1629
1C1Y	B	77	5	7	5	3	2A1P5A3A4	PF02196	710	64
1EUV	B	79	5	7	2	2	2A1P5A3A4	PF00240	11560	704
1FM0	D	81	5	5	2	2	2A1P5A3A4	PF02597	6781	3066
1BB9	A	83	5	6	4	3	5A1A2A3A4	PF14604	9839	6652
1PHT	A	83	5	8	5	3	2A1A5N3A4	PF07653	4617	6364
1KWA	A	88	5	8	4	7	1A5A4N2A3	PF00595	26099	6219
1COZ	B	126	5	7	5	14	5P4P1P2P3	PF01467	14169	8554
1E6K	A	130	5	7	4	16	2P1P3P4P5	PF00072	151337	19609

Table A.15: Topologies predicted using a HMM

Pdb	Topology	Pdb	Topology	Pdb	Topology
1A5K	1A2	1GYJ	1P2	1MK0	2A1A3
1AZP	1A2	1H59	1A2A3	1MOG	1A3A2
1B13	3A1A2	1H64	4P3P2P1P5	1NEG	5A1A2A3A4
1B33	1P3A2	1H6H	1A2A3	1NFJ	1P2A4A3
1B66	1A2A3A4	1H75	4A3A1A2	1NN7	2A1P3
1BB9	5A1A2A3A4	1HK9	5A1A2A3A4	1NO5	3P2A1A4
1BE4	2A3A1A4	1HUF	1A2A3A4	1NR2	1A2A3
1BIK	1A2N3A4	1HZ5	2A1A4A3	1NZ0	3P4A2A1
1BJP	1P2	1HZA	1A5A6N2A3A4	1O7Z	1A2A3
1BUN	1A2	1I0C	4A3A2A1A5	1OAP	1A4A2P3
1BX7	1A2	1ICF	1A2	1OCU	1A2A3
1BXY	2A1A3	1IGU	1A2P4A3P5	1ONJ	4A3A2A1A5
1C1Y	2A1A3A4A5	1IHK	1A2A4A3	1OOT	4A3A2A1A5
1CC7	2A3A1A4	1IIB	1P2A3P4	1P9Y	3A4A2A1
1CMX	2A1P4A3	1IQZ	4A1A3A2	1PCH	3A2A4A1
1COZ	5P4P1P2A3	1IR2	1P2A4A3	1PHT	4A3A2A1A5
1D0D	1A2	1J27	2A3A1A4	1PTQ	1A2
1D1M	4A5A1N2A3	1J2L	1A2A3A4	1PUG	1A2A3
1DCO	2A3A1	1J98	1A2A4A3	1PXW	1A4A2P3
1DPT	2P1A3P4	1JB0	1A3N2A4	1Q4R	2A3A1A4
1DUR	2A3A4A1	1JJ2	1A2	1Q5Y	1A2A3A4
1.00E+44	3P2A1A4	1JL3	2P1P3P4	1QGW	1A2
1EAY	2A3A1	1JOS	1A2P3	1QTN	2A3P5N1A3
1EC6	2A3A1	1JS2	2A1A3	1R8H	3A1A2

Continued...

Pdb	**Topology**	**Pdb**	**Topology**	**Pdb**	**Topology**
1EJB	2P1P3P4	1JZA	1A3A2	1RLK	2P1P4A3
1EM7	3A4P1A2	1K8K	1A2A3P4	1RMD	2A1A3A4
1EM8	4P3P1P2	1KLO	2A1A3A4	1S0Y	1P2A3A4
1ESR	1A2A3	1KQ1	5A1A2A3A4	1T3U	1A2
1EUV	2A1P5A3A4	1KRL	1A2A3	1TIG	1P2A4A3
1F9P	1A2A3	1KTE	2P1A3A4	1UB4	1A2A3A4
1FBQ	1A2N4A3	1KU6	1A2A3A4A5	1UBQ	2A1P5A3A4
1FD4	1A2A3A4	1KWA	1A5A4P3A2	1UCR	1A2A3
1FM0	2A1P5A3A4	1LN4	1P2P4A3	1UDV	2P1A4A3
1FX3	1A4A3A2	1LR7	2A1A4A3P5	1URN	4A1A3A2
1G2R	2A1A3	1LZW	2P1P3	1VIE	1A2A3A4A5
1G7C	2A3A1A4	1M8A	1A2A3	1XXA	1A2A3A4
1GCQ	1A2A3A4A5	1MFW	2A1P3P4	1YCQ	1A2
1GNK	2A3A1A4	1MHN	1A2A3A4	2HDA	4A3A2A1A5
1GNU	1A2P3P4	1MHX	3A4P1A2	5PTI	2A1A3

Appendix B

Supplementary Material

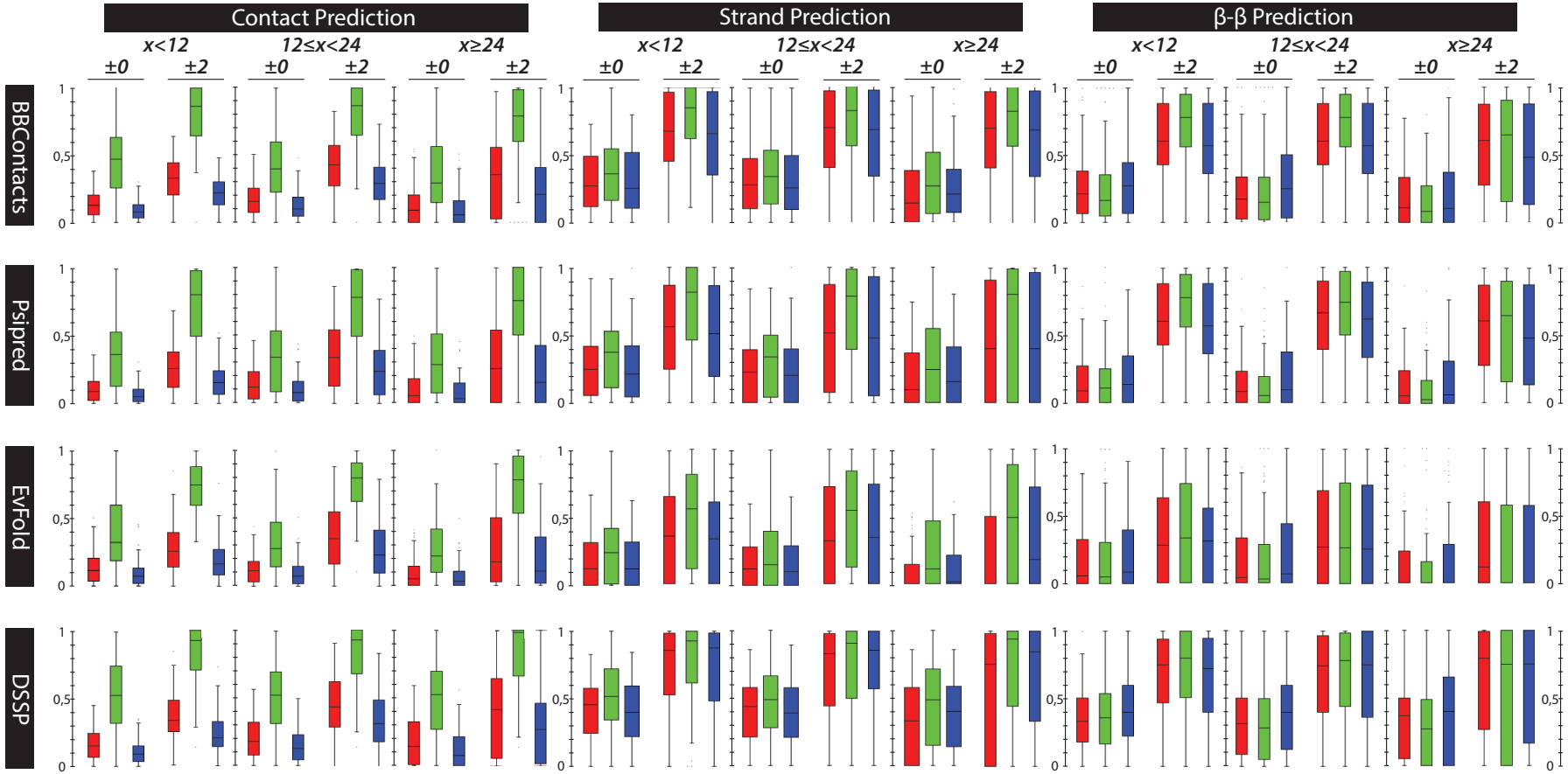

FIGURE B.1: Contact, Strand and β/β predictions performed by `efold` for the complete protein benchmark using different sequence information.

Performance is evaluated on precision (green), sensitivity (blue) and F-measure (red) of experimentally observed contacts. These metrics are reported for contacts which are 0, 12 and 24 residues apart, and when predicted contacts are within ±2 residues of an observed contact. Four different (i.e., `bbcontacts`, `PSIPRED`, `EVfold` and `DSSP`) algorithms were used to create the sequence information used by `efold` as input

Appendix C

Supplementary Material

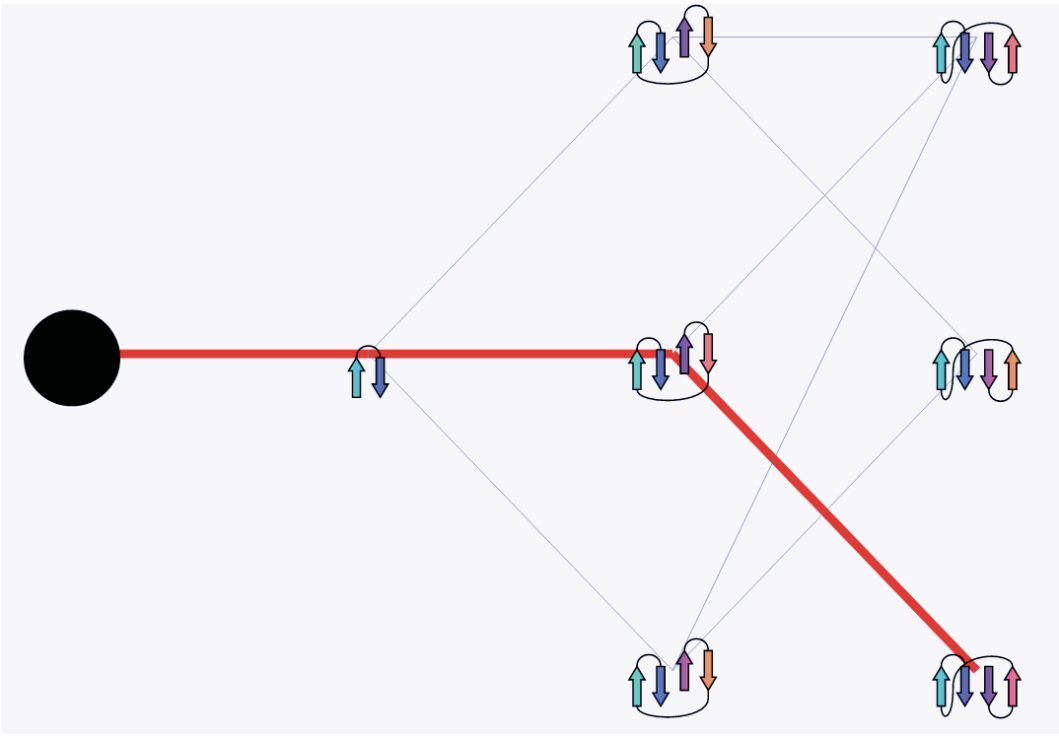

FIGURE C.1: An example of a pathway predicted by `efold` for the Protein G
The figure shows the flow network (with multiple sinks) of one individual run (out of one hundred runs) predicted by `efold` for the Protein G. Following the analogy of PF as a flux in a flow network, the weight of each reported pathway represents the probability that this specific trajectory arises from the network of folding pathways The predicted pathways are plotted using different colors and the width of a line represents the weight (a.k.a flow) of each path. If two or more paths share an edge, the visualizer will plot each path on top of the other to avoid the overlapping of lines.

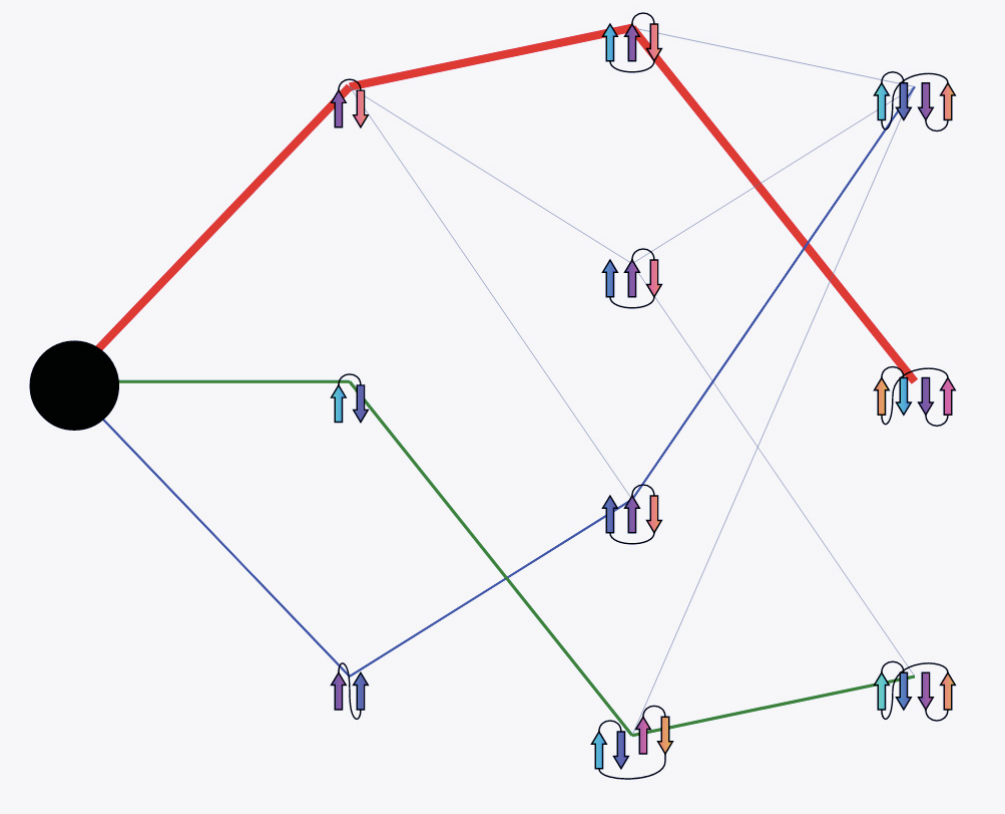

FIGURE C.2: An example of a pathway predicted by `efold` for the $N_U G_1$ mutant
The figure shows the flow network (with multiple sinks) of one individual run (out of one hundred runs) predicted by `efold` for the $N_U G_1$ mutant. Following the analogy of PF as a flux in a flow network, the weight of each reported pathway represents the probability that this specific trajectory arises from the network of folding pathways The predicted pathways are plotted using different colors and the width of a line represents the weight (a.k.a flow) of each path. If two or more paths share an edge, the visualizer will plot each path on top of the other to avoid the overlapping of lines.

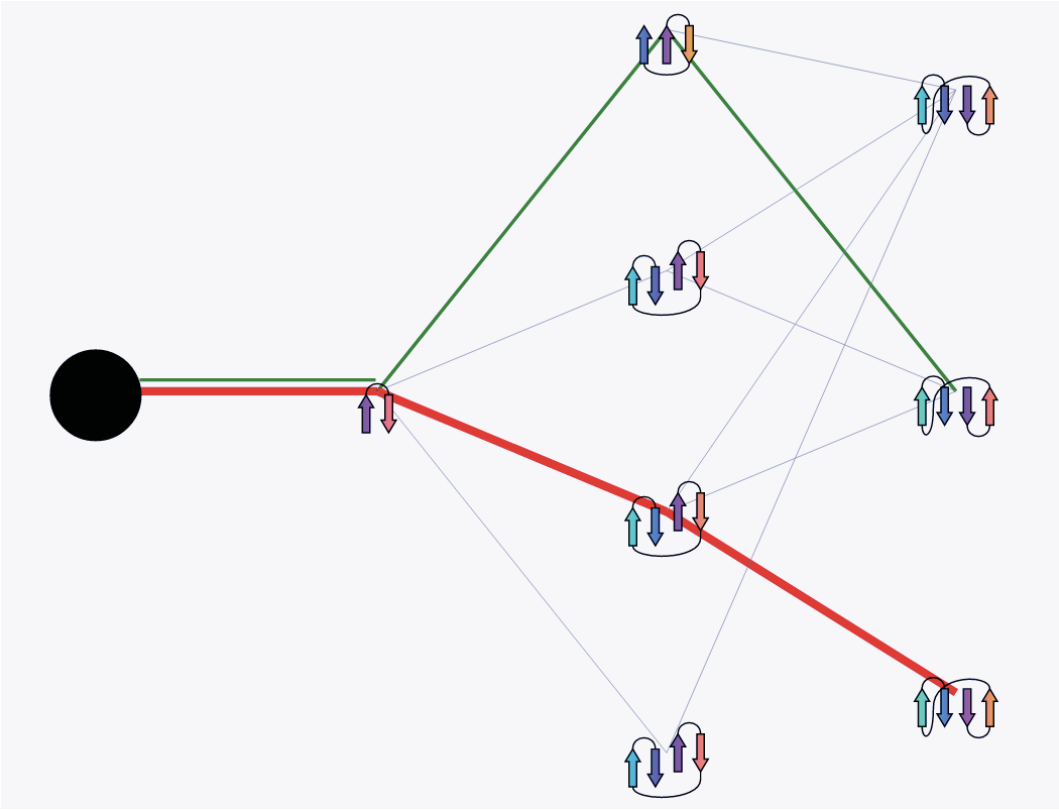

FIGURE C.3: An example of a pathway predicted by `efold` for the $N_U G_2$ mutant
The figure shows the flow network (with multiple sinks) of one individual run (out of one hundred runs) predicted by `efold` for the $N_U G_2$ mutant. Following the analogy of PF as a flux in a flow network, the weight of each reported pathway represents the probability that this specific trajectory arises from the network of folding pathways The predicted pathways are plotted using different colors and the width of a line represents the weight (a.k.a flow) of each path. If two or more paths share an edge, the visualizer will plot each path on top of the other to avoid the overlapping of lines.

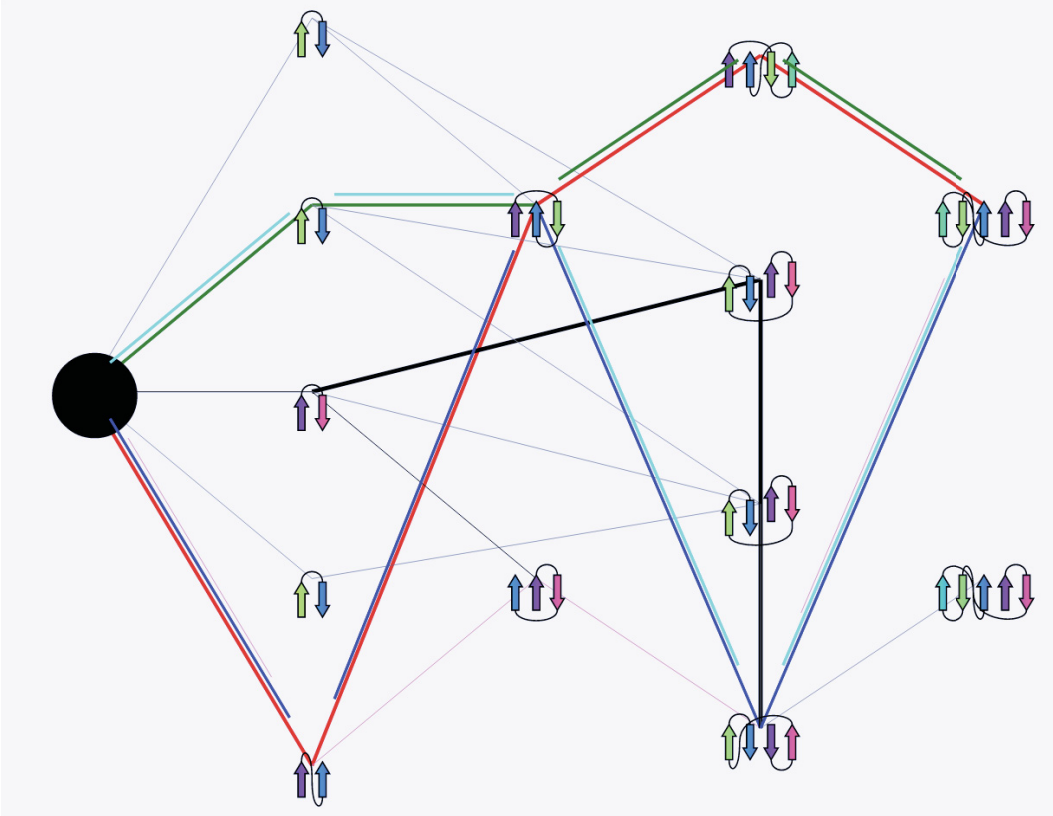

FIGURE C.4: An example of a pathway predicted by `efold` for the Ubiquitin protein
The figure shows the flow network (with multiple sinks) of one individual run (out of one hundred runs) predicted by `efold` for the Ubiquitin protein. Following the analogy of PF as a flux in a flow network, the weight of each reported pathway represents the probability that this specific trajectory arises from the network of folding pathways The predicted pathways are plotted using different colors and the width of a line represents the weight (a.k.a flow) of each path. If two or more paths share an edge, the visualizer will plot each path on top of the other to avoid the overlapping of lines.

Bibliography

[1] Daniel Kwak, Alfred Kam, David Becerra, Qikuan Zhou, Adam Hops, Eleyine Zarour, Arthur Kam, Luis Sarmenta, Mathieu Blanchette, and Jérôme Waldispühl. Open-phylo: a customizable crowd-computing platform for multiple sequence alignment. *Genome biology*, 14(10):R116, 2013.

[2] Wilson Soto and David Becerra. A multi-objective evolutionary algorithm for improving multiple sequence alignments. In *Brazilian Symposium on Bioinformatics*, pages 73–82. Springer, 2014.

[3] Francis Crick et al. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[4] A Szilagyi, J Kardos, Sz Osvath, L Barna, and P Zavodszky. Protein folding. In *Handbook of Neurochemistry and Molecular Neurobiology*, pages 303–343. Springer, 2007.

[5] Massimo Stefani. Protein misfolding and aggregation: new examples in medicine and biology of the dark side of the protein world. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1739(1):5–25, 2004.

[6] Michael Sela, Frederick H White Jr, and Christian B Anfinsen. Reductive cleavage of disulfide bridges in ribonuclease. *Science*, 125(3250):691–692, 1957.

[7] Christian B Anfinsen. Studies on the principles that govern the folding of protein chains, 1972.

[8] Cyrus Levinthal. Are there pathways for protein folding. *J. Chim. phys*, 65(1): 44–45, 1968.

[9] William J Wedemeyer and Harold A Scheraga. Protein folding: Overview of pathways. *eLS*, 2001.

[10] Ken A Dill, Hue Sun Chan, et al. From levinthal to pathways to funnels. *Nature structural biology*, 4(1):10–19, 1997.

[11] Valerie Daggett and Alan R Fersht. Is there a unifying mechanism for protein folding? *Trends in biochemical sciences*, 28(1):18–25, 2003.

[12] Nobuhiro Gō. The consistency principle in protein structure and pathways of folding. *Advances in biophysics*, 18:149–164, 1984.

[13] Ken A Dill, Sarina Bromberg, Kaizhi Yue, Hue Sun Chan, Klaus M Ftebig, David P Yee, and Paul D Thomas. Principles of protein foldinga perspective from simple exact models. *Protein Science*, 4(4):561–602, 1995.

[14] OB Ptitsyn. Molten globule and protein folding. *Advances in protein chemistry*, 47:83–229, 1995.

[15] Christopher M Dobson, Andrej Šali, and Martin Karplus. Protein folding: a perspective from theory and experiment. *Angewandte Chemie International Edition*, 37(7):868–893, 1998.

[16] Martin Karplus. How does a protein fold? *nature*, 369:19, 1994.

[17] Kalyan K Sinha and Jayant B Udgaonkar. Early events in protein folding. *Curr. Sci*, 96:1053–1070, 2009.

[18] Jaby Jacob, Bryan Krantz, Robin S Dothager, P Thiyagarajan, and Tobin R Sosnick. Early collapse is not an obligate step in protein folding. *Journal of molecular biology*, 338(2):369–382, 2004.

[19] OB Ptitsyn. Stages in the mechanism of self-organization of protein molecules. *Doklady Akademii Nauk SSSR*, 210(5):1213, 1973.

[20] Andreas Matouschek, James T Kellis, Luis Serrano, and Alan R Fersht. Mapping the transition state and pathway of protein folding by protein engineering. *Nature*, 340(6229):122–126, 1989.

[21] Luis Serrano, James T Kellis, Pauline Cann, Andreas Matouschek, and Alan R Fersht. The folding of an enzyme: Ii. substructure of barnase and the contribution of different interactions to protein stability. *Journal of molecular biology*, 224(3): 783–804, 1992.

[22] Seiji Tanaka and Harold A Scheraga. Hypothesis about the mechanism of protein folding. *Macromolecules*, 10(2):291–304, 1977.

[23] Alan R Fersht. Nucleation mechanisms in protein folding. *Current opinion in structural biology*, 7(1):3–9, 1997.

[24] Bengt Nölting and David A Agard. How general is the nucleation–condensation mechanism? *Proteins: Structure, Function, and Bioinformatics*, 73(3):754–764, 2008.

[25] Joseph D Bryngelson, Jose Nelson Onuchic, Nicholas D Socci, and Peter G Wolynes. Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins: Structure, Function, and Bioinformatics*, 21(3):167–195, 1995.

[26] Thomas Kiefhaber. Kinetic traps in lysozyme folding. *Proceedings of the National Academy of Sciences*, 92(20):9029–9033, 1995.

[27] Stefano Gianni, Carlo Travaglini-Allocatelli, Francesca Cutruzzola, Maurizio Brunori, MC Ramachandra Shastry, and Heinrich Roder. Parallel pathways in cytochrome c 551 folding. *Journal of molecular biology*, 330(5):1145–1152, 2003.

[28] Stefano Gianni, Maurizio Brunori, and Carlo Travaglini-Allocatelli. Plasticity of the protein folding landscape: Switching between on-and off-pathway intermediates. *Archives of biochemistry and biophysics*, 466(2):172–176, 2007.

[29] Caroline F Wright, Kresten Lindorff-Larsen, Lucy G Randles, and Jane Clarke. Parallel protein-unfolding pathways revealed and mapped. *Nature Structural & Molecular Biology*, 10(8):658–662, 2003.

[30] Judith Frydman. Folding of newly translated proteins in vivo: the role of molecular chaperones. *Annual review of biochemistry*, 70(1):603–647, 2001.

[31] Alice I Bartlett and Sheena E Radford. An expanding arsenal of experimental methods yields an explosion of insights into protein folding mechanisms. *Nature structural & molecular biology*, 16(6):582–588, 2009.

[32] Elena Papaleo. Integrating atomistic molecular dynamics simulations, experiments, and network analysis to study protein dynamics: strength in unity. *Frontiers in molecular biosciences*, 2, 2015.

[33] Wilfred F van Gunsteren, Jožica Dolenc, and Alan E Mark. Molecular simulation as an aid to experimentalists. *Current opinion in structural biology*, 18(2):149–153, 2008.

[34] András Fiser. Protein structure modeling in the proteomics era. *Expert Rev Proteomics*, 2004.

[35] András Fiser, Michael Feig, Charles L Brooks, and Andrej Sali. Evolution and physics in comparative protein structure modeling. *Accounts of chemical research*, 35(6):413–421, 2002.

[36] TL Blundell, BL Sibanda, MJE Sternberg, and JM Thornton. Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*, 326 (6111):347–352, 1987.

[37] Richard Bonneau and David Baker. Ab initio protein structure prediction: progress and prospects. *Annual review of biophysics and biomolecular structure*, 30(1):173–189, 2001.

[38] Jianyi Yang, Renxiang Yan, Ambrish Roy, Dong Xu, Jonathan Poisson, and Yang Zhang. The i-tasser suite: protein structure and function prediction. *Nature methods*, 12(1):7–8, 2015.

[39] Shen-Shu Sung. Helix folding simulations with various initial conformations. *Biophysical journal*, 66(6):1796, 1994.

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

[41] Kim T Simons, Charles Kooperberg, Enoch Huang, and David Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of molecular biology*, 268(1):209–225, 1997.

[42] Ram Samudrala, Yu Xia, Enoch Huang, and Michael Levitt. Ab initio protein structure prediction using a combined hierarchical approach. *Proteins: Structure, Function, and Bioinformatics*, 37(S3):194–198, 1999.

[43] James U Bowie and David Eisenberg. An evolutionary approach to folding small alpha-helical proteins that uses sequence information and an empirical guiding fitness function. *Proceedings of the National Academy of Sciences*, 91(10):4436–4440, 1994.

[44] Angel R Ortiz, Andrzej Kolinski, Piotr Rotkiewicz, Bartosz Ilkowski, and Jeffrey Skolnick. Ab initio folding of proteins using restraints derived from evolutionary information. *Proteins: Structure, Function, and Bioinformatics*, 37(S3):177–185, 1999.

[45] Philip Bradley, Kira MS Misura, and David Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871, 2005.

[46] Haiyan Liu, Marcus Elstner, Efthimios Kaxiras, Thomas Frauenheim, Jan Hermans, and Weitao Yang. Quantum mechanics simulation of protein dynamics on long timescale. *Proteins: Structure, Function, and Bioinformatics*, 44(4):484–489, 2001.

[47] Nancy M Amato and Guang Song. Using motion planning to study protein folding pathways. *Journal of Computational Biology*, 9(2):149–168, 2002.

[48] Jeffrey Skolnick, Andrzej Kolinski, et al. Monte carlo approaches to the protein folding problem. *Advances in chemical physics*, 105:203–242, 1999.

[49] Harold A Scheraga, Mey Khalili, and Adam Liwo. Protein-folding dynamics: overview of molecular simulation techniques. *Annu. Rev. Phys. Chem.*, 58:57–83, 2007.

[50] Chin-Hsien Tai, Hongjun Bai, Todd J Taylor, and Byungkook Lee. Assessment of template-free modeling in casp10 and roll. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):57–83, 2014.

[51] Ralf Jauch, Hock Chuan Yeo, Prasanna R Kolatkar, and Neil D Clarke. Assessment of casp7 structure predictions for template free targets. *Proteins: Structure, Function, and Bioinformatics*, 69(S8):57–67, 2007.

[52] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2012.

[53] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25 (17):3389–3402, 1997.

[54] Robert C Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797, 2004.

[55] Narayanan Eswar, Ben Webb, Marc A Marti-Renom, MS Madhusudhan, David Eramian, Min-yi Shen, Ursula Pieper, and Andrej Sali. Comparative protein structure modeling using modeller. *Current protocols in bioinformatics*, pages 5–6, 2006.

[56] Jonathan Greer. Comparative modeling methods: application to the family of the mammalian serine proteases. *Proteins: Structure, Function, and Bioinformatics*, 7 (4):317–334, 1990.

[57] Ron Unger, David Harel, Scot Wherland, and Joel L Sussman. A 3d building blocks approach to analyzing and predicting structure of proteins. *Proteins: Structure, Function, and Genetics*, 5(4):355–373, 1989.

[58] Timothy F Havel and Mark E Snow. A new method for building protein conformations from sequence alignments with homologues of known structure. *Journal of molecular biology*, 217(1):1–7, 1991.

[59] Donald Petrey, Zhexin Xiang, Christopher L Tang, Lei Xie, Marina Gimpelev, Therese Mitros, Cinque S Soto, Sharon Goldsmith-Fischman, Andrew Kernytsky, Avner Schlessinger, et al. Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. *Proteins: Structure, Function, and Bioinformatics*, 53(S6):430–435, 2003.

[60] David T Jones, WR Taylort, and Janet M Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.

[61] Andrew E Torda, James B Procter, and Thomas Huber. Wurst: a protein threading server with a structural scoring function, sequence profiles and optimized substitution matrices. *Nucleic acids research*, 32(suppl 2):W532–W535, 2004.

[62] Ambrish Roy, Alper Kucukural, and Yang Zhang. I-tasser: a unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4): 725–738, 2010.

[63] Yang Zhang and Jeffrey Skolnick. The protein structure prediction problem could be solved using the current pdb library. *Proceedings of the National Academy of Sciences of the United States of America*, 102(4):1029–1034, 2005.

[64] Roman A Laskowski, Malcolm W MacArthur, David S Moss, and Janet M Thornton. Procheck: a program to check the stereochemical quality of protein structures. *Journal of applied crystallography*, 26(2):283–291, 1993.

[65] Manfred J Sippl. Recognition of errors in three-dimensional structures of proteins. *Proteins: Structure, Function, and Genetics*, 17(4):355–362, 1993.

[66] Roland Liithy, JU Bowie, and D Eisenberg. Assessment of protein models with three-dimensional profiles. *Nature*, 356(6364):83–85, 1992.

[67] Cyrus Chothia and Arthur M Lesk. The relation between the divergence of sequence and structure in proteins. *The EMBO journal*, 5(4):823, 1986.

[68] Cyrus Chothia. Proteins. one thousand families for the molecular biologist. *Nature*, 357(6379):543, 1992.

[69] Daniel A Keedy, Christopher J Williams, Jeffrey J Headd, W Bryan Arendall, Vincent B Chen, Gary J Kapral, Robert A Gillespie, Jeremy N Block, Adam Zemla, David C Richardson, et al. The other 90% of the protein: Assessment beyond the cαs for casp8 template-based and high-accuracy models. *Proteins: Structure, Function, and Bioinformatics*, 77(S9):29–49, 2009.

[70] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (casp)round x. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):1–6, 2014.

[71] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (hp) model is np-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[72] Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding. *Journal of computational biology*, 5(3):423–465, 1998.

[73] Peter G Wolynes, Jose N Onuchic, D Thirumalai, et al. Navigating the folding routes. *SCIENCE-NEW YORK THEN WASHINGTON-*, pages 1619–1619, 1995.

[74] Sergio Raul Duarte Torres, David Camilo Becerra Romero, Luis Fernando Nino Vasquez, and Yoan Jose Pinzon Ardila. A novel ab-initio genetic-based approach for protein folding prediction. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 393–400. ACM, 2007.

[75] Herman JC Berendsen. A glimpse of the holy grail? *Science*, 282(5389):642–643, 1998.

[76] Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

[77] Yong Duan and Peter A. Kollman. Computational protein folding: From lattice to all-atom. *IBM Systems Journal*, 40(2):297–309, 2001.

[78] Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature Structural & Molecular Biology*, 9(9):646–652, 2002.

[79] David Becerra, Angelica Sandoval, Daniel Restrepo-Montoya, and F Nino Luis. A parallel multi-objective ab initio approach for protein structure prediction. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 137–141. IEEE, 2010.

[80] Ye Ding and Charles E Lawrence. A statistical sampling algorithm for rna secondary structure prediction. *Nucleic acids research*, 31(24):7280–7301, 2003.

[81] Kerson Huang. *Lectures on statistical physics and protein folding*. World Scientific, 2005.

[82] Ming-Hong Hao and Harold A Scheraga. Statistical thermodynamics of protein folding: Comparison of a mean-field theory with monte carlo simulations. *The Journal of chemical physics*, 102(3):1334–1348, 1995.

[83] Stefano Gianni, Ylva Ivarsson, Per Jemth, Maurizio Brunori, and Carlo Travaglini-Allocatelli. Identification and characterization of protein folding intermediates. *Biophysical chemistry*, 128(2):105–113, 2007.

[84] Alan R Fersht and Satoshi Sato. ϕ-value analysis and the nature of protein-folding transition states. *Proceedings of the National Academy of Sciences of the United States of America*, 101(21):7976–7981, 2004.

[85] Ted A Laurence, Xiangxu Kong, Marcus Jäger, and Shimon Weiss. Probing structural heterogeneities and fluctuations of nucleic acids and denatured proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 102 (48):17348–17353, 2005.

[86] Robert H Callender, R Brian Dyer, Rudolf Gilmanshin, and William H Woodruff. Fast events in protein folding: the time evolution of primary processes. *Annual review of physical chemistry*, 49(1):173–202, 1998.

[87] Aashish N Adhikari, Karl F Freed, and Tobin R Sosnick. De novo prediction of protein folding pathways and structure using the principle of sequential stabilization. *Proceedings of the National Academy of Sciences*, 109(43):17442–17447, 2012.

[88] Mark Moll, David Schwarz, and Lydia E Kavraki. Roadmap methods for protein folding. *Protein Structure Prediction*, pages 219–239, 2008.

[89] Nancy M Amato, Ken A Dill, and Guang Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *Journal of Computational Biology*, 10(3-4):239–255, 2003.

[90] Christopher Bystroff and Yu Shao. Fully automated ab initio protein structure prediction using i-sites, hmmstr and rosetta. *Bioinformatics*, 18(suppl 1):S54–S61, 2002.

[91] Julian GB Northey, Ariel A Di Nardo, and Alan R Davidson. Hydrophobic core packing in the sh3 domain folding transition state. *Nature Structural & Molecular Biology*, 9(2):126–130, 2002.

[92] Mohammed J Zaki, Vinay Nadimpally, Deb Bardhan, and Chris Bystroff. Predicting protein folding pathways. *Bioinformatics*, 20(suppl 1):i386–i393, 2004.

[93] Vibin Ramakrishnan, Sai Praveen Srinivasan, Saeed M Salem, Suzanne J Matthews, Wilfredo Colón, Mohammed Zaki, and Christopher Bystroff. Geofold: Topology-based protein unfolding pathways capture the effects of engineered disulfides on kinetic stability. *Proteins: Structure, Function, and Bioinformatics*, 80(3):920–934, 2012.

[94] Clare-Louise Towse and Valerie Daggett. Modeling protein folding pathways. *Reviews in Computational Chemistry*, 28:87–135, 2015.

[95] Christian B Anfinsen, Robert R Redfield, Warren L Choate, Juanita Page, and William R Carroll. Studies on the gross structure, cross-linkages, and terminal sequences in ribonuclease. *Journal of Biological Chemistry*, 207(1):201–210, 1954.

[96] Christian B Anfinsen et al. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[97] Stefan M Larson, Christopher D Snow, Michael Shirts, and Vijay S Pande. Folding@ home and genome@ home: Using distributed computing to tackle previously intractable problems in computational biology. *arXiv preprint arXiv:0901.0866*, 2009.

[98] Thomas J Lane, Diwakar Shukla, Kyle A Beauchamp, and Vijay S Pande. To milliseconds and beyond: challenges in the simulation of protein folding. *Current opinion in structural biology*, 23(1):58–65, 2013.

[99] S Gnanakaran, Hugh Nymeyer, John Portman, Kevin Y Sanbonmatsu, and Angel E Garcıa. Peptide folding simulations. *Current opinion in structural biology*, 13(2): 168–174, 2003.

[100] S Banu Ozkan, G Albert Wu, John D Chodera, and Ken A Dill. Protein folding by zipping and assembly. *Proceedings of the National Academy of Sciences*, 104(29): 11987–11992, 2007.

[101] Yong Duan and Peter A Kollman. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, 282(5389): 740–744, 1998.

[102] Morten Källberg, Gohar Margaryan, Sheng Wang, Jianzhu Ma, and Jinbo Xu. Raptorx server: a resource for template-based protein structure modeling. *Methods Mol Biol*, 1137:17–27, 2014. doi: 10.1007/978-1-4939-0366-5_2.

[103] Jeffrey K Weber and Vijay S Pande. Characterization and rapid sampling of protein folding markov state model topologies. *Journal of chemical theory and computation*, 7(10):3405–3411, 2011.

[104] John E Stone, David J Hardy, Ivan S Ufimtsev, and Klaus Schulten. Gpu-accelerated molecular modeling coming of age. *Journal of Molecular Graphics and Modelling*, 29(2):116–125, 2010.

[105] Vincent A Voelz, Gregory R Bowman, Kyle Beauchamp, and Vijay S Pande. Molecular simulation of ab initio protein folding for a millisecond folder ntl9 (1- 39). *Journal of the American Chemical Society*, 132(5):1526–1528, 2010.

[106] David E Shaw, Martin M Deneroff, Ron O Dror, Jeffrey S Kuskin, Richard H Larson, John K Salmon, Cliff Young, Brannon Batson, Kevin J Bowers, Jack C Chao, et al. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM*, 51(7):91–97, 2008.

[107] Valentina Tozzini. Coarse-grained models for proteins. *Current opinion in structural biology*, 15(2):144–150, 2005.

[108] Faruck Morcos, Terence Hwa, José N Onuchic, and Martin Weigt. Direct coupling analysis for protein contact prediction. *Protein Structure Prediction*, pages 55–70, 2014.

[109] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d structure computed from evolutionary sequence variation. *PloS one*, 6(12):e28766, 2011.

[110] David de Juan, Florencio Pazos, and Alfonso Valencia. Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 14(4):249–261, 2013.

[111] Joanna I Sułkowska, Faruck Morcos, Martin Weigt, Terence Hwa, and José N Onuchic. Genomics-aided structure prediction. *Proceedings of the National Academy of Sciences*, 109(26):10340–10345, 2012.

[112] Timothy Nugent and David T Jones. Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis. *Proceedings of the National Academy of Sciences*, 109(24):E1540–E1547, 2012.

[113] Bohdan Monastyrskyy, Daniel D'Andrea, Krzysztof Fidelis, Anna Tramontano, and Andriy Kryshtafovych. Evaluation of residue–residue contact prediction in casp10. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):138–153, 2014.

[114] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, TN Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[115] R Elber and M Karplus. Multiple conformational states of proteins: a molecular dynamics analysis of myoglobin. *Science*, 235(4786):318–321, 1987.

[116] Jérôme Waldispühl, Charles W O'Donnell, Srinivas Devadas, Peter Clote, and Bonnie Berger. Modeling ensembles of transmembrane β-barrel proteins. *Proteins: Structure, Function, and Bioinformatics*, 71(3):1097–1112, 2008.

[117] Charles William O'Donnell. *Ensemble modeling of [beta]-sheet proteins*. PhD thesis, Massachusetts Institute of Technology, 2011.

[118] JS McCaskill. The equilibrium partition function and base pair binding probabilities for rna secondary structure. *Biopolymers*, 29:1105–1119, 1990.

[119] J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for rna secondary structure. *Biopolymers*, 29(6-7):1105–1119, 2004.

[120] Jérôme Waldispühl, Charles W O'Donnell, Sebastian Will, Srinivas Devadas, Rolf Backofen, and Bonnie Berger. Simultaneous alignment and folding of protein sequences. *Journal of Computational Biology*, 21(7):477–491, 2014.

[121] Solomon Shenker, Charles W O'Donnell, Srinivas Devadas, Bonnie Berger, and Jérôme Waldispühl. Efficient traversal of beta-sheet protein folding pathways using ensemble models. *Journal of Computational Biology*, 18(11):1635–1647, 2011.

[122] Charles W O'Donnell, Jérôme Waldispühl, Mieszko Lis, Randal Halfmann, Srinivas Devadas, Susan Lindquist, and Bonnie Berger. A method for probing the mutational landscape of amyloid structure. *Bioinformatics*, 27(13):i34–i42, 2011.

[123] Christopher M Dobson. Protein folding and misfolding. *Nature*, 426(6968):884–90, Dec 2003. doi: 10.1038/nature02261.

[124] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–10, Nov 2002. doi: 10.1038/nature01254.

[125] Rebecca Nelson and David Eisenberg. Recent atomic models of amyloid fibril structure. *Curr Opin Struct Biol*, 16(2):260–5, Apr 2006. doi: 10.1016/j.sbi.2006.03.007.

[126] Fabrizio Chiti and Christopher M Dobson. Protein misfolding, functional amyloid, and human disease. *Annu Rev Biochem*, 75:333–66, 2006. doi: 10.1146/annurev.biochem.75.101304.123901.

[127] Ronald D Hills, Jr and Charles L Brooks, 3rd. Hydrophobic cooperativity as a mechanism for amyloid nucleation. *J Mol Biol*, 368(3):894–901, May 2007. doi: 10.1016/j.jmb.2007.02.043.

[128] Mohamed Raef Smaoui, Frédéric Poitevin, Marc Delarue, Patrice Koehl, Henri Orland, and Jérôme Waldispühl. Computational assembly of polymorphic amyloid fibrils reveals stable aggregates. *Biophys J*, 104(3):683–93, Feb 2013. doi: 10.1016/j.bpj.2012.12.037.

[129] J. Brady and M. Karplus. Configuration entropy of the alanine dipeptide in vacuum and in solution - a molecular-dynamics study. *J. Am. Chem. Soc.*, 107(21):6103–6105, 1985. ISSN 0002-7863.

[130] M Schaefer, C Bartels, and M Karplus. Solution conformations and thermodynamics of structured peptides: molecular dynamics simulation with an implicit solvation model. *J Mol Biol*, 284(3):835–48, Dec 1998. doi: 10.1006/jmbi.1998.2172.

[131] Shlomit R Edinger, Christian Cortis, Peter S Shenkin, and Richard A Friesner. Solvation free energies of peptides: Comparison of approximate continuum solvation models with accurate solution of the poisson-boltzmann equation. *The Journal of Physical Chemistry B*, 101(7):1190–1197, 1997.

[132] Patrice Koehl and Marc Delarue. Aquasol: An efficient solver for the dipolar poisson-boltzmann-langevin equation. *J Chem Phys*, 132(6):064101, Feb 2010. doi: 10.1063/1.3298862.

[133] Donald Kennedy and Colin Norman. What don't we know? *Science*, 309(5731):75–75, 2005.

[134] Jui-Yoa Chang. Distinct folding pathways of two homologous disulfide proteins: bovine pancreatic trypsin inhibitor and tick anticoagulant peptide. *Antioxidants & redox signaling*, 14(1):127–135, 2011.

[135] Scott K Silverman, Michael L Deras, Sarah A Woodson, Stephen A Scaringe, and Thomas R Cech. Multiple folding pathways for the p4-p6 rna domain. *Biochemistry*, 39(40):12465–12475, 2000.

[136] Ryo Kitahara and Kazuyuki Akasaka. Close identity of a pressure-stabilized intermediate with a kinetic intermediate in protein folding. *Proceedings of the National Academy of Sciences*, 100(6):3167–3172, 2003.

[137] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of computational biology*, 1(4):337–348, 1994.

[138] Isaac Elias. Settling the intractability of multiple alignment. *Journal of Computational Biology*, 13(7):1323–1339, 2006.

[139] Iain M Wallace, Gordon Blackshields, and Desmond G Higgins. Multiple sequence alignments. *Current opinion in structural biology*, 15(3):261–266, 2005.

[140] Heng Li and Nils Homer. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in bioinformatics*, 11(5):473–483, 2010.

[141] Mathieu Blanchette. Computation and analysis of genomic multi-sequence alignments. *Annu. Rev. Genomics Hum. Genet.*, 8:193–213, 2007.

[142] Noah M Daniels, Shilpa Nadimpalli, and Lenore J Cowen. Formatt: Correcting protein multiple structural alignments by incorporating sequence alignment. *BMC bioinformatics*, 13(1):259, 2012.

[143] Debora S Marks, Thomas A Hopf, and Chris Sander. Protein structure prediction from sequence variation. *Nature biotechnology*, 30(11):1072–1080, 2012.

[144] L. Wang, J. Leebens-Mack, et al. The impact of multiple protein sequence alignment on phylogenetic estimation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(4):1108–1119, 2011.

[145] Noah M Daniels, Raghavendra Hosur, Bonnie Berger, and Lenore J Cowen. Smurflite: combining simplified markov random fields with simulated evolution improves remote homology detection for beta-structural proteins into the twilight zone. *Bioinformatics*, 28(9):1216–1222, 2012.

[146] Liam J McGuffin, Kevin Bryson, and David T Jones. The psipred protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.

[147] Gearóid Fox, Fabian Sievers, and Desmond G Higgins. Using de novo protein structure predictions to measure the quality of very large multiple sequence alignments. *Bioinformatics*, page btv592, 2015.

[148] Da-Fei Feng and Russell F Doolittle. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of molecular evolution*, 25(4):351–360, 1987.

[149] Tobias Rausch, Anne-Katrin Emde, David Weese, Andreas Döring, Cedric Notredame, and Knut Reinert. Segment-based multiple sequence alignment. *Bioinformatics*, 24(16):i187–i192, 2008.

[150] J D. Thompson et al. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.

[151] C B. Do, M SP. Mahabhashyam, et al. Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, 15(2):330–340, 2005.

[152] Ari Löytynoja and Nick Goldman. webprank: a phylogeny-aware multiple sequence aligner with interactive alignment browser. *BMC bioinformatics*, 11(1):579, 2010.

[153] K. Katoh et al. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic acids research*, 30(14):3059–3066, 2002.

[154] Mathieu Blanchette, W James Kent, Cathy Riemer, Laura Elnitski, Arian FA Smit, Krishna M Roskin, Robert Baertsch, Kate Rosenbloom, Hiram Clawson, Eric D Green, et al. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome research*, 14(4):708–715, 2004.

[155] Orla O'Sullivan, Karsten Suhre, Chantal Abergel, Desmond G Higgins, and Cedric Notredame. 3dcoffee: combining protein sequences and structures within multiple sequence alignments. *Journal of molecular biology*, 340(2):385–395, 2004.

[156] Paulo Nuin, Zhouzhi Wang, and Elisabeth Tillier. The accuracy of several multiple sequence alignment programs for proteins. *BMC bioinformatics*, 7(1):471, 2006.

[157] I M. Wallace, O. O'Sullivan, et al. M-coffee: combining multiple sequence alignment methods with t-coffee. *Nucleic acids research*, 34(6):1692–1699, 2006.

[158] F. Ortuno, J P. Florido, et al. Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on nsga-ii. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.

[159] F. Ortuño, O. Valenzuela, et al. Predicting the accuracy of multiple sequence alignment algorithms by using computational intelligent techniques. *Nucleic acids research*, 41(1):e26–e26, 2013.

[160] Francisco M Ortuño et al. Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns. *Bioinformatics*, page btt360, 2013.

[161] Dannie Durand and Martin Farach-Colton. On the design of optimization criteria for multiple sequence alignment. In *Biological Evolution and Statistical Physics*, pages 22–36. Springer, 2002.

[162] Iva Kostadinova. Citizen science-the new helping hand for scientists, 2011.

[163] Vickie Curtis. Online citizen science games: opportunities for the biological sciences. *Applied & translational genomics*, 3(4):90–94, 2014.

[164] Benjamin M Good and Andrew I Su. Crowdsourcing for bioinformatics. *Bioinformatics*, page btt333, 2013.

[165] Alexander Kawrykow, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Luis Sarmenta, Mathieu Blanchette, Jérôme Waldispühl,

et al. Phylo: a citizen science approach for improving multiple sequence alignment. *PloS one*, 7(3):e31362, 2012.

[166] Bjarne Knudsen. Optimal multiple parsimony alignment with affine gap cost using a phylogenetic tree. In *International Workshop on Algorithms in Bioinformatics*, pages 433–446. Springer, 2003.

[167] Walter M Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971.

[168] Abdoulaye Banire Diallo, Vladimir Makarenkov, and Mathieu Blanchette. Ancestors 1.0: a web server for ancestral sequence reconstruction. *Bioinformatics*, 26(1): 130–131, 2010.

[169] Osnat Penn, Eyal Privman, Haim Ashkenazy, Giddy Landan, Dan Graur, and Tal Pupko. Guidance: a web server for assessing alignment confidence scores. *Nucleic acids research*, 38(suppl 2):W23–W28, 2010.

[170] Andrew M Waterhouse, James B Procter, David MA Martin, Michèle Clamp, and Geoffrey J Barton. Jalview version 2?a multiple sequence alignment editor and analysis workbench. *Bioinformatics*, 25(9):1189–1191, 2009.

[171] F. Sievers, A. Wilm, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, 7(1), 2011.

[172] C. Notredame et al. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000.

[173] J D. Thompson et al. Balibase 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins: Structure, Function, and Bioinformatics*, 61(1): 127–136, 2005.

[174] Benjamin P Blackburne and Simon Whelan. Measuring the distance between multiple sequence alignments. *Bioinformatics*, 28(4):495–502, 2012.

[175] J. Branke, K. Deb, et al. Finding knees in multi-objective optimization. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 722–731. Springer, 2004.

[176] Eckart Zitzler et al. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization*, pages 862–876. Springer, 2007.

[177] Sitao Wu, Andras Szilagyi, and Yang Zhang. Improving protein structure prediction using multiple sequence-based contact predictions. *Structure*, 19(8):1182–1191, 2011.

[178] Sergey Ovchinnikov, Lisa Kinch, Hahnbeom Park, Yuxing Liao, Jimin Pei, David E Kim, Hetunandan Kamisetty, Nick V Grishin, and David Baker. Large-scale determination of previously unsolved protein structures using evolutionary information. *Elife*, 4:e09248, 2015.

[179] David E Kim, Frank DiMaio, Ray Yu-Ruei Wang, Yifan Song, and David Baker. One contact for every twelve residues allows robust and accurate topology-level protein structure modeling. *Proteins: Structure, Function, and Bioinformatics*, 82 (S2):208–218, 2014.

[180] Fabian Birzele and Stefan Kramer. A new representation for protein secondary structure prediction based on frequent patterns. *Bioinformatics*, 22(21):2628–2634, 2006.

[181] David Becerra, Diana Vanegas, Giovanni Cantor, and Luis Niño. An association rule based approach for biological sequence feature classification. In *2009 IEEE Congress on Evolutionary Computation*, pages 3111–3118. IEEE, 2009.

[182] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22 (12):2577–2637, 1983.

[183] Jessica Andreani and Johannes Söding. bbcontacts: prediction of β-strand pairing from direct coupling patterns. *Bioinformatics*, page btv041, 2015.

[184] David Shortle, Kim T Simons, and David Baker. Clustering of low-energy conformations near the native structures of small proteins. *Proceedings of the National Academy of Sciences*, 95(19):11158–11162, 1998.

[185] Simon Olsson, Beat Rolf Vgeli, Andrea Cavalli, Wouter Boomsma, Jesper Ferkinghoff-Borg, Kresten Lindorff-Larsen, and Thomas Hamelryck. Probabilistic determination of native state ensembles of proteins. *Journal of chemical theory and computation*, 10(8):3484–3491, 2014.

[186] Vijay S Pande, Alexander Yu Grosberg, Toyoichi Tanaka, and Daniel S Rokhsar. Pathways for protein folding: is a new view needed? *Current opinion in structural biology*, 8(1):68–79, 1998.

[187] Joan L Arolas, Francesc X Aviles, Jui-Yoa Chang, and Salvador Ventura. Folding of small disulfide-rich proteins: clarifying the puzzle. *Trends in biochemical sciences*, 31(5):292–301, 2006.

[188] Sanzo Miyazawa and Robert L Jernigan. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18(3):534–552, 1985.

[189] Manfred J Sippl. Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins. *Journal of molecular biology*, 213(4):859–883, 1990.

[190] Phil Bradley, Lenore Cowen, Matthew Menke, Jonathan King, and Bonnie Berger. Betawrap: Successful prediction of parallel β-helices from primary sequence reveals an association with many microbial pathogens. *Proceedings of the National Academy of Sciences*, 98(26):14819–14824, 2001.

[191] J. Waldispühl, B. Berger, P. Clote, and J.M. Steyaert. Predicting transmembrane β-barrels and interstrand residue interactions from sequence. *PROTEINS: Structure, Function, and Bioinformatics*, 65(1):61–74, 2006.

[192] Adam Zemla, Česlovas Venclovas, Krzysztof Fidelis, and Burkhard Rost. A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Bioinformatics*, 34(2): 220–223, 1999.

[193] Vincent Moulton, Michael Zuker, Michael Steel, Robin Pointon, and David Penny. Metrics on rna secondary structures. *Journal of Computational Biology*, 7(1-2): 277–292, 2000.

[194] Michael T Wolfinger, W Andreas Svrcek-Seiler, Christoph Flamm, Ivo L Hofacker, and Peter F Stadler. Efficient computation of rna folding dynamics. *Journal of Physics A: Mathematical and General*, 37(17):4731, 2004.

[195] Christoph Flamm, Walter Fontana, Ivo L Hofacker, and Peter Schuster. Rna folding at elementary step resolution. *Rna*, 6(3):325–338, 2000.

[196] Alexander P Gultyaev, FHD Van Batenburg, and Cornelis WA Pleij. The computer simulation of rna folding pathways using a genetic algorithm. *Journal of molecular biology*, 250(1):37–51, 1995.

[197] Peter E Leopold, Mauricio Montal, and José N Onuchic. Protein folding funnels: a kinetic approach to the sequence-structure relationship. *Proceedings of the National Academy of Sciences*, 89(18):8721–8725, 1992.

[198] S Banu Ozkan, Ken A Dill, and Ivet Bahar. Computing the transition state populations in simple protein models. *Biopolymers*, 68(1):35–46, 2003.

[199] CW Gardiner. Handbook of stochastic methods for physics, chemistry and the natural sciences. *Applied Optics*, 25:3145, 1986.

[200] Kyozi Kawasaki. Diffusion constants near the critical point for time-dependent ising models. i. *Physical Review*, 145(1):224, 1966.

[201] Michael Wolfinger. *The energy landscape of RNA folding*. na, 2001.

[202] Jean-Pierre Hansen and Ian R McDonald. *Theory of simple liquids*. Elsevier, 1990.

[203] Volker A Eyrich, Daron M Standley, Anthony K Felts, and Richard A Friesner. Protein tertiary structure prediction using a branch and bound algorithm. *Proteins: Structure, Function, and Bioinformatics*, 35(1):41–57, 1999.

[204] Mehmet Serkan Apaydin, Amit Pal Singh, Douglas L Brutlag, and J-C Latombe. Capturing molecular energy landscapes with probabilistic conformational roadmaps. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 932–939. IEEE, 2001.

[205] Dmitry N Ivankov and Alexei V Finkelstein. Theoretical study of a landscape of protein folding- unfolding pathways. folding rates at midtransition. *Biochemistry*, 40(33):9957–9961, 2001.

[206] Dmitry N Ivankov and Alexei V Finkelstein. Protein folding as flow across a network of folding- unfolding pathways. 1. the mid-transition case. *The Journal of Physical Chemistry B*, 114(23):7920–7929, 2010.

[207] Dmitry N Ivankov and Alexei V Finkelstein. Protein folding as flow across a network of folding- unfolding pathways. 2. the ?in-water? case. *The Journal of Physical Chemistry B*, 114(23):7930–7934, 2010.

[208] Akira R Kinjo and Ken Nishikawa. Recoverable one-dimensional encoding of three-dimensional protein structures. *Bioinformatics*, 21(10):2167–2170, 2005.

[209] Rajanish Giri, Angela Morrone, Carlo Travaglini-Allocatelli, Per Jemth, Maurizio Brunori, and Stefano Gianni. Folding pathways of proteins with increasing degree of sequence identities but different structure and function. *Proceedings of the National Academy of Sciences*, 109(44):17772–17776, 2012.

[210] Patrick A Alexander, Yanan He, Yihong Chen, John Orban, and Philip N Bryan. The design and characterization of two proteins with 88% sequence identity but different structure and function. *Proceedings of the National Academy of Sciences*, 104(29):11963–11968, 2007.

[211] Yanan He, Yihong Chen, Patrick Alexander, Philip N Bryan, and John Orban. Nmr structures of two designed proteins with high sequence identity but different fold and function. *Proceedings of the National Academy of Sciences*, 105(38):14412–14417, 2008.

[212] Sehat Nauli, Brian Kuhlman, and David Baker. Computer-based redesign of a protein folding pathway. *Nature structural & molecular biology*, 8(7):602–605, 2001.

[213] Sehat Nauli, Brian Kuhlman, Isolde Le Trong, Ronald E Stenkamp, David Teller, and David Baker. Crystal structures and increased stabilization of the protein g variants with switched folding pathways nug1 and nug2. *Protein science*, 11(12): 2924–2931, 2002.

[214] Robert D Finn. Pfam: the protein families database. *Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics*, 2012.

[215] S. Kmiecik and A. Kolinski. Folding pathway of the b1 domain of protein g explored by multiscale modeling. *Biophysical journal*, 94(3):726–736, 2008.

[216] I.A. Hubner, J. Shimada, and E.I. Shakhnovich. Commitment and nucleation in the protein g transition state. *Journal of molecular biology*, 336(3):745–761, 2004.

[217] A.M. Gronenborn, D.R. Filpula, N.Z. Essig, A. Achari, M. Whitlow, P.T. Wingfield, GM Clore, et al. A novel, highly stable fold of the immunoglobulin binding domain of streptococcal protein g. *Science(Washington)*, 253(5020):657–661, 1991.

[218] F.J. Blanco, G. Rivas, and L. Serrano. A short linear peptide that folds into a native stable β-hairpin in aqueous solution. *Nature Structural & Molecular Biology*, 1(9):584–590, 1994.

[219] J. Kuszewski, G.M. Clore, and A.M. Gronenborn. Fast folding of a prototypic polypeptide: the immunoglobulin binding domain of streptococcal protein g. *Protein Science*, 3(11):1945–1952, 2008.

[220] Sophie E Jackson. Ubiquitin: a small protein folding paradigm. *Organic & biomolecular chemistry*, 4(10):1845–1853, 2006.

[221] Ajazul Hamid Wani and Jayant B Udgaonkar. Revealing a concealed intermediate that forms after the rate-limiting step of refolding of the sh3 domain of pi3 kinase. *Journal of molecular biology*, 387(2):348–362, 2009.

[222] Isaac A Hubner, Katherine A Edmonds, and Eugene I Shakhnovich. Nucleation and the transition state of the sh3 domain. *Journal of molecular biology*, 349(2): 424–434, 2005.

[223] Feng Ding, Weihua Guo, Nikolay V Dokholyan, Eugene I Shakhnovich, and Joan-Emma Shea. Reconstruction of the src-sh3 protein domain transition state ensemble using multiscale molecular dynamics simulations. *Journal of molecular biology*, 350 (5):1035–1050, 2005.

[224] Viara P Grantcharova, David S Riddle, and David Baker. Long-range order in the src sh3 folding transition state. *Proceedings of the National Academy of Sciences*, 97(13):7084–7089, 2000.

[225] Jose C Martínez and Luis Serrano. The folding transition state between sh3 domains is conformationally restricted and evolutionarily conserved. *Nature Structural & Molecular Biology*, 6(11):1010–1016, 1999.

[226] G Chelvanayagam and P Argos. Prediction of protein folding pathways: Bovine pancreatic trypsin inhibitor. *Cytotechnology*, 11(1):S67–S71, 1993.

[227] Christian Kambach, Stefan Walke, Robert Young, Johanna M Avis, Eric de la Fortelle, Veronica A Raker, Reinhard Lührmann, Jade Li, and Kiyoshi Nagai. Crystal structures of two sm protein complexes and their implications for the assembly of the spliceosomal snrnps. *Cell*, 96(3):375–387, 1999.

[228] Jianlin Cheng and Pierre Baldi. Three-stage prediction of protein β-sheets by neural networks, alignments and graph algorithms. *Bioinformatics*, 21(suppl 1):i75–i84, 2005.

[229] S Walter Englander and Leland Mayne. The nature of protein folding pathways. *Proceedings of the National Academy of Sciences*, 111(45):15873–15880, 2014.

[230] Jun Shimada and Eugene I Shakhnovich. The ensemble folding kinetics of protein g from an all-atom monte carlo simulation. *Proceedings of the National Academy of Sciences*, 99(17):11175–11180, 2002.

[231] Bryan A Krantz, Robin S Dothager, and Tobin R Sosnick. Discerning the structure and energy of multiple transition states in protein folding using ψ-analysis. *Journal of molecular biology*, 337(2):463–475, 2004.

[232] Stefano Piana, Kresten Lindorff-Larsen, and David E Shaw. Atomic-level description of ubiquitin folding. *Proceedings of the National Academy of Sciences*, 110(15): 5915–5920, 2013.

[233] Tobin R Sosnick, Robin S Dothager, and Bryan A Krantz. Differences in the folding transition state of ubiquitin indicated by φ and ψ analyses. *Proceedings of the National Academy of Sciences of the United States of America*, 101(50):17377–17382, 2004.

[234] Heather M Went and Sophie E Jackson. Ubiquitin folds through a highly polarized transition state. *Protein Engineering Design and Selection*, 18(5):229–237, 2005.

[235] Péter Várnai, Christopher M Dobson, and Michele Vendruscolo. Determination of the transition state ensemble for the folding of ubiquitin from a combination of φ and ψ analyses. *Journal of molecular biology*, 377(2):575–588, 2008.

[236] Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[237] Philippe Derreumaux. Role of supersecondary structural elements in protein g folding. *The Journal of chemical physics*, 119(9):4940–4944, 2003.

[238] Lydia Tapia, Xinyu Tang, Shawna Thomas, and Nancy M Amato. Kinetics analysis methods for approximate folding landscapes. *Bioinformatics*, 23(13):i539–i548, 2007.

[239] Qingwu Yang and Sing-Hoi Sze. Predicting protein folding pathways at the mesoscopic level based on native interactions between secondary structure elements. *BMC bioinformatics*, 9(1):320, 2008.

[240] Michael C Baxa, Wookyung Yu, Aashish N Adhikari, Liang Ge, Zhen Xia, Ruhong Zhou, Karl F Freed, and Tobin R Sosnick. Even with nonnative interactions, the

updated folding transition states of the homologs proteins g & l are extensive and similar. *Proceedings of the National Academy of Sciences*, 112(27):8302–8307, 2015.

[241] Philipp Neudecker, Paul Robustelli, Andrea Cavalli, Patrick Walsh, Patrik Lundström, Arash Zarrine-Afsar, Simon Sharpe, Michele Vendruscolo, and Lewis E Kay. Structure of an intermediate state in protein folding and aggregation. *Science*, 336 (6079):362–366, 2012.

[242] José M Martín-García, Irene Luque, Pedro L Mateo, Javier Ruiz-Sanz, and Ana Cámara-Artigas. Crystallographic structure of the sh3 domain of the human c-yes tyrosine kinase: loop flexibility and amyloid aggregation. *FEBS letters*, 581(9): 1701–1706, 2007.

[243] Per J Kraulis. Similarity of protein g and ubiquitin. *Science*, 254(5031):581–582, 1991.

[244] John P Overington. Comparison of three-dimensional structures of homologous proteins. *Current Opinion in Structural Biology*, 2(3):394–401, 1992.

[245] A Maxwell Burroughs, S Balaji, Lakshminarayan M Iyer, and L Aravind. Small but versatile: the extraordinary functional and structural diversity of the beta-grasp fold. *Biol Direct*, 2(3):18, 2007.

[246] Stephen W Michnick and Eugene Shakhnovich. A strategy for detecting the conservation of folding-nucleus residues in protein superfamilies. *Folding and Design*, 3(4):239–251, 1998.

[247] Nikolas S Burkoff, Csilla Várnai, and David L Wild. Predicting protein β-sheet contacts using a maximum entropy based correlated mutation measure. *Bioinformatics*, page btt005, 2013.

[248] Zhiyong Wang and Jinbo Xu. Predicting protein contact map using evolutionary and physical constraints by integer programming. *Bioinformatics*, 29(13):i266–i273, 2013.

[249] Marco Lippi and Paolo Frasconi. Prediction of protein β-residue contacts by markov logic networks with grounding-specific weights. *Bioinformatics*, 25(18):2326–2333, 2009.

[250] Castrense Savojardo, Piero Fariselli, Pier Luigi Martelli, and Rita Casadio. Bcov: a method for predicting β-sheet topology using sparse inverse covariance estimation and integer programming. *Bioinformatics*, page btt555, 2013.

[251] Stefan Seemayer, Markus Gruber, and Johannes Söding. Ccmpred?fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–3130, 2014.

[252] Marcin J Skwark, Daniele Raimondi, Mirco Michel, and Arne Elofsson. Improved contact predictions using the recognition of protein like contact patterns. *PLoS Comput Biol*, 10(11):e1003889, 2014.

[253] Burkhard Rost. Review: protein secondary structure prediction continues to rise. *Journal of structural biology*, 134(2-3):204–218, 2001.

[254] Rhys Heffernan, Kuldip Paliwal, James Lyons, Abdollah Dehzangi, Alok Sharma, Jihua Wang, Abdul Sattar, Yuedong Yang, and Yaoqi Zhou. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific reports*, 5, 2015.

[255] Ji-Tao Huang and Ming-Tao Wang. Secondary structural wobble: the limits of protein prediction accuracy. *Biochemical and biophysical research communications*, 294(3):621–625, 2002.

[256] Renxiang Yan, Jiangning Song, Weiwen Cai, and Ziding Zhang. A short review on protein secondary structure prediction methods. *Pattern Recognition in Computational Molecular Biology: Techniques and Approaches*, page 99, 2015.

[257] Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in Bioinformatics*, page bbw129, 2016.

[258] Jayant B Udgaonkar. Multiple routes and structural heterogeneity in protein folding. *Annu. Rev. Biophys.*, 37:489–510, 2008.

[259] Kevin Drew, Patrick Winters, Glenn L Butterfoss, Viktors Berstis, Keith Uplinger, Jonathan Armstrong, Michael Riffle, Erik Schweighofer, Bill Bovermann, David R Goodlett, et al. The proteome folding project: proteome-scale prediction of structure and function. *Genome research*, 21(11):1981–1994, 2011.

[260] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction: Progress and new directions in round xi. *Proteins: Structure, Function, and Bioinformatics*, 84(S1):4–14, 2016.

[261] Alexander S Rose and Peter W Hildebrand. Ngl viewer: a web application for molecular visualization. *Nucleic acids research*, page gkv402, 2015.

[262] Robert M Hanson. Jmol–a paradigm shift in crystallographic visualization. *Journal of Applied Crystallography*, 43(5):1250–1260, 2010.

[263] Michael J Hartshorn. Astexviewer tm?: a visualisation aid for structure-based drug design. *Journal of computer-aided molecular design*, 16(12):871–881, 2002.

[264] Cédric Binisti, Ahmed Ali Salim, and Pierre Tufféry. Ppg: online generation of protein pictures and animations. *Nucleic acids research*, 33(suppl 2):W320–W323, 2005.

[265] Rajarshi Maiti, Gary H Van Domselaar, and David S Wishart. Moviemaker: a web server for rapid rendering of protein motions and interactions. *Nucleic acids research*, 33(suppl 2):W358–W362, 2005.

[266] PM Stothard. Combosa3d: combining sequence alignments with three-dimensional structures. *Bioinformatics*, 17(2):198–199, 2001.

[267] Warren L DeLano. The pymol molecular graphics system. -, 2002.

[268] Roman A Laskowski, James D Watson, and Janet M Thornton. Profunc: a server for predicting protein function from 3d structure. *Nucleic acids research*, 33(suppl 2):W89–W93, 2005.

[269] Yongwook Choi, Gregory E Sims, Sean Murphy, Jason R Miller, and Agnes P Chan. Predicting the functional effect of amino acid substitutions and indels. *PloS one*, 7 (10):e46688, 2012.

[270] Christian D Schenkelberg and Christopher Bystroff. Interactiverosetta: a graphical user interface for the pyrosetta protein modeling suite. *Bioinformatics*, page btv492, 2015.

[271] Jeffrey J Gray, Sidhartha Chaudhury, Sergey Lyskov, et al. *The PyRosetta interactive platform for protein structure prediction and design: a set of educational modules*. Lulu. com, 2010.

[272] Frazier N Baker and Aleksey Porollo. Coeviz: a web-based tool for coevolution analysis of protein residues. *BMC bioinformatics*, 17(1):119, 2016.

[273] Christopher W Wood, Marc Bruning, Amaurys Á Ibarra, Gail J Bartlett, Andrew R Thomson, Richard B Sessions, R Leo Brady, and Derek N Woolfson. Ccbuilder: an interactive web-based tool for building, designing and assessing coiled-coil protein assemblies. *Bioinformatics*, page btu502, 2014.

[274] Yiwen Chen, Feng Ding, Huifen Nie, Adrian W Serohijos, Shantanu Sharma, Kyle C Wilcox, Shuangye Yin, and Nikolay V Dokholyan. Protein folding: then and now. *Archives of biochemistry and biophysics*, 469(1):4–19, 2008.

[275] Leonid B Pereyaslavets, Igor V Sokolovsky, and Oxana V Galzitskaya. Foldnucleus: web server for the prediction of rna and protein folding nuclei from their 3d structures. *Bioinformatics*, page btv369, 2015.

[276] M Michael Gromiha, A Mary Thangakani, and Samuel Selvaraj. Fold-rate: prediction of protein folding rates from amino acid sequence. *Nucleic acids research*, 34 (suppl 2):W70–W74, 2006.

[277] Shantanu Sharma, Feng Ding, Huifen Nie, Daniel Watson, Aditya Unnithan, Jameson Lopp, Diane Pozefsky, and Nikolay V Dokholyan. ifold: a platform for interactive folding simulations of proteins. *Bioinformatics*, 22(21):2693–2694, 2006.

[278] Guang Song and Nancy M Amato. A motion-planning approach to folding: From paper craft to protein folding. *IEEE Transactions on Robotics and Automation*, 20 (1):60–71, 2004.

[279] Shawna Thomas, Guang Song, and Nancy M Amato. Protein folding by motion planning. *Physical biology*, 2(4):S148, 2005.

[280] Peter W Rose, Andreas Prlić, Ali Altunkaya, Chunxiao Bi, Anthony R Bradley, Cole H Christie, Luigi Di Costanzo, Jose M Duarte, Shuchismita Dutta, Zukang Feng, et al. The rcsb protein data bank: integrative view of protein, gene and 3d structural information. *Nucleic Acids Research*, 45(D1):D271–D281, 2017.

[281] Bonnie Berger, Noah M Daniels, and Y William Yu. Computational biology in the 21st century: scaling with compressive algorithms. *Communications of the ACM*, 59(8):72–80, 2016.

[282] Marianne A Grant. Integrating computational protein function prediction into drug discovery initiatives. *Drug development research*, 72(1):4–16, 2011.

[283] Ke Xia, Marta Manning, Helai Hesham, Qishan Lin, Christopher Bystroff, and Wilfredo Colón. Identifying the subproteome of kinetically stable proteins via diagonal 2d sds/page. *Proceedings of the National Academy of Sciences*, 104(44): 17329–17334, 2007.

[284] Hue Sun Chan and Ken A Dill. Energy landscapes and the collapse dynamics of homopolymers. *The Journal of chemical physics*, 99(3):2116–2127, 1993.

[285] DT Jones, K Bryson, A Coleman, Liam J McGuffin, MI Sadowski, JS Sodhi, and JJ Ward. Prediction of novel and analogous folds using fragment assembly and fold recognition. *PROTEINS: Structure, Function, and Bioinformatics*, 61(S7):143–151, 2005.

[286] Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-grained protein models and their applications. *Chemical Reviews*, 116(14):7898–7936, 2016.

[287] Sebastian Kmiecik and Andrzej Kolinski. Characterization of protein-folding pathways by reduced-space modeling. *Proceedings of the National Academy of Sciences*, 104(30):12330–12335, 2007.

[288] David Baked and A Agard. Perspectives in biochemistry. *Biochemistry*, 33:7505–7509, 1994.

[289] Aaron R Dinner and Martin Karplus. The thermodynamics and kinetics of protein folding: a lattice model analysis of multiple pathways with intermediates. *The Journal of Physical Chemistry B*, 103(37):7976–7994, 1999.

[290] Themis Lazaridis and Martin Karplus. Thermodynamics of protein folding: a microscopic view. *Biophysical chemistry*, 100(1):367–395, 2002.

[291] Amnon Horovitz and Alan R Fersht. Co-operative interactions during protein folding. *Journal of molecular biology*, 224(3):733–740, 1992.

[292] Junmei Chen and Wesley E Stites. Higher-order packing interactions in triple and quadruple mutants of staphylococcal nuclease. *Biochemistry*, 40(46):14012–14019, 2001.

[293] Michael Socolich, Steve W Lockless, William P Russ, Heather Lee, Kevin H Gardner, and Rama Ranganathan. Evolutionary information for specifying a protein fold. *Nature*, 437(7058):512–518, 2005.

[294] J Hofrichter, JH Sommer, ER Henry, WA Eaton, K Imaizumi, K Imai, and I Tyuma. Frazier, r. d. b., & suzuki, e.(1973) in physical principles and techniques of protein chemistry, academic press, new york. gibson, qh (1959) prog. biophys. biophys. chem. 9, 1-53. gill, s. j., di cera, e., doyle, m. l., bishop, g. a., & robert. *Biochemistry*, 27:824–832, 1988.

[295] Rainer Jaenicke. Folding and association of proteins. *Progress in biophysics and molecular biology*, 49(2-3):117–237, 1987.

[296] Michelle Hollecker and Thomas E Creighton. Evolutionary conservation and variation of protein folding pathways: Two protease inhibitor homologues from black mamba venom. *Journal of molecular biology*, 168(2):409–437, 1983.

[297] Fabrizio Chiti, Niccolò Taddei, Paul M White, Monica Bucciantini, Francesca Magherini, Massimo Stefani, and Christopher M Dobson. Mutational analysis of acylphosphatase suggests the importance of topology and contact order in protein folding. *Nature Structural & Molecular Biology*, 6(11):1005–1009, 1999.

[298] David S Riddle, Viara P Grantcharova, Jed V Santiago, Eric Alm, Ingo Ruczinski, and David Baker. Experiment and theory highlight role of native state topology in sh3 folding. *Nature Structural & Molecular Biology*, 6(11):1016–1024, 1999.

[299] Chiaki Nishimura, Stefan Prytulla, H Jane Dyson, and Peter E Wright. Conservation of folding pathways in evolutionarily distant globin sequences. *Nature Structural & Molecular Biology*, 7(8):679–686, 2000.

[300] Robert B Best, Gerhard Hummer, and William A Eaton. Native contacts determine protein folding mechanisms in atomistic simulations. *Proceedings of the National Academy of Sciences*, 110(44):17874–17879, 2013.

[301] M Michael Gromiha and Samuel Selvaraj. Importance of long-range interactions in protein folding. *Biophysical chemistry*, 77(1):49–68, 1999.

[302] Daisuke Kihara. The effect of long-range interactions on the secondary structure formation of proteins. *Protein Science*, 14(8):1955–1963, 2005.

[303] Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, 2012.

[304] Jakob Bohr, Henrik Bohr, Søren Brunak, Rodney MJ Cotterill, Henrik Fredholm, Benny Lautrup, and Steffen B Petersen. Protein structures from distance inequalities. *Journal of molecular biology*, 231(3):861–869, 1993.

[305] Gianluca Pollastri, Alessandro Vullo, Paolo Frasconi, and Pierre Baldi. Modular dag–rnn architectures for assembling coarse protein structures. *Journal of Computational Biology*, 13(3):631–650, 2006.

[306] Stanislav G Galaktionov and Garland R Marshall. Properties of intraglobular contacts in proteins: an approach to prediction of tertiary structure. In *HICSS (5)*, pages 326–335, 1994.

[307] Michele Vendruscolo, Edo Kussell, and Eytan Domany. Recovery of protein structure from contact maps. *Folding and Design*, 2(5):295–306, 1997.

[308] Marco Vassura, Luciano Margara, Pietro Di Lena, Filippo Medri, Piero Fariselli, and Rita Casadio. Reconstruction of 3d structures from protein contact maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 5(3):357–367, 2008.

[309] Michal J Pietal, Janusz M Bujnicki, and Lukasz P Kozlowski. Gdfuzz3d: a method for protein 3d structure reconstruction from contact maps, based on a non-euclidean distance function. *Bioinformatics*, 31(21):3499–3505, 2015.

[310] Alexei A Podtelezhnikov and David L Wild. Reconstruction and stability of secondary structure elements in the context of protein structure prediction. *Biophysical journal*, 96(11):4399–4408, 2009.

[311] Heinz Breu and David G Kirkpatrick. Unit disk graph recognition is np-hard. *Computational Geometry*, 9(1-2):3–24, 1998.

[312] Zheng Dai, David Becerra, and Jérôme Waldispühl. On stable states in a topologically driven protein folding model. *(Submitted) Journal of Computational Biology*, 0(0):0–0, 2017.