Single Mutation Effects on Protein Secondary Structure

by

Raul Ivan Perez Martell

B.Sc., Monterrey Institute of Technology and Higher Education, Mexico, 2016
M.Sc. in Computer Science, University of Victoria, Canada, 2020

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

in the Department of Computer Science

We acknowledge and respect the Lək̓ʷəŋən (Songhees and Xʷsepsəm / Esquimalt) Peoples on whose territory the university stands, and the Lək̓ʷəŋən and W̱SÁNEĆ Peoples whose historical relationships with the land continue to this day.

Single Mutation Effects on Protein Secondary Structure

by

Raul Ivan Perez Martell

B.Sc., Monterrey Institute of Technology and Higher Education, Mexico, 2016
M.Sc. in Computer Science, University of Victoria, Canada, 2020

Supervisory Committee:

Dr. Ulrike Stege, Supervisor
Department of Computer Science, University of Victoria

Dr. Hosna Jabbari, Supervisor
Department of Computer Science, University of Victoria

Dr. Julian J. Lum, Outside Member
Department of Biochemistry and Microbiology, University of Victoria

Abstract

Human diversity often manifests through single nucleotide polymorphisms (SNPs). Among these polymorphisms, SNPs that alter amino acids can modify a protein's three-dimensional (3D) structure. Such single amino acid mutations can impact the protein's function and potentially elicit diseases or affect drug interactions. Thus, understanding protein single point mutations is crucial for precision medicine, as it helps tailor treatments based on individual genetic variations.

Protein tertiary structure prediction models like AlphaFold2 have revolutionized the field with unprecedented accuracy, yet predicting structural changes arising from single amino acid mutations remains a challenge. The complexity introduced by these mutations calls for models that can incorporate mutational information into their predictions. As atomic locations can be susceptible to any number of changes that might or might not affect function, we focus on the secondary structure to provide concrete results on possible protein structural deformation that may occur from single amino acid mutations.

We assess state-of-the-art structure prediction methods regarding backbone deformations caused by single amino acid mutations. We categorize these deformations as local, distant, or global based on the proximity of structural changes to the mutation site. Our analysis utilizes a diverse dataset from the Protein Data Bank, comprising over 500 protein clusters with experimentally determined structures and documented mutations.

Our findings indicate that single amino acid mutations can significantly affect the accuracy of structure prediction methods. These mutations often lead to predicted structural changes even when the actual secondary structures remain unchanged, suggesting that current methods overestimate the impact of single amino acid mutations. This issue is particularly evident in advanced prediction algorithms, which struggle to accurately model proteins with stable mutations. We also found that the addition of low-performing prediction methods during structural analysis can positively impact the results on some proteins, particularly those with low levels of homology. Furthermore, proteins that form complexes or bind ligands—such as membrane and transport proteins—are inaccurately predicted due to the absence of extra-molecular interaction data in the models, highlighting how single amino acid mutations can complicate accurate structure prediction.

Due to these findings, we propose a novel refinement strategy for protein secondary structure prediction that leverages single amino acid mutational data. As part of this strategy, we introduce Mut2Dens, a model that not only yields more consistent predictions for mutational data but also maintains robust predictive performance on non-mutational

datasets. These refined models take multiple predicted secondary structures and generate a mutation-aware secondary structure.

In particular, Mut2Dens employs the extremely randomized trees (ExtraTree) algorithm to avoid overfitting and make effective use of the limited mutational data available from experimentally determined three-dimensional structures. By combining predictions from highly accurate structure prediction models, we create an ensemble that integrates their strengths while enhancing mutational capabilities. This refinement strategy also improves the non-mutational performance of state-of-the-art methods by addressing their most inaccurate and least confident predictions.

Moreover, our refinement strategy reduces improbable outcomes in mutated protein structures—such as transforming π-helices into β-sheets—that can still occur in current prediction models. Finally, by using interpretable machine learning algorithms, we can reveal the underlying biological knowledge from the refinement model. The insights gained from Mut2Dens can be corroborated with known mutational outcomes, helping users pinpoint discrepancies across structure prediction models and make more informed decisions regarding the predicted structures.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The completion of this dissertation represents the culmination of a remarkable journey of learning and discovery, one that would not have been possible without the continuous support and encouragement from my amazing supervisors, Professors Ulrike Stege and Hosna Jabbari. With my deepest appreciation, I acknowledge their persistent guidance, dedication, unwavering encouragement, and invaluable mentorship. Ulrike and Hosna, I am nothing but privileged and proud to have had you both as my advisors during this journey. You have significantly shaped my academic and personal growth during this PhD. Thank you for your patience, sincere positivity, brilliant research ideas, and for keeping an eye on my research and many important aspects of my life. I will eternally cherish the academic, professional, and life lessons derived from your invaluable guidance. I extend my heartfelt thanks to Hausi Muller, as he has been nothing but helpful and inspirational with his uplifting and contagious happiness. Thank you for also providing me with invaluable advice during this journey and for the fun expeditions with Ulrike and all the lab members.

My sincere and profound gratitude extends to my supervisory committee, Dr. Julian J. Lum, for their rigorous review and clever insights into my work. I am also grateful to Dr. Dan Tulpan for acting as the external examiner of my dissertation, and Dr. Juergen Ehlting for serving as the chair of my final oral examination. Conducting our research and achieving the expected contributions was only possible with the institutions that sponsored and funded our work. I acknowledge and affirm my appreciation to the University of Victoria, the National Sciences and Engineering Research Council (NSERC) of Canada, Microsoft AI for Health who supported us through their Azure grant, and the University of Alberta, which allowed us to use their 'Industry Sandbox and AI Computing' entrepreneurship resources for our research needs.

I must extend my thankfulness to the administrative staff of the Faculty of Engineering and Computer Science at the University of Victoria, Nancy Chan, Wendy Beggs, Kath Mizzilano, Aimee Coueslan, and Erin Robinson for your diligence and continuous assistance provided to many aspects of our research and life as a student. I was incredibly fortunate to be part of the talented PITA, COBRA, RIGI, and HighTechU teams, where I met some of the most brilliant and welcoming people I know. Felipe, Priya, Dominique, Alison, Tristan, Finn, Mike, Mateo, Luke, Michael, Miguel, Morgan, Mina, Juan, Jose, Sunil, Giovanni, Alvi, Karan, Samantha, Saasha, Addie, Noah, Tara, Lance, Haley, Andrew, and Connie, working with you has been truly rewarding, and I am grateful for the friendships we have formed along the way. I will always cherish our great trips together. Felipe, our friendship has been invaluable and a source of great joy to me. Priya, thank you for all your help and for being a great friend. I hope our paths cross again in the future. Thank you to Andrew Maclean, for

being an amazing friend and for all the opportunities you provided me during my studies at UVic. We should meetup and enjoy more of the Highland games in the future.

I owe the possibility of having this academic journey to my parents and brother, which without whom I would not have been able to even commence this endeavour. Finally, I lovingly thank my incredible wife Nicole for her cheerfulness and resolute help during my studies. Thank you so much for putting up with me throughout this journey.

Dedication

To my incredible and lovely wife Nicole,
my generous parents Sergio and Susana,
and my amazing brother Sergio.

Part I

Motivation and Context

Chapter 1

Introduction

Contents

Proteins are a major component of organisms on our planet. They are one of the most abundant biomolecules within cells, and take part in most functional and structural processes in biological systems [Legrain et al., 2001]. These processes arise through chemical reactions involving proteins and other molecules, such as the distribution of oxygen by red blood cells within a body [Giardina et al., 1995]. Proteins are themselves composed of molecules known as amino acids, which are also known as residues. Amino acids are molecules with a chemical structure, shown in Fig. 1.1 A, containing an amino group (NH_2), a hydrogen atom (H), a carboxyl group ($COOH$), and a side chain (R group), all linked to a central carbon atom ($C\alpha$). The distinguishing factor between amino acids is their attached side chain. For example, the chemical structure of phenylalanine with its specific side chain composition is shown in Fig. 1.1 B. A peptide bond occurs when the amino group of an amino acid covalently bonds with a carboxyl group of another amino acid through a dehydration synthesis reaction, thus losing a water molecule (H_2O) in the process [Morot-Gaudry et al., 2001]. These peptide bonds allow the formation of the chain of amino acids that make up a protein.

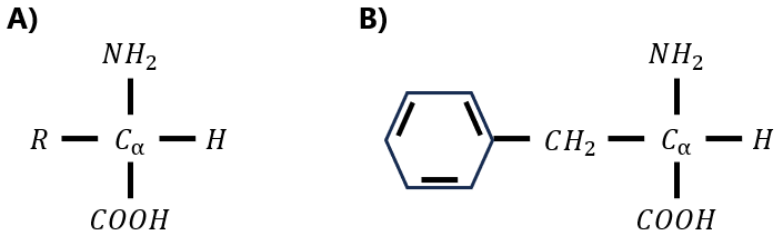


Figure 1.1.|: **Amino acid chemical structure.** Depiction of the chemical structure of: A) a generic amino acid, B) phenylalanine.

The process of transcribing DNA into RNA and then translating the RNA into protein became known as the central dogma of biology [Bustamante et al., 2011]. This dogma explains how the genetic information is utilized to form functional molecules within organisms. Deoxyribonucleic acid, otherwise known as DNA, is an important biomolecule that holds the instructional blueprint of most organisms on our planet. Whereas protein deals with mostly functional or structural capabilities in organisms, DNA is utilized as the instructions in the creation or synthesis of other biomolecules, such as DNA, ribonucleic acid (RNA), and protein, within the cells. RNA is synthesized from DNA by RNA polymerase, along other enzymes, through a process called transcription. Here, a type of RNA called messenger RNA (mRNA) is synthesized to be utilized as a template for the creation of a protein. During protein synthesis, another type of RNA molecule called transfer RNA (tRNA) is capable of transporting amino acids to a macromolecule known as a ribosome, which attaches to the messenger RNA for use as a template, to synthesize the chain of amino acids in a process known as translation. During translation, the tRNA helps the ribosome decode the mRNA through codons, sequences of three nucleotides. Every tRNA molecule contains a specific anti-codon that binds to the codon sequence of the mRNA, bringing a specific amino acid to the ribosome to add to the protein chain. This procedure continues until the ribosome encounters a stop codon, which releases the completed protein.

Variation in the human genome is commonly found via single nucleotide polymorphism (SNP) [Auton et al., 2015]. There are SNPs that occur in coding regions of the genome, having downstream effects by changing the amino acid sequence of proteins, and are referred to as *missense* mutations or non-synonymous SNP (ns-SNP) [Kumar et al., 2009]. SNPs can also lead to stop codons, known as *nonsense* mutations [Berkowitz et al., 1968], which result in the truncation of amino acid chains. Single amino acid mutations are often associated with one or more disease phenotypes [Stenson et al., 2020], thereby potentially affecting the underlying protein's ability to interact with other molecules. Genetic diseases associated with single amino acid mutations include inflammatory and autoimmune diseases, either as a causative or susceptibility factor [Khoruddin et al., 2021]. For example, cystic fibrosis could be predicted from ns-SNP in 40% of the cases [Cutting, 2015]. Also, ns-SNP in BRCA1 can be predictive for breast cancer [Coluccio et al., 2015]. Furthermore, ns-SNP in major prion proteins PrP have been found in neurodegenerative diseases [Bernardi and Bruni, 2019]. Understanding how single amino acid mutations alter the protein's shape, stability, and interactions with other molecules is crucial for advancing our knowledge of proteins and the progression of related fields, such as drug therapeutics.

The structure of a protein can be classified into multiple levels [Eisenhaber et al., 1995]. These levels can be visualized in Fig. 1.2. The ordered chain or sequence of amino acids, described by their side chain, define the primary structure of a protein. During the translation process, the protein chain is created by linking one amino acid at a time. The secondary structure of a protein is mediated by non-covalent bonds, such as hydrogen bonds, between carboxyl and amino groups of different amino acids in the protein chain. These hydrogen bonds form into regular patterns or *motifs* that are categorized by their local folding structure of the backbone atoms in the protein chain, such as helices and sheets.

Tertiary structure is defined by the atomic positions and interactions that occur between side chains and backbone atoms of the protein chain. The tertiary structure describes the specific three-dimensional shape of the protein, which provide its functionality through its ability to bind and react with other molecules. Finally, the quaternary structure is defined by the composition of two or more protein chains that are bound together through atomic interactions to make a protein assembly that acts as a unit.

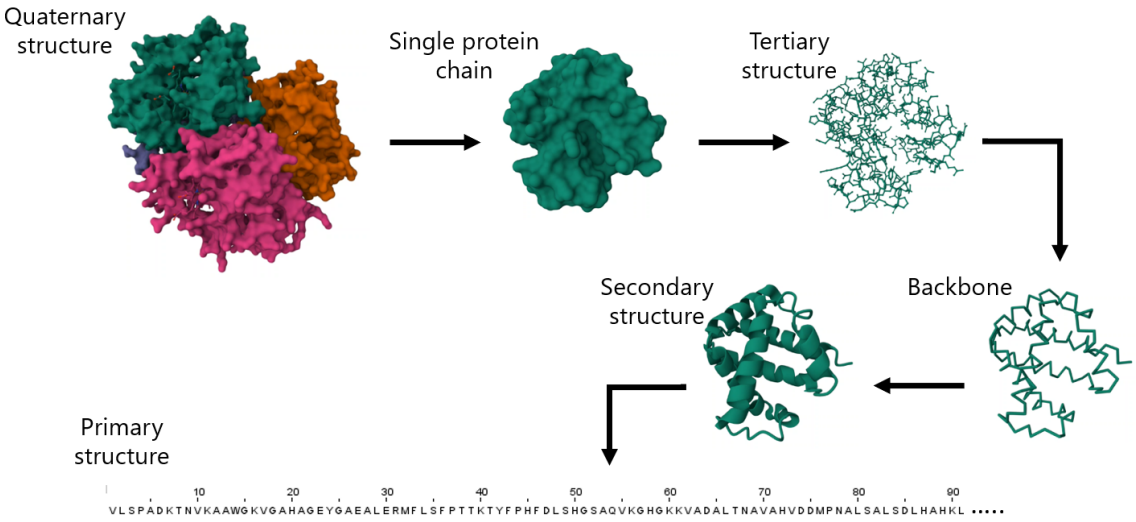


Figure 1.2.|: **Levels of protein structure.** Depiction of the multiple levels of protein structure starting from the highest level (quaternary) at the top left. The specific protein depicted is human foetal deoxyhaemoglobin.

Protein structure gives essential information to infer its functionality inside an organism. Obtaining a protein structure is possible through experimental procedures, such as X-ray crystallography, nuclear magnetic resonance, or electron microscopy [Egli, 2016]. These experimental procedures require large amounts of time, effort, and resources to produce accurate protein structures. These methods provide diffraction patterns, local conformations, and molecular shape representations, respectively. However, experimental data alone is often insufficient and require supplementary information, i.e. amino acid sequences and established atomic geometries, to refine structural models. This integrated approach ensures consistency between empirical observations and expected molecular configurations. Obtaining the sequences of proteins requires substantially less resources due to the advancements in next generation sequencing [Hu et al., 2021] (NGS). The immense quantity of genomic and transcriptomic data from NGS has facilitated the inference of millions of protein sequences. These protein sequences would then be utilized to predict their secondary and tertiary structures without the need for the costly experimental procedures.

Advancements in machine learning and sequence alignment techniques have led to significant progress in predicting secondary and tertiary protein structures [Bokor and Tantos, 2021, David and Sternberg, 2023]. Highly accurate tertiary structure prediction became possible with the inception of AlphaFold2 [McBride et al., 2023], a machine learning model that achieved performance comparable to experimental meth-

ods for proteins with numerous homologs. Homologs are proteins encoded by genes with shared ancestry, which are likely to have similar structure and functionality. The addition of evolutionary data or homology, which groups protein sequences of similar functionality from different species, aids these models in understanding how certain sequences fold. Therefore, the absence of homology leads to lower performance in these models, as structural relations for the protein sequence cannot be made. Secondary structure prediction deals with the secondary structure classification for each amino acid in the protein sequence. Different types of secondary structure exist, depending on the number of secondary structures motifs available for classification. Initially, three-state secondary structure was proposed by grouping the structural motifs into α-helices and β-sheets [Lim, 1974]. Regions that did not belong in those two groups were denoted as coils. Although secondary structure prediction in three states is important, its coarse-grained representation of the backbone structure requires further refinement through expansion into eight classes: $3-10$ helices, α-helices, π-helices, β-sheets, isolated β-bridges, hydrogen-bonded turns, non-hydrogen-bonded bends, and coils [Kabsch and Sander, 1983a]. Eight-state secondary structure allows for structural state deviations that significantly differ from standard helix and sheet conformations, which can result in better structure resolution.

Protein three-dimensional structure is inherently noisy due to the dynamic nature of atomic positions. This includes atomic vibrations, environmental conditions during experimentation, and the inherent flexibility of protein structures. Therefore, obtaining a precise and stable representation of the protein's architecture through three-dimensional structure is challenging. Protein secondary structure helps mitigate these inconsistencies and variations introduced by noisy atomic coordinates and atomic-level fluctuations by discretizing the atomic-level details, thereby elucidating mutational effects in the protein's architecture or backbone structure. This removal of confounding atomic-level variations introduced by experimental and physical factors can increase the reliability of observed structural changes.

A conclusive result has not yet been determined for the ability of current highly accurate structure prediction methods to correctly determine protein structures derived from single amino acid mutations. Recent studies [McBride et al., 2023, Keskin Karakoyun et al., 2023, Ahdritz et al., 2024] have evaluated high-performing structure prediction methods on single amino acid mutations, but their backbone analyses are limited to *three-state secondary structure* (Q3). In this dissertation, we investigate the capabilities of state-of-the-art structure prediction methods on correctly assessing backbone structural changes from single amino acid mutation through eight-state secondary structure prediction. We evaluate the prediction methods using conventional measures for secondary structure prediction, as well as mutational measures that we develop to more carefully assess their mutational capabilities. We analyse the weaknesses of current prediction methods through a mutational benchmark and find that the frequency of secondary structural interchanges observed in predictions remains higher than what current experimental evidence suggests. These undesirable secondary structure predicted changes lead to unreliable structure predictions for single amino acid mutated sequences. We address these limitations of current prediction methods by developing a refinement strategy that relies solely on single

amino acid mutational data to correct the frequency of secondary structural changes. Our refinement strategy is implemented in the *Mut2Dens* model, which integrates the strengths of multiple prediction methods, improving performance on mutational datasets without compromising results on non-mutational datasets.

The remainder of this chapter presents the details concerning the problem addressed by this dissertation and our contributions, as follows. Sections 1.1 – 1.3 explain the problem statement, addressed challenges and corresponding research questions. Section 1.4 summarizes the goals of this dissertation, and Section 1.5 explains how the remainder of this dissertation is organized.

1.1. Problem Statement

Based on the context previously presented, we state the research problem addressed in this dissertation as follows:

> *The advent of highly accurate protein structure prediction models in recent times has accelerated the knowledge in protein functionality and binding capabilities through its advanced prediction of protein shapes, although low-confidence and low-quality structure predictions are still possible. It is still unclear whether these highly accurate protein structure prediction models are capable of correctly predicting the protein shape for proteins derived from single amino acid mutations. Therefore, to assess their capabilities on single amino acid mutations and improve their most unfavorable predictions: i) the effects of single amino acid mutations on experimental protein structure must be analysed; ii) an assessment of current structure prediction methods should be performed on single amino acid mutation data; iii) discrepancies, if any exist, must be examined between experimental and predicted data for single amino acid mutations; and iv) low-quality structure predictions must be addressed.*

1.2. Addressed Challenges

We constrain the problem statement with key challenges, helping us pose the research questions that drive our exploration for a solution.

CH1: There is a limited number of mutational structure data available for proteins. While proteomics has generated numerous sequence data, protein tertiary structure data, which is required to assign its secondary structure is still relatively limited.

CH2: Experimental data is primarily obtained from crystallographic data, which necessitates that proteins form ordered crystal structures. Therefore, mutated proteins are limited to those who can be crystallized, and might not represent the complete breadth of mutated structures.

CH3: The secondary structure of a protein might not have the resolution to show mutational effects. Conversely to tertiary structure, the classification of structural motifs within the protein backbone might not separate slight atomic location differences. This can lead to incomplete representations of the mutational effect on the protein structure.

CH4: Secondary structure prediction has been extensively researched, as such the possibility to improve predictions are diminished. Furthermore, tertiary structure prediction methods have reached similar performance to secondary structure prediction methods.

1.3. Research Questions

The aforementioned challenges help us pose questions to guide or search towards a solution for the research problem. As the research problem is considerably complex, we utilize our research questions to address its complexities in a more digestible manner throughout this dissertation.

[RQ1]:How can we collect experimental data for single amino acid mutations that cause both local and distant structural changes in a protein?

- [RQ1.1]:Is the breadth of the experimental data broad enough, where it contains single amino acid mutations that are stable and disruptive?
- [RQ1.2]:Which single amino acid mutations are likely to cause local structural changes?
- [RQ1.3]:Which single amino acid mutations are likely to cause distant structural changes?

[RQ2]:How can we evaluate the performance of methods predicting backbone structural changes in proteins?

- [RQ2.1]:Are conventional secondary structure measures sufficient to evaluate mutational performance?

[RQ3]:Are any of the selected structure prediction methods sufficiently precise to show the effect of single amino acid mutations on protein backbone structure?

- [RQ3.1]:If not sufficiently precise, what are possible improvements that can be made to these methods regarding single amino acid mutations?

[RQ4]:Are secondary structure prediction methods redundant now that tertiary structure prediction methods have reached comparable performance to the former?

1.4. Dissertation Goals and Scope

We now establish the scope of this dissertation by specifying its general and specific goals. The latter allows us to focus on providing specific solutions that together address the research problem.

The general goal of our work is to advance protein structure prediction for single amino acid mutational changes, aiming to contribute to the improvement of the reliability from drug therapeutics in disease phenotypes due to protein folding differences in people.

G1: Create a mutational dataset with a wide breadth of mutational effects — containing local and distant mutational effects on protein structure, as well as stable and disruptive mutations.

G2: Evaluate the performance of structure prediction methods on backbone structural changes in proteins through the mutational dataset. The evaluated methods should include both state-of-the-art secondary and tertiary structure prediction methods.

G3: Analyse and investigate deficiencies on evaluated prediction methods regarding backbone structural changes due to single amino acid mutations.

G4: Create a secondary structure prediction method that is mutation-cognizant, comparable to current state-of-the-art prediction methods on mutation-agnostic predictions, and outperforms them in mutation-specific predictions.

1.5. Dissertation Outline

The remainder of this dissertation is organized as follows.

Chapter 2 Context and State-of-the-Art Background This dissertation addresses a research problem that integrates computer science, molecular biology, and machine learning. We introduce the fundamental concepts of each area relating to protein structure prediction methods, alongside their analysis and evaluation. We also cover related work and software that contributed to the work in this thesis.

Part II: Contributions

Chapter 3 Contributions Overview This chapter briefly describes our investigation of protein structure prediction methods: in particular, conceptualizing the current performance of these methods on single amino acid mutation data. We use protein secondary structure

to simplify the process and evaluate their performance on mutational data. Our contributions culminate in the creation of a novel refinement strategy and model for protein secondary structure prediction to improve deficiencies found during evaluation.

Chapter 4 Single amino acid mutations: Backbone structure positional effects In Chapter 4, we discuss single amino acid mutations and their importance with regards to protein structure prediction methods. We discovered that single amino acid mutations can significantly alter the accuracy of structure prediction methods, especially in advanced prediction algorithms, and can complicate accurate structure prediction. Next, we outline terminology and methods used for analyzing and evaluating the performance of structure prediction methods. Results showed difficulty in predicting stable mutations and inability to detect improbable mutational changes. Although current prediction models achieve high accuracy, they need to be further refined in order to become more stable, mutation-aware, and accurate.

Chapter 5 Single amino acid mutation knowledge can decrease prediction inaccuracies on protein secondary structure In this chapter, we utilize knowledge obtained previously from single amino acid mutations to propose a refinement strategy. This strategy improved the prediction of protein structures in both mutational and non-mutational manners. The research led to the implementation of Mut2Dens, an ensemble model that improves upon current methods with our mutational dataset to generate mutation-aware secondary structures. Finally, we evaluate Mut2Dens and compare it to state-of-the-art prediction methods. This showcases the complementary nature of our refinement strategy for low-confidence predictions from highly accurate prediction methods.

Part III: Summary

Chapter 6 Conclusions This chapter summarizes our research challenges and how we addressed them, followed by the goals established in our pursuit for assessing and improving the performance of protein structure prediction methods. We end by discussing our contributions and their significance, as well as future work opportunities drawn from our research.

1.6. Chapter Summary

This chapter presented the motivation, problem statement, addressed challenges, questions, and goals of this dissertation. From the problem statement, we identified the addressed research challenges and questions. Then, to limit the scope of our contributions, we established a set of goals based on the identified challenges. Overall, this chapter introduced protein structure prediction and structural effects from single amino acid mutation as the main topics of this dissertation.

Chapter 2

Background

Contents

The research problem we address in this dissertation intersects computer science, molecular biology, and machine learning. Hence, this chapter presents fundamental concepts from these areas as related to the scope of this dissertation.

This chapter is organized as follows. Section 2.1 extends the concepts from the introduction to get a comprehensive view of the biological procedures and ideas that allow and facilitate the creation of protein structure prediction methods. Section 2.2 introduces the fundamental computational concepts for the analysis of protein structure. In Section 2.3,

we describe the machine learning techniques and validation procedures utilized for structure prediction tasks. Section 2.4 details the structure prediction methods that have advanced the field into its current state. Finally, Section 2.5 describes the software packages that made our research possible.

2.1. Molecular Biology

The study of living organisms on our planet started as early researchers classified them into groups by the similarity of their visual characteristics. Later, microorganisms were discovered, and a better system of classification was required to identify and group them, as smaller organisms were poorly distinguishable through visual characteristics [Padian, 1999].

While different types of organisms have many distinctive traits, their internal workings consist of the same family of molecules. As previously mentioned, their genetic material consists of chains of nucleic acids, while their functional components mostly consist of chains of amino acids and chains of nucleic acids to a lesser degree [Ho et al., 2018].

These molecules are housed inside membrane-bound units, known as cells, for living organisms. Current understanding of the instructions by which cells undergo their processes is still at its infancy, although much progress has been achieved. These main instructions are encoded by DNA and are located inside a membrane-bound organelle called the nucleus for eukaryotic cells [Lamond and Earnshaw, 1998], in a region called nucleoid for prokaryotic cells [Robinow and Kellenberger, 1994], and as circular DNA molecules within the cells of archaea [Poole and Penny, 2001]. This similarity of all organisms utilizing nucleic acid chains for their genetic material allowed researchers to classify even visually indistinguishable organisms through their genetic variety.

The central dogma of molecular biology states the unidirectional flow of information, which mostly occurs from DNA to RNA and into protein, or simply from nucleic acid to protein. More recently, the original inception of this dogma has been extended to allow for additional transferring of biological information through nucleic acids. These additions were discovered from their extensive manifestation in viruses [Bustamante et al., 2011]. An extension of the information flow within nucleic acids includes DNA replication, where DNA is synthesized to produce copies of the original genetic material. Here, DNA is synthesized from DNA by DNA polymerase and other biological catalysts or enzymes. This replication capability is also possible for RNA, through a process known as RNA replication through RNA-dependent RNA polymerases. Finally, RNA can also be utilized to produce DNA in a process called reverse transcription, closing the information loop between nucleic acids.

While RNA is also functionally active within cells, its unstable and transient nature [Sachs, 1993, Svenningsen et al., 2017] is disfavored to more varied, stable and efficient proteins. From structural support to external and internal signal responders, proteins are required to perform most of the cell's diverse functionality. Their functionality arises

from their composition of amino acids, which gives them a defined shape or structure that allows them to bind to specific molecules. This binding capability produces reactions with its target ligand, yielding a function within the cell.

Protein shape can be transient depending on its environment [Schilder and Ubbink, 2013]. The atoms within a protein interact with molecules within the protein and with molecules in its surroundings. The potential energy function, which specifies the total potential energy of a system of atoms as a function of all their positions, allow for stability and atomic bonds that results in a conformation for the protein [Privalov, 1997]. These electrostatic interactions require energy to displace, and as such the stable conformations tend to favor a system of interactions with the least amount of energy — as any perturbation to the conformation would require additional energy outside the system.

As previously stated, protein structure can be classified into four levels: primary, secondary, tertiary and quaternary [Rehman et al., 2022]. A protein is also considered a polypeptide because it is a continuous, unbranched chain of peptides [Stein and Moore, 1961]. The start of this chain is known as the N-terminus and is derived from the exposed amino group at that extremity of the chain. Similarly, the end of the chain is known as the C-terminus and is derived from the exposed carboxyl group at that extremity of the chain. This chained structure of peptides form the *backbone* of the protein. Secondary structure is the local spatial arrangement of the protein backbone, such as α-helices and β-sheets, which are stabilized by hydrogen bonds between the peptide atoms. Common angle patterns found in protein backbones were first discovered by Pauling, Corey and Branson, with more patterns discovered afterward [Eisenberg, 2003]. These patterns later became known as secondary structure elements and form the basis of the protein structure for self-assembly into its final shape or conformation. This collection of secondary structure elements for a protein is known as the protein's secondary structure. Tertiary structure is the three-dimensional shape of the entire polypeptide chain, determined by interactions among the backbone and side chain atoms of amino acids, such as ionic bonds, van der Waals forces, and hydrophobic or hydrophilic interactions [Sobolev et al., 1999, Conte et al., 1999]. Quaternary structure is the three-dimensional arrangement of the subunits in a protein that consists of more than one polypeptide chain, such as hemoglobin.

There are 20 standard amino acids encoded directly by codons found in mRNA in most organisms on our planet [Morot-Gaudry et al., 2001]. Other rarer amino acids, known as nonstandard, include selenocysteine and pyrrolysine [Atkins and Gesteland, 2002]. Post-translational modifications [Ramazi and Zahiri, 2021] can also modify standard amino acid side chains into different molecules, such as hydroxyproline via hydroxylation of proline, which adds a hydroxyl group ($-OH$) to proline.

Handling large protein sequences required notation to abbreviate the amino acids in a concise manner. A collaboration between the International Union of Pure and Applied Chemistry (IUPAC) and the International Union of Biochemistry and Molecular Biology, formerly known as the Union of Biochemistry (IUB), created standardized nomenclature for

the standard, ambiguous, and unspecified amino acids in protein sequences. The nomenclature included 1- and 3- letter codes for each amino acid. This nomenclature was later expanded to include nonstandard amino acids within their 3-letter nomenclature. The standard amino acids, along their IUPAC-IUB abbreviations are shown in Table 2.1.

Amino acid	3-letter code	1-letter code
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartate	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamate	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Table 2.1.: **Amino acid abbreviation list.** The 20 standard amino acids and their abbreviations proposed by IUPAC-IUB.

Protein tertiary structure can be approximated by the angles of bonds between certain backbone atoms. *Ramachandran* angles [Ramachandran et al., 1963] are torsion angles ϕ, ψ, and ω that describe rotation of the backbone around the bonds between $N - C\alpha$, $C\alpha - C'$ and $C' - N$ respectively. Often, ω measures $180°$ and thus is rarely used. However, discarding ω might lead to structure reconstruction inaccuracies. Likewise, Frenet-Serret frames [Frenet, 1852, Serret, 1851] are used in protein modelling to generate the backbone structure of a protein. The backbone geometry is based on two local parameters, *curvature* and *torsion*. These parameters have fixed values for each $C\alpha$ in an amino acid. The Frenet-Serret formulas describe the backbone structure using the backbone itself as reference frame. The complete backbone structure is therefore built sequentially, requiring at least four $C\alpha$ atom locations to create the necessary angles to reconstruct subsequent amino acids. This characterization of tertiary structure through backbone atoms leads to the potential of estimating sidechain atoms based on the location of $C\alpha$ atoms.

Single nucleotide polymorphisms (SNPs) are variations in a single DNA base at a spe-

cific position in the genome. They are the most common type of genetic variation among people and can influence health, disease, drug response and other traits. SNPs can occur in both coding and non-coding regions of the genome. SNPs in coding regions can be classified as synonymous or non-synonymous. Synonymous SNPs do not change the amino acid sequence of the protein. Non-synonymous SNPs, which include missense and nonsense mutations, change the amino acid sequence of the protein and can lead to profound changes in its structure. Missense mutations alter the protein by mutating a single amino acid, while nonsense mutations lead to the truncation of a protein chain and can involve the removal of multiple amino acids from the chain. The truncation of the protein chain can result in the loss of partial or complete functionality of the protein. Similarly, single amino acid mutations can affect the structure and function of the protein, and may cause diseases or alter a drug response by changing the protein's binding affinity to it. In non-direct manners, protein can also be affected by SNPs in non-coding regions where gene expression, or mRNA structure is altered, which can lead to disease susceptibility [Sauna and Kimchi-Sarfaty, 2022].

A protein sequence and structure can also be altered by processes outside of missense or nonsense mutations. Post-translational modifications can occur to a protein after its synthesis, where enzymes catalyze reactions in the newly created protein to change its covalently-linked amino acid chain. This process can modify a protein in many forms, e.g. cleave peptide bonds in the protein, or change existing functional groups to other molecules. While many processes exist inside the cell that can alter a protein sequence, we focus on single amino acid mutations as a this form of alteration can also have profound effects on its structure.

While many proteins adopt a well-defined structure to carry out their function, a significant fraction consists of polypeptide segments that are not likely to have a defined structure, known as intrinsically disordered regions (IDRs) [Deiana et al., 2019], but are nevertheless functional. These IDRs are an important component to the protein functionality, although having a prevalence of amino acid mutations [Vacic and M. Iakoucheva, 2012]. Therefore, making these proteins' binding capabilities to other molecules difficult to analyse.

There are many ways to measure the effects of single amino acid mutations on the structure of a protein. Measurements can be done through different levels of protein structure by comparing the original non-mutated, also known as wild-type, protein to the mutated protein. For example, comparing the proteins through their primary sequence can tell us which amino acid was mutated and where in the protein sequence the mutation occurred. Further structural information can be obtained by comparing their secondary structures, where we can obtain the changes throughout the protein backbone. This information can show how the backbone is altered and the alteration's distance in the sequence to the mutation occurrence. Tertiary structure can further increase the information by comparing the atomic locations within the wild-type and mutated proteins.

While tertiary structure increases the comparison's information quantity, it is non-trivial to qualify these changes. A myriad of structure similarity measures to quantify these changes have been proposed and utilized throughout the field's existence. These measures

differ in how they define and quantify the structural similarity, and they have different advantages and limitations. The prominent measures described here are well-established in the field [Kryshtafovych et al., 2021] of protein structural comparison, but there are many others that have been developed for specific purposes or applications.

These tertiary structure similarity measures are mostly utilized to validate the protein structures from prediction methods against experimentally obtained structures. Most structure similarity measures require a form of structure superimposition. This can be done by translating and rotating the protein structure to most closely superimpose the structures, or by solely rotating the structures using a common origin [Kabsch, 1976]. This origin is usually calculated as the central three-dimensional point in the structures. Sequence similarity is also required for many of the structure similarity measures. This is required to create an efficient superimposition of the structures and to have a one-to-one atomic correspondence during the comparison.

Root mean square deviation (RMSD) [Kabsch, 1976] requires a one to one amino acid correspondence, but instead of identical sequence similarity, it leverages any type of structural superimposition algorithm for the proteins to calculate the distance between their amino acids. Usually the comparison is done with the $C\alpha$ atoms, but it is possible to utilize all the atoms in the proteins to obtain a score. A smaller value means more similarity between proteins. It is calculated as the square root of the average squared distance between the corresponding atoms of the two structures. RMSD is sensitive to the size and orientation of the structures, and it does not account for the global or local topology of the structures. Therefore, RMSD may not reflect the true similarity between two structures that have different shapes or folds.

Longest common subsequence (LCS) [Zemla, 2003] also requires a one to one amino acid correspondence for a superimposition to be done, thus the compared proteins must have identical sequences. LCS is usually calculated as a measure for the longest number of amino acids in a protein that do not deviate more than 5 Å. LCS can use different distance thresholds, such as 1, 2 and 5 Å, to measure different levels of similarity. The LCS score can range from 0 to the length of the protein, with higher scores indicating greater similarity between two structures.

Global distance test (GDT) [Zemla, 2003] also requires a one to one amino acid correspondence for a superimposition to be done. GDT can use different distance thresholds, such as 1, 2, 4 and 8 Å, to measure different levels of similarity. GDT is usually calculated as a measure for the total number of amino acids in a protein and is averaged over different thresholds to obtain a single score, called GDT_TS. The distance between the corresponding $C\alpha$ atoms for each of the corresponding proteins are taken into consideration for the final score. The GDT_TS score ranges from 0 to 100, with higher scores indicating greater similarity between two structures.

Global distance calculation (GDC) [Kryshtafovych et al., 2014] is a modification to the GDT algorithm to account for the functional ends of protein sidechains. This measure is therefore usually known as GDC-sc where sc stands for sidechain. To calculate this

measure, the $C\alpha$ atom utilized for calculating GDT is switched to a characteristic atom near the end of each sidechain type.

CACA [Kryshtafovych et al., 2014], named after a pair of $C\alpha$ atoms, is a measure that indicates the mean distance between adjacent $C\alpha$ atoms in a protein and therefore does not require superimposition or sequence similarity. Despite this, the proteins must have the same number of amino acids to compare the mean distances of $C\alpha$ atoms between the structures.

Local distance difference test (lDDT) [Mariani et al., 2013] indicates the distance difference of atoms in a protein structure within a certain threshold range from each atom. This measure does not require superimposition of the molecules to compute the score and can also produce scores at a per-residue level, which can show how a residue is located relative to other amino acids.

Template modeling score (TM-score) [Zhang and Skolnick, 2004] indicates the similarity between two structures by a score between 0 and 1, where a higher score means a better alignment. It uses the distance between pairs of aligned residues to calculate a score and weighs them according to this distance so as to prevent outliers from heavily penalizing the score. The distance score is a logistic function that decreases from 1 to 0 as the distance increases from 0 to a cutoff value. The cutoff value is proportional to the length of the target structure, so TM-score is less dependent on the size of the structures. TM-score also considers both global and local similarity by using a dynamic programming (DP) algorithm to find the optimal alignment.

While quantifying these structural changes is done through these tertiary structure similarity measures, qualifying the structural changes requires a way to categorize different discrete changes to the structure in an unambiguous manner. Protein tertiary structure is the protein's overall conformation, with its three-dimensional atomic arrangement tied to the precise spatial coordination of secondary structure elements. Because tertiary structure depends on a dynamic environment, its conformation is constantly fluctuating, albeit partially stabilized by interatomic bonds. The hierarchical model of protein folding [Baldwin and Rose, 1999a, Baldwin and Rose, 1999b] supports this view, positing that pre-formed local secondary structures fold progressively into larger superstructures with native-like topology, finalizing in the protein's overall conformation. Secondary structure is important for the local structure within the protein, as it provides stability and regularity to the protein backbone. It also affects the overall shape and function of the protein, as different secondary structure elements allow for different properties and interactions within the protein and with other molecules. Consequently, analyzing the protein backbone topology through its secondary structure elements can be used to qualify and detect folding transitions or structural changes. The classification of the secondary structure elements through their assignment using tertiary structure can then be utilized as a way to qualify structural changes due to single amino acid mutations.

As previously mentioned, the secondary structure is a hierarchical level below tertiary structure, and thus a protein's secondary structure can be assigned through the atomic positions given by its tertiary structure. Experimentally, there are different methods to

determine the structure of proteins at different levels of complexity. The primary structure can be determined by methods such as Edman degradation [Smith, 2001], mass spectrometry [Hunt et al., 1986], or inferred through the large amounts of data from genomic and transcriptomic NGS techniques. The secondary structure can also be determined experimentally by methods such as circular dichroism spectroscopy [J. Miles et al., 2021] or hydrogen-deuterium exchange mass spectrometry [Yan and Maier, 2009]. Most effort is devoted to the determination of tertiary structure by several methods, including:

- **X-ray crystallography** [Ilari and Savino, 2008]: This method requires the protein to be purified and crystallized, then exposed to an X-ray beam. The X-rays are diffracted by the atoms in the protein, forming a characteristic pattern that can be analyzed to obtain a map of the protein's electron density. The atomic model of the protein can then be built based on the electron density map and from the known geometry of amino acids.
- **Nuclear magnetic resonance (NMR) spectroscopy** [Wüthrich, 1989]: This method does not require the protein to be crystallized, but it does require high concentrations of the protein in solution. The protein is placed in a strong magnetic field, and radio waves are used to excite the nuclei of certain atoms, such as hydrogen, carbon, and nitrogen. The resulting signals from the protein's atoms reflect the local environment and distance to other atoms, which can be used to calculate the three-dimensional structure of the protein.
- **Cryogenic-electron microscopy (cryo-EM)** [Zhang et al., 2020]: This method is suitable for large proteins or protein complexes that are difficult to crystallize or solubilize. The protein is flash-frozen in a thin layer of ice, and then imaged by an electron microscope. The images are processed and averaged to obtain a three-dimensional reconstruction of the protein at near-atomic resolution.

These methods have different advantages and disadvantages in terms of resolution, speed, cost, and applicability. Depending on the research question and the availability of resources, one or more methods can be used to determine the structure of a protein.

Like with secondary structure elements, the tertiary structure of a protein can be grouped into regions containing common patterns and evolutionary relations. A domain is a region of the protein that has a stable three-dimensional structure and often a specific function, such as binding to another molecule or catalyzing a reaction. Domains can be shared among different proteins that belong to the same gene family or superfamily, indicating a common evolutionary origin. A protein superfamily is a group of proteins that have similar folds, which are the general aspects of protein architecture, such as helix bundle, beta-barrel, or Rossman fold. Superfamilies can be further subdivided into families, which are proteins that have more sequence similarity and functional similarity within each superfamily [Orengo et al., 1994].

Efforts in discovering the secondary structure elements began after their predicted existence through model-building by Pauling et al [Pauling et al., 1951]. The α-helix was first discovered in myoglobin through X-ray crystallography experiments by

Kendrew et al. [Kendrew et al., 1958], while the β-sheet was later found within the X-ray structure of lysozyme. Additional secondary motifs arose after the discovery of β turns that had well-defined precise conformations. A decade after, Kabsch and Sander [Kabsch and Sander, 1983a] generalized the most common secondary structural motifs into the following:

- 3-turn helix ($3-10$ helix) – A helix-like structure with a minimum length of 3 residues.
- 4-turn helix (α helix) – A helix-like structure with a minimum length of 4 residues.
- 5-turn helix (π helix) – A helix-like structure with a minimum length 5 residues.
- β turn – A hydrogen bonded turn-like structure involving three, four, or five residues.
- β-sheet – Extended strand conformation forming a pleated sheet structure.
- Isolated β-bridge – A single pair β-sheet hydrogen bond formation, thus no extended strand occurring.
- Bend – A non-hydrogen-bond based assignment defining high curvature backbone conformations, generally above $70°$.

During that period, repeating proline residues were found to form helix-like structures. These secondary structure conformations arising from sequential proline residues were termed as polyproline helices. Polyproline I helix (PPI) [Traub and Shmueli, 1963] comprises of a right-handed compact helix structure, while polyproline II helix ($PPII$) [Cowan and McGAVIN, 1955] comprises of a left-handed helix structure. Further subclassification of previous secondary structure elements, e.g. helices by their handedness, have been refined within the field. Moreover, additional secondary structure element classifications, e.g. Ω-loops [Leszczynski and Rose, 1986], have been discovered. Alas, as with the subclassifications, these secondary structure elements are rarely encountered within proteins and as such not commonly utilized.

Like with tertiary structure, secondary structure can be compared through similarity measures between the structures of proteins. Here, secondary structure elements are located in a sequential manner corresponding to the amino acid sequence of a protein, the comparison is treated like sequence similarity. Here, one-to-one comparisons of each of the amino acid secondary structures are performed by two commonly utilized measures, ACCURACY and SEGMENT OVERLAP. ACCURACY measures the percentage of matching residues in the secondary structures. SEGMENT OVERLAP provides a more nuanced evaluation by considering the overlap between secondary structure elements or segments, where their boundaries are given flexibility by leniently penalizing minor adjustments. This flexibility allows for small changes to the secondary structure without impacting its overall topology.

Our overview in tertiary structure measures identified the need for alternative measures, which we obtained through secondary structure. Although the local structures within the protein are considered, the tertiary structure scores focus on distances between atoms or amino acids. These distances do not allow for nuanced changes to the structure, such

as considering the angles between the amino acids. Therefore, focusing on the backbone changes through secondary structure measures can allow for structural changes to be distinctive to changes due to protein flexibility.

2.2. Computational biology

Molecular biology, as many other fields, began advancing at an accelerated rate after the emergence of computational technology in the 20^{th} century. Computers made the collection and storage of substantial amounts of biological data feasible, alongside an increase to the accuracy and speed to the analysis of this data.

Early models of protein structure, such as the α-helix and β-sheet structures theorized by Pauling et al. [Pauling et al., 1951], were developed through theoretical calculations of the possible physicochemical interactions within the molecule. Their theoretical modeling considered the bond angles, lengths, along other steric constraints within the protein backbone, and validated through physical models of polypeptide chains. The data was obtained through the extensive analysis of X-ray crystallographic data. This process required extensive expertise in physics and chemistry, and thus was a time-consuming endeavor.

After computers became more widely available and accessible at the end of the 20^{th} century, the rate of data acquisition grew exponentially giving rise to the term of 'Big Data'. The analysis of such amount of data required efficient algorithms to ameliorate time-consuming human calculations. Although much progress has been done to achieve algorithms with high efficiency, computational complexity theory has not yet proven whether many problems, including protein structure prediction, can be efficiently solved using traditional computational techniques. Interest in alternative methods of computation, such as quantum computing, have been proposed and are actively being investigated. Despite recent progress [Castelvecchi, 2024], these alternatives do not generally surpass traditional computation limitations.

Nowadays, protein structure data is mainly found in the protein data bank (PDB) [Burley et al., 2017]. The PDB, first established by the Cambridge Crystallographic Data Centre and Brookhaven National Laboratory in 1971 and later transferred to the Research Collaboratory for Structural Bioinformatics (RCSB), is a free and publicly accessible archive of macromolecular structural data. The archive initially contained protein crystallographic data obtained through X-ray crystallography. A few years after, RNA and DNA crystallographic data started being deposited into the PDB, thus increasing its breadth to other macromolecules. With the inception of electron microscopy and subsequently nuclear magnetic resonance models, further structural data became available. By the end of the millennium, in 2000, more than $10,000$ structures were available in the PDB archive. Fourteen years later, the number of structures grew to more than $100,000$. Less than 10 years after that, $200,000$ structures were part of the archive. This explosion of data has allowed researchers to increase the knowledge within molecular biology, but also has allowed the accurate prediction of molecular structures.

Protein structure prediction is done at the multiple levels of structural classification. Utilizing the protein primary structure, otherwise known as the protein sequence, protein secondary structure prediction is the task of predicting the protein's secondary structure. As previously mentioned in chapter 1, the secondary structure is given as a sequence of secondary structure elements relating to the protein sequence. Thus, for each residue in the protein, its involvement in a secondary structure element is predicted. To validate protein secondary structure predictions, a structure assignment for the protein is required. Although secondary structure can be obtained through experimental methods described previously, secondary structure is usually assigned through its tertiary structure. Early methods of secondary structure assignment required experts in the field to manually assign the structure by analyzing the locations, distances and bonds of backbone atoms. This methodology was prone to human error and variability, and as such computational methods were created to standardize this process. Many secondary structure assignment computational methods were developed, and as with humans, their different methodologies led to different secondary structure assignments. Therefore, this problem required a *de facto* standard computational method. The *de facto* method was largely chosen by the community to be the pioneering method DSSP [Kabsch and Sander, 1983a], which has been extensively tested and utilized for this task. A list of recent secondary structure assignment methods and their methodologies for secondary structure assignment is given in Table 2.2.

As the name implies, protein tertiary structure prediction is the task of predicting a protein's tertiary structure. This includes predicting the relative location of all atoms within the protein. Thus, tertiary structure contains significantly more data than the primary and secondary structure levels. In order for computers to effectively store and process protein tertiary structure, the creators of the PDB created a file format specifically for this task. The PDB format allowed researchers to contain all atomic location data, along with protein metadata, such as its name, physical properties, and potential functionality, in a single packaged file. As the PDB format was created in the early days of computing, where punch cards were utilized as the main storage devices, the format was limited to the technology of the time. While the digital PDB format continued to evolve after punch cards became redundant, its size and extensibility limitations caused complications for newer experimental technologies and bigger macromolecular sizes. Therefore, the RCSB created an extensible format, known as the PDB Exchange/Macromolecular Crystallographic Information File (PDBx/mmCIF), to solve these issues [Adams et al., 2019].

Computational prediction of protein structure at the secondary and tertiary level are problems that have yet to be efficiently and exactly solved. An efficient algorithm is one that is guaranteed, for all possible inputs, to have a runtime bounded by a polynomial function in regards to the size of the problem. For protein structure prediction, the problem size is considered from the length of the amino acid sequence that makes up its primary structure. Problems are considered intractable when no efficient algorithm exists for it. These structure prediction problems have been found to be intractable even through simplified theoretical models. The simplification process involves reducing the atoms in the structure, which is known as coarse-graining [Kneller and Hinsen, 2015]. This process usually involves reducing the structure to only include backbone atoms or reduce each

Method name	Methodology
Levitt & Greer [Levitt and Greer, 1977]	Hydrogen bonds and geometry
DSSP [Kabsch and Sander, 1983a]	Hydrogen bonds
DEFINE [Richards and Kundrot, 1988]	Geometry
P-CURVE [Sklenar et al., 1989]	Geometry
SSTRUC [Mizuguchi et al., 1998]	Hydrogen bonds
TCM [Colloc'h et al., 1993]	DSSP, DEFINE, P-CURVE
STRIDE [Frishman and Argos, 1995]	Dihedral angles and hydrogen bonds
PROMOTIF [Hutchinson and Thornton, 1996]	Hydrogen bonds
YASSPA [Novotny and Kleywegt, 2005]	Geometry
P-SEA [Labesse et al., 1997]	Geometry
PROSS [Srinivasan and Rose, 1999]	Dihedral angles
XTLSSTR [King and Johnson, 1999]	Geometry
STICK [Taylor, 2001]	Geometry
SECSTR [Fodje and Al-Karadaghi, 2002]	Hydrogen bonds
DSSPcont [Carter et al., 2003]	Hydrogen bonds
VoTAP [Dupuis et al., 2004]	Geometry
Zhang & Skolnick [Zhang and Skolnick, 2004]	Geometry
Taylor et al. [Taylor et al., 2005]	Geometry
KAKSI [Martin et al., 2005]	Dihedral angles and geometry
PALSSE [Majumdar et al., 2005]	Geometry
SEGNO [Cubellis et al., 2005]	Dihedral angles and geometry
β-Spider [Parisien and Major, 2005]	Geometry and contact energy
SKSP [Zhang et al., 2008]	STRIDE, KAKSI, SECSTR, P-SEA
APSA [Raganathan et al., 2008]	Geometry
PROSIGN [Hosseini et al., 2008]	Geometry
DSSP-PPII [Chebrek et al., 2014]	Dihedral angles and hydrogen bonds
PMML [Konagurthu et al., 2011]	Geometry
SABA [Park et al., 2011]	Geometry
SHAFT [Koch and Cole, 2011]	Dihedral angles and hydrogen bonds
SST [Konagurthu et al., 2012]	Geometry
DISICL [Nagy and Oostenbrink, 2014]	Dihedral angles
PCASSO [Law et al., 2014]	Geometry
PSSC [Zacharias and Knapp, 2014]	Dihedral angles and hydrogen bonds
ASSP [Kumar and Bansal, 2015]	Geometry
Kneller & Hinsen [Kneller and Hinsen, 2015]	Geometry
RaFoSA [Salawu, 2016]	Geometry
SACF [Cao et al., 2016]	Geometry
SCOT [Brinkjost et al., 2020]	H-Bonds, angles, and geometry

Table 2.2.: **Secondary structure assignment methods.** List of automated secondary structure assignment methods, and the atomic data utilized during assignment (Methodology).

residue to its single $C\alpha$ atom. This coarse-graining process almost always includes the removal of the side chain atoms in proteins. To reconstruct a realistic protein structure with all the atoms from these simplified models, a process known as protein backmapping is utilized [Jones et al., 2025]. Once solved, this backmapping reconstruction procedure assigns the position of all original atoms in the protein structure. For this computational reconstruction to take place, a library of known protein structures can be employed to identify which combination of ϕ and ψ angles corresponds to each amino acid in the sequence. Otherwise, physics-based or knowledge-based force fields [Brooks et al., 2009], or machine learning approaches [Kmiecik et al., 2007, Senior et al., 2020] can be utilized to guide the conformational search of the protein's atoms into their native state. Once all of the atomic locations are assigned, a three-dimensional model of the protein consistent to experimental data is achieved.

As previously stated, protein functionality is derived from the protein's shape and binding affinity to other molecules. To facilitate the acquisition of possible functionalities for a protein, researchers created computational databases that group proteins into superfamilies and families based on their folds, such as CATH [Sillitoe et al., 2021], SCOP [Andreeva et al., 2020], SCOPe [Chandonia et al., 2019], Pfam [Mistry et al., 2021] and ECOD [Schaeffer et al., 2017]. These databases use different methods and criteria to assign experimental structures of proteins from the Protein Data Bank (PDB) [Burley et al., 2017] to evolutionary superfamilies. CATH, for example, stands for Class, Architecture, Topology, and Homologous superfamily, and uses a combination of automated and manual procedures to group proteins into four levels of hierarchy. SCOP, on the other hand, stands for Structural Classification of Proteins, and uses a more subjective and expert-based approach to group proteins into seven levels of hierarchy. Both databases aim to provide comprehensive and accurate classifications of protein structures that can help researchers understand the relationships between protein structure, function, and evolution. The SCOPe database is an extension of the SCOP database and provides a more detailed classification of protein structures while correcting errors present in the original SCOP database. The Pfam database focuses on protein families and includes their annotations and multiple sequence alignment (MSA) results generated using hidden Markov models to provide a wide coverage of protein families through its accurate classification methods. Recently, Pfam has been migrated into InterPro but maintains its functionality as a protein family classification database. Finally, the Evolutionary Classification of protein Domains (ECOD) database groups PDB structures based on evolutionary connections and homology of protein domains. This process is distinct from solely structure-based classifications such as SCOP and CATH because it also utilizes remote homology for the classification of protein domains.

2.3. Machine learning

Machine learning (ML) is a discipline focused on the creation of algorithms that learn and improve their learning over time autonomously and without the use of human derived

instructions. ML algorithms use real-world measures in the form of observations or data as experience for a task they perform. These algorithms learn when their performance at the task improves by a measure that quantifies the amount of its improvement [Mitchell, 1997]. While there are many approaches to developing learning algorithms, we focus here on supervised learning. Supervised learning makes use of a collection of training data $(\mathbf{X}, \mathbf{Y})$, denoting pairs of input features and their expected outputs[1]. In a statistical sense, the goal when devising supervised learning algorithms is to create a model of a learnable function f from the specific training data points $(x, y) \in (\mathbf{X}, \mathbf{Y})$. Then, the function f should be able to approximate the expected output y from the respective input x, as shown in Equation 2.1.

$$f(x) \approx y \tag{2.1}$$

The function f is said to generalize well, if it also approximates the correct result for unseen data $(x, y) \notin (\mathbf{X}, \mathbf{Y})$.

2.3.1. Machine learning algorithms

There exists many machine learning algorithms to create prediction models. We investigate both parametric and nonparametric types of supervised ML algorithms. Parametric algorithms make assumptions on the distribution of the data, while non-parametric are distribution-free algorithms that do not assume a specific distribution for the data. For the parametric approach, we selected neural-type algorithms for their advances and continuous improvement in performance across various fields of study. We define neural-type algorithms as algorithms that generate neural network models [Kriegeskorte and Golan, 2019]. As for the nonparametric approach, tree-type algorithms were selected for their high interpretability and their ability to capture non-linear relationships without the need of feature scaling. We define tree-type algorithms as algorithms that generate decision tree models [Navada et al., 2011] or an ensemble of decision trees [Banfield et al., 2007]. A description of the select types of algorithms is given below.

1. *Tree*-type ML algorithms create a tree-like structure of optimal decision splits to determine an outcome.

2. *Neural*-type ML algorithms create an interconnected network-like structure of linear and non-linear functions.

The investigated tree-type algorithms include Decision Trees [Quinlan, 1986], Random Forests [Breiman, 2001], and Extremely randomized trees [Geurts et al., 2006] (ExtraTree). A Decision Tree works by splitting the data into subsets based on the value of input features. This process is repeated recursively, creating a tree-like structure where each internal node represents a decision based on a feature, each branch represents the

[1] Also known as labels

outcome of the decision, and each leaf node represents a possible outcome. Decision Trees are easy to interpret and visualize but can be prone to overfitting [Schaffer, 1991].

A Random Forest model consists of multiple decision trees during training and outputs a majority prediction from all trees. It introduces randomness by selecting a random subset of features for each tree and using bootstrap samples of the data. ExtraTree models are similar to random Forests but introduce more randomness to the decision splits. At each split, the decision threshold is drawn at random for each candidate feature and the best of these randomly-generated thresholds is selected. This results in a more diverse set of trees and can lead to a reduction in variance [Geurts et al., 2006]. ExtraTree models are computationally efficient and can handle large datasets effectively. The randomization also reduces overfitting and improves generalization. A depiction of the tree-type algorithms can be seen in Fig. 2.1.

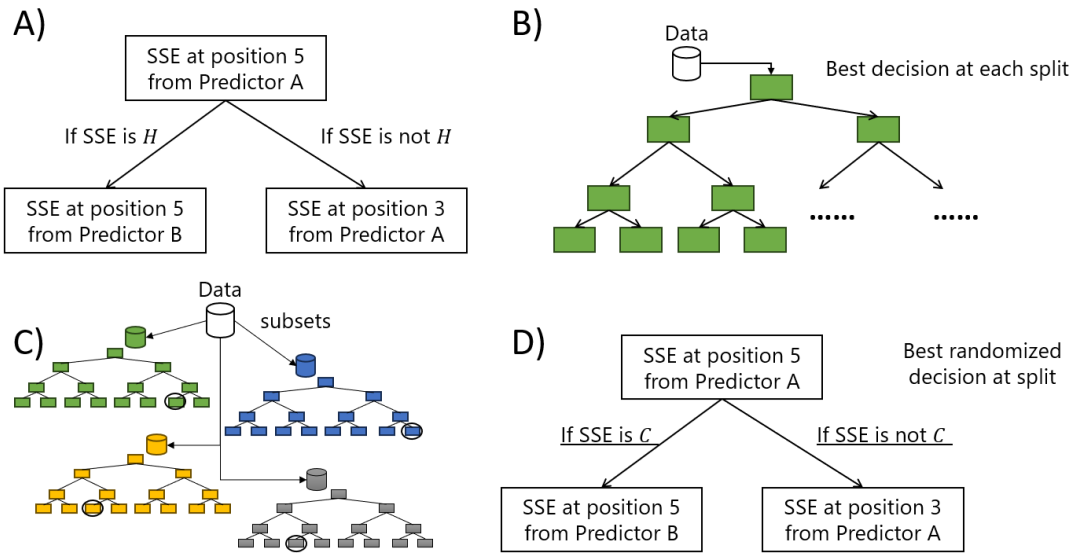


Figure 2.1.|: **Summary of tree-type machine learning algorithms.** A) Depiction of a decision threshold for tree-type algorithms. Squares contain the threshold while the arrows show possible decisions depending on the input. B) Depiction of a Decision Tree model, which has a tree-like (directed acyclic) graph. The best decision thresholds are created according to a quality criteria (Gini impurity). All data is utilized to create the tree. C) Random Forest model depiction showing the data being split and used to create multiple decision trees. A majority vote will become the final decision of the trees. D) ExtraTree model depiction, which functions similarly to a Random Forest but where decision thresholds are randomly selected and the best random threshold is utilized.

Neural-type algorithms create models that consist of layers of interconnected nodes, or neurons, which process its input data and passes that processed input to other neurons depending on activation functions in the network. Therefore, these models are known as neural networks. Each neuron processes its input by utilizing learnable parameters

in linear functions that get optimized to match the desired output through gradient descent. The activation functions introduce non-linearity, which allows the modeling of complex patterns of data. The number of neuron layers that process the data can become quite large, giving rise to *deep* neural networks. Here, we give a brief overview of common deep neural network architectures, including Fully-connected, Convolutional, Recurrent, and Transformer. More information on neural network architectures can be found in [Chitty-Venkata et al., 2022, Perez Martell et al., 2022].

Fully Connected Neural Networks [Rosenblatt, 1958] consist of neuron layers where each neuron is connected to every neuron in the previous and next layers. Convolutional Neural Networks [LeCun et al., 1989] are designed to process grid-like data, such as images. They consist of convolutional layers of neurons where learnable parameters are reused to capture local patterns in the data. Recurrent Neural Networks (RNN) [Graves, 2012] have neurons that form directed cycles, which reuse learnable parameters to maintain a "memory" of previous inputs. This makes RNNs suitable for tasks like language modeling and sequence prediction. Transformer Neural Networks are also designed to handle sequential data but differ from RNNs by using self-attention mechanisms [Vaswani et al., 2017] to process all elements of the sequence simultaneously.

2.3.2. Feature selection methods

Within the field of statistics, methods that quantify the dependency between variables can help in identifying significant input features that contribute to correctly predict expected outcomes in machine learning models. Common statistical methods for the selection of the most significant features include:

- χ^2 [Landis and Koch, 1977]: Compares the observed frequencies in each category to the frequencies expected. A greater difference between the observed and expected frequencies suggest a potential association or significance from the input to the predicted outcome.
- Mutual Information [Shannon, 1948]: Quantifies the amount of dependence between two variables. Greater values indicate higher certainty of the predicted outcome by the input.
- Analysis of Variance (ANOVA) [Landis and Koch, 1977]: Compares the mean and variance between and within groups to determine the statistical significance between the groups. This process identifies the amount of influence that groups have on the predicted outcome.

2.3.3. Model validation

Validating machine learning models is necessary to ensure their performance and their ability to generalize well on unseen data. Common validation techniques include the use of data splits and cross-validation. These techniques provide a robust validation for models,

although measures should be taken to ensure that a statistically representative sample of problem is obtained.

After a model has been trained, estimating its general performance on any possible input data is challenging. A perfect method would require the collection of all possible input data that the model could have as input. This perfect method might not be possible to achieve in a limited amount of time, or when such data is unavailable. A performance estimate could be done by acquiring data different to its training data. This could also prove difficult as the collection of data might be a time-consuming or resource-intensive process. Therefore, the original training sample data could be split into several datasets, where a subset sample would be utilized for the training of the method. For a large enough dataset, the data is commonly split into training, validation, and testing datasets, where 80% of the original sample is used for the training dataset, 10% for the validation dataset and 10% for the testing dataset. These splits are common examples, but the amount of data in each split will depend on the size of the dataset and algorithmic considerations such as training performance. The testing and validation datasets are also usually evenly split. The training dataset is used to optimize the models predictive effectiveness by minimizing the training error. This ensures that the model will accurately predict the training data.

Training a model for a longer period of time will not always result in obtaining a better performing model. When solely utilizing the training error, the model's output will only depend on the training dataset. Remember that machine learning requires models to be generalizable. As training progresses, and the training error decreases, the learnable function could become a mapping from the training dataset's input to its labels. Intuitively, one can think of the model as 'memorizing' the data. This process will lead to an increase in the error of the non-training datasets. This phenomenon is known as *overfitting*. Therefore, the validation dataset is used after a certain amount of training steps to estimate the model's ability to generalize. Thus, the validation dataset can help the training procedure stop before overfitting.

During the model training process, an expert can recognize when the training and validation errors have reached a desirable value to end the process. Otherwise, the process could be stopped to reassess the modeling methodology when the errors do not converge. When the training error $Error_T$ and validation error $Error_V$ stabilize to a low value ($Error_T \approx Error_V \approx 0$), we can assume that the model is trained and potentially generalizable. Once the model has been trained, the testing dataset is used to get the generalization estimate for the population as a whole by calculating the models testing error. When the testing error is also a low value ($Error_{TEST} \approx 0$), it is likely that the model is generalizable.

The previous methodology for training, validating and testing a models generalization works well if the testing data used is a representative sample of the problem. For cases where it is unknown if the small testing sample is representative, there exists a method called *cross-validation* that utilizes all the original sample data as testing data. The basic approach, known as n-fold cross validation, takes the training dataset and splits it into n datasets. This process is detailed in Algorithm 1,

Algorithm 1: Cross-validation

Input: Number of folds n
Data: Dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ with m samples

Split $\mathcal{D}$ into n equally sized folds $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$;
Initialize an empty array **Errors** of size n to store the performance for each fold;
for $i \leftarrow 1$ **to** n **do**
    $\mathcal{D}_{\text{TRAIN}} \leftarrow \mathcal{D} \setminus \mathcal{D}_i$;
    Split $\mathcal{D}_{\text{TRAIN}}$ into two sets:
     $\mathcal{D}^V_{\text{TRAIN}}$ with 20% of the samples
     $\mathcal{D}^T_{\text{TRAIN}}$ with 80% of the samples;
    $\mathcal{D}_{\text{TEST}} \leftarrow \mathcal{D}_i$;
    Train the model f on $\mathcal{D}^T_{\text{TRAIN}}$ and validate with $\mathcal{D}^V_{\text{TRAIN}}$;
    Validate f on $\mathcal{D}_{\text{TEST}}$ and compute error E;
    $\mathbf{Errors}_i \leftarrow E$;
end
Compute the average error $Error_{TEST} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{Errors}_i$;
Return the estimated error $Error_{TEST}$

There are many types of cross-validation approaches. The approach by Burman [Burman, 1990] that we used in our experiments is one that has been extended to account for imbalanced data in classification models by making folds that contain roughly the same amount data for each label. This method is called stratified cross-validation. It creates balanced folds by sampling from the training dataset while taking into consideration their labels. Other approaches can also be found in Burman's work [Burman, 1990].

There is a subfield of methods for model generalization called *regularization*. The main idea of these methods is to penalize the model from learning high-dimensional functions to avoid overfitting. Regularization methods vary depending on the learning algorithms being used. The simplest and most common regularization technique is early stopping. As previously noted, the learning process can be stopped when the validation error reaches an adequate value, or when the training and validation errors start to diverge significantly. Early stopping makes this an automatic process by keeping track of the validation error and ending the learning process when this error stops decreasing. Other forms of early stopping and regularization, such as data augmentation or dropout, can be found in [Tian and Zhang, 2022].

2.4. Related work

For many proteins, their structure can be fully characterized by all the positions of all the atoms in the molecule. Calculating the energies between all the atoms is computationally unfeasible because of the NP-complete nature of the problem [Unger and Moult, 1993]. Therefore, simplified structure models have been created to lower the complexity of the

problem to make models that solve the problem in a more practical and time efficient manner.

Early prediction models treated protein structures as lattices, which represented the single-chain polymers on a 2D or 3D lattice conformational space. The residues can only lie on the lattice vertices and the bonds between residues lie on the edges of the lattice. The lattice structure is *self-avoiding*, which means that residues are not allowed to intersect. Most lattice models study proteins made up of only two types of residues, hydrophobic (H) and polar (P). This simplified interaction potential, known as the HP model, is designed to reflect the free-energy gain seen in real proteins when hydrophobic residues are removed from contact with water. Other models have also been studied, which consider van der Waals-type interactions between residues, or which try to model more accurately the known interactions between the different amino acids.

The small resolution of the lattices in the early models limited the positions of the atoms, which in turn made it difficult to interpret secondary structure elements present in the protein. With the increase in the accumulation of sequence data in the late 20th century, many attempts have been made to predict protein secondary structures from their primary sequence using a variety of parameters such as amino acid frequency [Blout et al., 1960, D.R., 1964, Havsteen, 1966, Krigbaum and Knutton, 1973] and energy calculations [Kotelchuck and Scheraga, 1969]. Secondary structure prediction involves the classification of protein backbone structural motifs for each amino acid. The most common classifications involve three-state and eight-state secondary structure. Three-state classification contains 3 types of secondary structures, while eight-state classification contains 8 types of secondary structures allowing for more fine-grained structures. As a result of these early studies, it was possible to somewhat identify and quantify the secondary structure of proteins based on sequence data alone. One of the most common secondary structure prediction methods at that time was proposed by Chou and Fasman [Chou and Fasman, 1979]. In this method the researchers conducted a statistical survey of 15 proteins in which the α-helix, β-sheet and β-turn conformational potential of all 20 standard amino acids was established. A set of empirical rules were then derived, which allowed for the determination of the folding of the secondary structural regions in the proteins. The method is simple and easy to use and is reported to have reasonable accuracy when compared to results obtained from X-ray data. However, some of the rules are open to interpretation, and as a result various authors [Burgess and Scheraga, 1975, Nishikawa, 1983, Kabsch and Sander, 1983b] had only limited success using the Chou and Fasman method. Therefore, Pham [Pham, 1981] wrote a computer program for the Chou and Fasman method to expedite the calculations and to clarify some of the ambiguities in that method. Although these early results looked promising, Kabsch and Sander [Kabsch and Sander, 1983b] examined various methods for the prediction of secondary structure of 62 proteins but found that none of the methods predicted better than $\sim 55\%$ of the residues correctly for three-state secondary structure (Q3) prediction; with the Chou and Fasman method correctly predicting only 50% of the residues.

2.4.1. Secondary structure prediction

With the advancements of computing and machine learning, significant progress on structure prediction was achieved. The highest three-state classification accuracy without relying on structure templates was measured at around 85% [Yang et al., 2018]. These improvements came from increasingly larger databases of protein sequences and structures for training, the use of template secondary structure information and more powerful deep learning techniques. As we are approaching the theoretical limit for three-state prediction of 90% [Yang et al., 2018], focus shifted on solving the eight-state prediction problem, which is much more complicated and challenging. The theoretical limit of secondary structure prediction for eight classes is not well established, but current template-less methods manage around 70% [Sidi and Keasar, 2020]. Although controversy exists [Zacharias and Knapp, 2014, Drew and Janes, 2019] around the eight classes as they are not equally distributed and some are more difficult to predict than others, further improvements are still being achieved.

The field's improvements started to slow down at the fifth iteration of the 'Critical assessment of techniques for protein structure prediction' (CASP5) competition, which took place in 2002. Initially, CASP [Kryshtafovych et al., 2023] aimed to advance the prediction and modeling of protein structure from amino acid sequences, but later expanded to other macromolecules. CASP5 was the last meeting that accepted secondary structure predictions. Their methods abstract[2] details each ranked group's methodology for the different predicted categories. The top 10 predictors' results were posted in their meeting manuscript [Aloy et al., 2003]. The models were evaluated on using segment overlap measure [Zemla et al., 1999], which is currently still being utilized in the field.

Current secondary structure prediction models are heavily influenced by the CASP5 methods, including SSPro [Magnan and Baldi, 2014], PsiPred [Buchan and Jones, 2019], and its derivatives. During the early 2000s, machine learning models were the primary predictors being developed because of their increased prediction accuracy. These models take into account co-evolutionary data for each protein to predict the secondary structure. Since CASP5 there have been multiple reviews surveying the status quo in the field [Ho et al., 2021, Smolarczyk et al., 2020, Jiang et al., 2017].

We focus on locally usable protein secondary structure prediction software instead of web servers to avoid any potential server limitations and facilitate privacy-oriented applications required in the medical field. As the most recent review [Ho et al., 2021] explains, predictive performance is similar in recent models. The biggest difference between the models can be seen from predicting the eight-state secondary structure. We carefully selected publicly available top models for eight-state secondary structure prediction by considering models from all previously mentioned reviews and additionally models that were published that outperformed top models but were not included in these reviews. These models include ssPro8 [Magnan and Baldi, 2014],

[2] https://predictioncenter.org/decoysets2019/meeting.cgi?casp=CASP5

Spot-1D [Hanson et al., 2019], RaptorX Property Predictor [Wang et al., 2016a], SPOT-1D-Single [Singh et al., 2021], and SPOT-1D-LM [Singh et al., 2022]. We had to leave out other models that are not currently available or required extensive training in the case of deep learning models without publicly available pre-trained models. Although relevant since the inception of secondary structure predictors, PsiPred had to be excluded as it is only a three-state predictor. Similarly to PsiPred, other three-state secondary structure that were discarded include PHD [Rost et al., 1994a], SOPMA [Geourjon and Deléage, 1995], SPINE-X [Faraggi et al., 2012], SPARROW [Bettella et al., 2012], and JPRED [Drozdetskiy et al., 2015]. Other eight-state secondary structure prediction software that were disregarded include SPIDER3 [Heffernan et al., 2017] and SPIDER3-Single [Heffernan et al., 2018] as SPOT-1D and SPOT-1D-Single have superseded these tools. Software packages that are not readily available include MUFOLD-SS [Fang et al., 2018], SCORPION [Yaseen and Li, 2014], and eCRRNN [Zhang et al., 2018].

PsiPred [Buchan and Jones, 2019] is one of the long-standing secondary structure prediction software with over 20 years of development. PsiPred was created as a web server but is also offered as a standalone software. PsiPred was originally a multi-layer neural network based method that utilizes position specific scoring matrices (PSSM) generated by PSI-BLAST [Jones, 1999]. PSSM gives PsiPred the co-evolutionary information required to predict a secondary structure by characterizing protein domains. This co-evolutionary information was later employed by most secondary structure prediction methods. Aftwerwards, PsiPred was updated to a deeper neural network architecture with two hidden layers rather than just one, and with rectifier activations rather than sigmoid. The other change is that the input window has been extended from 15 residues to 33 thanks to using sparse connections between the input layer and first hidden layer [Buchan and Jones, 2019].

ssPro8 [Magnan and Baldi, 2014] utilizes a three-stage workflow. As with other classic tools, it uses PSI-BLAST to derive multiple sequence alignment and profile probabilities. Also, it uses an ensemble of 100 Bidirectional Recursive Neural Networks (BRNNs) trained on the data to generate a first set of probability predictions for each secondary structure class. Finally, ssPro8 derives secondary structure predictions from the ensemble by using sequence-based structural similarity in regions with more than 45% sequence similarity.

RaptorX Property Predictor [Wang et al., 2016a] was created as a web server, but later offered as a standalone software. It uses a deep learning method with a Convolutional Neural Field architecture [Wang et al., 2016b]. This architecture combines the advantages of both Conditional Random Fields and Convolutional Neural Networks, which captures not only a complex sequence-structure relationship, but also models secondary structure correlation among adjacent residues.

SPOT-1D [Hanson et al., 2019] is a model with an ensemble of nine Bidirectional Recurrent Neural Networks and Residual Networks hybrid models to identify and propagate short and long term dependencies throughout the sequence. This model predicts structural data from sequence data consisting of two evolutionary profiles from three iterations of

PSI-BLAST and from HHBLITS. The predicted data are used to create an amino acid contact map through SPOT-Contact. Finally, the amino acid sequence is also utilized to predict their physicochemical properties. All these data are utilized as input to the ensemble of neural networks which produces the final secondary structure prediction. Even though SPOT-1D was not part of the latest reviews [Ho et al., 2021, Smolarczyk et al., 2020, Jiang et al., 2017], it was selected since it was compared to other models and found to be well performing among the best protein secondary structure prediction software.

SPOT-1D-Single [Singh et al., 2021] is an ensemble of three neural network architectures. Similar to SPOT-1D, the architectures are variants of Residual Networks and Bidirectional Recurrent Neural Networks. The main difference is that the input is only a single amino acid sequence, without the evolutionary data from PSI-BLAST, HHBLITS or additional data from SPOT-Contact.

SPOT-1D-LM [Singh et al., 2022] is an ensemble of large language models that have recently improved prediction accuracy for sequential tasks in many fields. The pre-trained large language models utilized here consist of ESM-1b and ProtTrans which turn the initial single amino acid sequence into features for a deep learning model with the same ensemble architecture as SPOT-1D-Single. The ensemble architecture then outputs the secondary structure prediction. The main advantage for these large language models is the included evolutionary information built into them from their training phase. SPOT-1D-LM managed to get comparable prediction performance to models with evolutionary information, such as SPOT-1D. This, of course with the added speed benefit from not requiring computationally expensive PSI-BLAST and other evolutionary information procuring software.

2.4.2. Tertiary structure prediction

Tertiary structure prediction methods encompass statistical- and machine learning- based models. The models can be categorized into two main categories: Physics-based and Template-based. Although they do not account for all models, they are the most well regarded before the deep learning models substantial improvement. These early models are described below.

Rosetta [Alford et al., 2017] utilizes energy minimization and relaxation with force fields, and molecular dynamics simulation to predict the structure of a protein. This physics-based approach is simplified as the current computational capabilities to solve these problems would be insufficient for proteins that are not small.

HHPred [Söding, 2005] uses protein homology detection to predict a protein structure by searching a wide choice of databases such as PDB, SCOP, Pfam, SMART and CDD. It implemented a pairwise comparison of profile hidden Markov models (HMMs) for the homologous proteins.

I-TASSER [Zhang, 2008] (Iterative Threading Assembly Refinement) is a protein structure prediction method that uses a combination of comparative modelling, threading, and molecular dynamics simulation to predict the three-dimensional structure of proteins. It

has been a widely used and well-established method for predicting protein structures for its high performance during CASP competitions [Huang et al., 2014]. It identifies a set of protein templates that are structurally similar to the target protein and follows with comparative modelling and molecular dynamics simulations to get a final prediction. Current deep learning methods have outperformed its structure prediction capabilities. Subsequently, the benefits of deep learning methodologies has been integrated to I-TASSER for improved performance [Zheng et al., 2023].

RaptorX [Peng and Xu, 2011] is also a statistical method for template-based protein structure predictions that improves alignment accuracy by exploiting structural information from multiple templates. It can also align templates and sequences through single and multiple template threading, and alignment quality prediction. The structure prediction is done by a nonlinear scoring function to combine homologous information (i.e., sequence profile), and template structure information in a very flexible way.

Predicting a protein's three-dimensional structure is currently done mostly with deep learning models. These models are trained on large datasets of known protein structures and use this training data to make predictions about the structures of new proteins. These models can be categorized by the amount of co-evolutionary information utilized to achieve a prediction. The categories are single sequence-based, PSSM-based, MSA-based, and MSA+End-to-end differentiable. Some of the most recent and notable protein structure prediction tools in these categories include the following:

RaptorX [Xu et al., 2021] was the first tool to transition into a deep learning-based structure prediction tool. RaptorX predicts protein secondary and tertiary structures, contact and distance map, solvent accessibility, disordered regions, functional annotation and binding sites. The structure prediction is done through the prediction of contact and distance between amino acid residue pairs from from multiple sequence alignment profiles.

AlphaFold2 [Jumper et al., 2021] (AF2) is a deep learning model that achieved near-atomic accuracy on the 14^{th} iteration of CASP. It utilizes evolutionary data of similar proteins to predict the distance matrix of the target protein. The distance matrix represents the distances between all pairs of amino acids in the protein and can then be used to generate a three-dimensional model of the protein structure. AF2 combined the use of end-to-end differentiability and transformer-based networks for protein structure prediction with multiple sequence alignment to exploit co-evolutionary data as templates for the prediction. AF2 predictions were experimentally validated after it was utilized to successfully obtain a small molecule hit compound for Cyclin-dependent Kinase 20 [Ren et al., 2023]. Alphafold [Senior et al., 2020] is the previous version of AF2 which used convolutional networks instead and was also not end-to-end differentiable. This architectural difference in the deep learning model had substantial implications in their performance difference. Following the success of Alphafold, the researchers behind Rosetta created a deep learning model called trRosetta [Du et al., 2021] that functions very similarly to Alphafold and consequently create RoseTTAFold [Baek et al., 2021] in a similar manner to AF2.

ColabFold [Mirdita et al., 2022] is a set of open source Python scripts that makes other tertiary structure prediction models, such as AF2 and ESMFold [Lin et al., 2023], available

for use in GPU accelerated computing environments, mainly through Google Colaboratory[3]. The scripts have also been made available for use in a local computing environments by Yoshitaka Moriwaki[4]. ColabFold also manages to improve the speed of the prediction methods, without a significant loss of predictive performance.

RGN [AlQuraishi, 2019] and NEMO [Ingraham et al., 2019] predict protein structures directly from a position specific scoring matrix (PSSM) or primary sequences. These systems were 'end-to-end differentiable' as the complete process (input to output) was done in a neural network and optimized through differentiable primitives (derivation of PSSM). RGN implicitly folded proteins using recurrent neural networks then sequentially placed the backbone atoms. NEMO folded proteins using a neural network that could explicitly simulate 3D structures and subsequently refine the predicted structure.

Predicting a protein structure without co-evolutionary information is currently trying to be solved through the use of large language models. RGN2 [Chowdhury et al., 2022] is an end-to-end differentiable system that predicts protein structure from single protein sequences by using the protein language model named AminoBERT as a submodule. It involves two primary innovations relative to RGN and other machine learning-based structure prediction approaches. First, it uses the amino acid sequence itself as the primary input as opposed to a PSSM. Second, it makes use of a transformer protein language model to learn structural information and uses a geometric module to generate the backbone structure.

Similar to RGN2 and one of the first models to utilize large language models for protein folding is ESMFold [Lin et al., 2023]. It is also a deep learning-based method for predicting protein structure from sequence. It is based on the ESM-1b language model, which is a transformer-based language model that was pre-trained on a large corpus of protein sequences. ESMfold uses the ESM-1b model to predict the distance matrix between all pairs of residues in the protein sequence and then uses this distance matrix to generate a 3D structure of the protein.

Advancements in protein folding from deep learning models can be attributed mostly to the addition of geometric inductive biases to avoid learning from insufficient data. Advancements occurred also from the use of topological group theory, e.g. SE(3) [Wu and Carricato, 2020] invariant or equivariant techniques, because of their differentiability properties. Finally, further advancements were made through the use of Transformers and Message Passing Neural Networks(MPNN) or Graph Neural Networks (GNN) to implement these geometric constraints and biases.

2.4.3. Protein similarity

Protein structure similarity tools are computational methods that compare and align the three-dimensional shapes of proteins. They can be used to infer functional and evolutionary relationships, identify structural motifs, and classify proteins into families. There are

[3] https://colab.research.google.com/
[4] https://github.com/YoshitakaMo/localcolabfold

different types of protein structure similarity tools, depending on whether they perform local or global alignment. Local alignment tools find the best matching regions between two or more structures, while global alignment tools find the best overall superimposition of the entire structures. Some examples of protein structure similarity tools are:

- **DALI** [Holm, 2020]: a local alignment tool to compare protein structures in 3D. It can search the PDB for similar structures or perform pairwise or multiple alignment of user-submitted structures. Comparing protein structures can be done in multiple ways depending on the abstraction level of the 3D structural data. DALI makes use of distance maps, distance matrices containing all pairwise distances between residue centers ($C\alpha$) in a protein chain. These matrices are a 2D representation which can be used to reconstruct the 3D structure of the protein with the exception of its overall chirality. To align two pairs of proteins, the first step is dividing the distance matrices into overlapping submatrices of a fixed size to find pairs of similar contact patterns in a similar manner to the convolution operation. These contact patterns are then extended through association and iteratively improved by a random walk algorithm. To quantify similarity of patterns, DALI uses the elastic similarity score which is tolerant to cumulative gradual geometrical distortions.

- **FATCAT** [Li et al., 2020]: a local alignment tool that uses a flexible algorithm to find optimal alignments of protein structures. It can handle both rigid and flexible structures and can compare structures from the PDB or user-submitted coordinates. FATCAT also utilizes distance matrices to align proteins by looking into aligned fragment pairs (AFP), which are similar to the overlapping submatrices method from DALI. FATCAT detects an AFP if the RMSD of the fragment is less than a certain threshold. The AFPs can be seen as secondary structures within the protein and FATCAT connects them when the two proteins have similar secondary structures in a short distance by their 3D coodinates matching or by introducing a twist to the protein to create a better alignment. It is important to note that FATCAT tries to find the optimal structure alignment with the least number of twists. These twisting rearrangements is what makes this method a flexible structure alignment method since the protein's structure is being altered to find the match. This is important because proteins are not rigid molecules and the alterations might fit reality better.

- **GANGSTA** [Guerler and Knapp, 2008]: a global alignment tool that uses a two-level hierarchical approach to maximize pair contacts and relative orientations between secondary structural elements (α-helices and β-strands) with a genetic algorithm. Residue pair contacts from the best SSE alignments are then optimized. The method is able to detect significant structural similarity of functionally important folds with non-sequential SSE connectivity and has comparable performance for structure alignments with strictly sequential SSE connectivity to other structure alignment methods. This is done through the GANGSTA score and Contact Map Overlap approach by analyzing the similarity of contact maps. Contact maps capture a 3D structure in condensed form, representing the 3D protein conformation as a symmetrical, square, boolean matrix of contacts.

- **CE** [Shindyalov and Bourne, 1998]: a global alignment tool that uses a heuristic heavy combinatorial expansion algorithm to find the longest continuous segments of similar structures. The alignment consists of the longest continuous path of aligned fragment pairs (AFPs) of size 8 in a similarity matrix. The AFPs are paired through a similarity measure while the extension of these pairs depend on similarity and gap length. Similarity is evaluated through three distance measures on residue centers or $C\alpha$: one to one residue distances, all neighboring residue distances, and superimposed rigid structures by RMSD. An AFP extension has 3 possibilities: consider all AFPs to extend, only the best AFP or some intermediate strategy. The longest alignment path is then evaluated through a statistical z-score. Most accurate alignment is obtained through three conditions that choose the candidates to extend, then the best one, and finally whether to extend or terminate the path. In case the z-score is too high, further optimization is done by filtering the 20 best paths by RMSD and testing relocations for the gaps of the best path. Finally performing dynamic programming on the distance matrix of the best path and continuing this process only if alignment length is less than 95% of original or RMSD is less than 110% at that point.

- **LGA** [Zemla, 2003]: a global alignment tool that uses a combination of local and global methods to align protein structures. LGA uses a scoring function which utilizes two components: LCS and GDT. LGA was established for the detection of regions of local and global structure similarities between proteins. LCS measures the length of the longest continuous segment of residues that are structurally aligned between two proteins. GDT measures the percentage of residues that are within a certain distance cutoff between two proteins. These local and global components allow LGA to rank the level of similarity between two structures.

- **TM-align** [Zhang and Skolnick, 2005]: a global alignment tool that uses a dynamic programming algorithm to align protein structures based on $C\alpha$ atoms distances and secondary structures through aligned fragment pairs. The initial alignment is done through dynamic programming of the these secondary structures. Then it does gapless matching by threading the smaller protein to the larger one and scoring them by TM-score. Lastly a half/half combination of the two previous matrices are utilized to align the structures using dynamic programming. After this initial alignment, it iteratively rotates the structures through a TM-score rotation matrix repeated until the alignment becomes stable (usually 2 to 3 iterations). mTM-align [Dong et al., 2018] extends TM-align to perform multiple structure alignment based on a guide tree and a progressive alignment strategy. mTM-align can also calculate multiple structure alignment scores (TM-scores) and generate consensus structures.

- **ProBis** [Konc and Janežič, 2010]: a local alignment tool to detect binding sites that may lack sequence and global structural conservation by transforming the surface residues (all atoms) into a graph and using a maximum clique algorithm to find the local structural similarities to the protein binding site.

Specialized tools for finding specific similarities within proteins also exist. PDBsphere [Zemla et al., 2022] is an alignment tool that assesses similarity of local protein

regions relevant to ligand binding. It comprises an exhaustive library of protein structure regions ("spheres") adjacent to complexed ligands derived from the PDB. The PDBspheres library contains more than 2 million spheres, organized to facilitate searches by sequence and/or structure similarity of protein-ligand binding sites or interfaces between interacting molecules. PDBSpheres uses LGA structure similarity tool detect structure similarities between a protein of interest and the library spheres.

Protein structure similarity tools are helpful for functionally similar proteins because they can reveal common features that are related to their function, such as folds, domains, cavities and interfaces. They can also suggest possible functional analogies or homologies between proteins that share structural similarity but not sequence similarity. Sadowski et al. [Sadowski and Taylor, 2012] compared structure alignment methods through an inconsistency measure, which scores aligned triplets of residues in each protein through binary classification. It was found that all methods show greater consistency for functional methods than non-functional ones. FATCAT produced good structural scores but was highly inconsistent. On average TM-align was both consistent and geometrically sensitive. The most significant factor to inconsistency was found to be the gap scoring which does not account very well for insertions or deletions (indels). Inconsistencies were also found in repetitive sites because of the ambiguities for alignment algorithms in these periodic structures (e.g. α helices).

2.4.4. Protein structural data

Obtaining the data to create models that solve protein structure problems is an enormous task that has been undertaken by the protein structural community for more than 50 years. Data at all hierarchical levels of protein structure that can be utilized and analyzed to solve structural problems are contained within databases. These databases allow for the research community to improve the field's knowledge through their accessibility and substantial volume of data. The databases are categorized as follows,

Sequence databases:

- NCBI Protein database
- UniProt
- Prosite
- InterPro
- Kyoto Encyclopedia of Genes and Genomes
- NCBI Clusters of Orthologous Genes
- EMBL Simple Modular Architecture Research Tool

The data contained in these databases are an aggregate of Swiss-Prot (SP) [Bairoch and Boeckmann, 1994], its supplementary database SP-TrEMBL [Bairoch and Apweiler, 1996], and Ensembl [Hubbard et al., 2002] from the European Molecular Biology Laboratory (EMBL) with added metadata such as protein functionality and classification. The American counterpart to the European organization EMBL is the National Center for Biotechnology Information (NCBI) of the National Institute of Health (NIH).

Structure databases:

- RCSB Protein data bank
- Structural Classification of Proteins (SCOP)
- SCOP extended
- Class, Architecture, Topology, and Homologous superfamily database
- NCBI Conserved domain database
- Protein Common Interface Database
- Swiss-Model
- AlphaFold database

The data in the structure databases contains experimentally obtained structures for proteins using a variety of methods. The most common methods of obtaining these structures are X-ray crystallography, different versions of electron microscopy and tomography, as well as nuclear magnetic resonance methods. These databases can also contain metadata and sequence data when available to facilitate the structural analysis.

Machine learning-specific datasets:

- ProteinNet
- SidechainNet

The data here contain sequences, structures (secondary and tertiary), MSAs, PSSMs, and standardized training, validation, and test splits. This data is ready for use in common machine learning frameworks like Scikit-learn[Pedregosa et al., 2011], PyTorch [Paszke et al., 2019] and Tensorflow [Abadi et al., 2016].

2.5. Software libraries and packages

This section addresses the software required for the technical implementation of our contributions in this thesis. It describes the open source libraries, as well as individual packages available for creating and evaluating the empirical tests that we present in this thesis.

2.5.1. Python

Python[5] is a high-level, general-purpose programming language with a design philosophy emphasizing code readability. Its accessible nature has made machine learning researchers adopt this programming language not only for its readability, but also for its huge community. The libraries made by the community make code prototyping a fast and efficient task. Specifically for scientific computing projects, there exists different software packages that make the programming environment setup an easy task. Anaconda[6] distribution is one such software package, which contains Python, along with many scientific libraries such as NumPy and Matplotlib.

2.5.2. Biopython

Biopython[7] is set of libraries and applications for bioinformatics using the Python programming language. It contains many tools needed for work in the bioinformatics field. Biopython is a project of the Open Bioinformatics Foundation, which is a nonprofit, volunteer-run group that promotes open-source software development within the biological research community. It offers data structures designed for genomic analysis, as well as tools for use in population genomics and structural bioinformatics. It also contains an interface to BioSQL made for supporting a shared database schema for storing sequence data.

2.5.3. SciPy

SciPy[8] is a Python-based ecosystem of open-source software for mathematics, science, and engineering. This ecosystem of packages is often used as a Python alternative to MATLAB because of its more modern and organized nature.

2.5.4. NumPy

NumPy[9] is an open-source project led by volunteers made for scientific computing with Python. It contains data structures for numerical computations such as an N-dimensional array object with many useful functions for efficient mathematical and logical operations on arrays and matrices. It is mostly useful for calculating linear algebra functions and Fourier transformations. This is all made efficient with its tools for integration of Fortran and C/C++ code, which offer high performance computations. NumPy is part of the SciPy ecosystem.

[5] https://www.python.org/
[6] https://anaconda.org/
[7] https://biopython.org/
[8] https://scipy.org/
[9] https://numpy.org/

2.5.5. Matplotlib

Matplotlib[10] is an open-source project that is part of the SciPy ecosystem, and supported by a community of volunteers. It is a Python 2D plotting library which produces publication quality figures and interactive plotting environments. It makes the creation of plots such as histograms and scatterplots an effortless process.

2.5.6. Pandas

Pandas[11] is an open-source library providing high-performance, easy-to-use data structures and data analysis tools for Python. It is actively supported by a community of volunteers and, like Matplotlib, is also a part of the SciPy ecosystem. It is used mostly for its Input/Output tools in the handling of large volumes of data. It provides many high-performing data transformation functions as well as pre-processing tools for data intensive tasks.

2.5.7. Scikit-learn

Scikit-learn[12] is an open-source machine learning library built on top of SciPy and maintained by a group of volunteers. It provides tools for efficient predictive data analysis including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

2.5.8. Julia

Like Python, Julia[13] is a high-level, general-purpose interpreted programming language designed for scientific analysis and computation. Unlike Python, it is designed as a high-performance language that can efficiently be compiled to native code as it is executed. This compilation procedure allows computation performances comparable to compiled languages, e.g. C. While the compilation step can take a non-trivial amount of time, this process is only required when the compiled version is not contained within volatile memory.

2.5.9. Neural Network Libraries

Common libraries that help in implementing neural networks are described here. These libraries contain many desirable functions when working with neural networks. This makes neural network research efficient and less error-prone during implementation.

[10] https://matplotlib.org/
[11] https://pandas.pydata.org/
[12] https://scikit-learn.org/stable/
[13] https://julialang.org/

2.5.9.1. Tensorflow

Tensorflow[14] is an 'end-to-end' open-source platform for neural networks. 'End-to-end' means that it is possible to use this program alone to create the whole machine learning pipeline. The pipeline refers to reading the data, processing it, and learning from it to create neural network models. After the model creation step, it can deliver useful user-friendly results or predictions to the end user. As mentioned previously, Tensorflow uses a graph-based approach for its backpropagation computations. Tensorflow's core library is made to develop and train neural network models at a low level, meaning that it is possible to work on the mathematical elements and array expressions that make up the layers of the neural networks. Tensorflow also contains an implementation of the Keras API specification, which is described below.

2.5.9.2. Keras

Keras[15] is an open-source deep learning library for Python. It is a high-level neural network API capable of running on top of TensorFlow, with a focus on fast experimentation. It allows for fast prototyping thanks to its implementations of popular deep learning architectures and layers. It also helps in the training of the neural networks, having implementations of popular optimizers, activation functions, and other pre-processing methods. It also contains popular datasets for use in the model's training and testing process.

2.5.9.3. PyTorch

PyTorch[16] is an 'end-to-end' open-source neural network framework focusing on accelerating the path from research prototyping to production deployment. Similar to Tensorflow, it is used to develop and train neural network models at a medium level. PyTorch lies between Tensorflow and Keras, providing more control and flexibility than Keras, and offering more abstraction than Tensorflow's fine or granular level of control over the neural network model's creation. Unlike Tensorflow, PyTorch uses a tape-based approach for its backpropagation computations, making runtime debugging an easy task. More recently, both PyTorch and Tensorflow allow for the transition between tape-based, or what they call 'eager' mode, and graph-based approaches.

2.5.9.4. PyTorch lightning

Like Keras, PyTorch Lightning[17] is an open-source deep learning library that provides a high-level interface for PyTorch. It allows for fast prototyping due to the inclusion of standardized implementations of training procedures, and popular optimizers, activation func-

[14] https://www.tensorflow.org/
[15] https://keras.io/
[16] https://pytorch.org/
[17] https://lightning.ai/docs/pytorch/stable/

tions and other pre-processing methods. It allows for the effortless transformations of data types and switching of accelerators, such as Graphics Processing Units or Tensor Processing Units.

Part II

Contributions

Chapter 3

Contributions Overview

Contents

In this chapter, we introduce the contributions we provide during our investigation of protein structure prediction methods. We conceptualize the current performance of structure prediction methods on single amino acid mutation data. This conceptualization using secondary structure allows us to view the problem differently than in previous studies.

3.1. Single amino acid mutation conceptualization

Protein primary structure can be transformed through different biological procedures. Single amino acid mutations can occur as an amino acid being replaced for another amino acid, through a newly inserted amino acid, or by deleting an amino acid. Insertion or deletion of amino acids in the protein, known as indels, affect the length of the protein, while replacement mutations preserve its length. To simplify and focus the effects of single amino acid mutations on protein structure, we concentrate on single amino acid mutations that preserve the length of the protein. This allow us to perceive the effects of amino acid replacements without the physical implications on protein length.

Highly accurate protein structure prediction methods have been developed, closely matching atomic locations of experimentally-obtained structures. The evaluation procedures that have deemed these methods as accurate stem from measures that consider the overall structure of the protein. These measures emphasize the global structure of the protein. Although the local structures within the protein are considered, the utilization of a single score value for a protein lacks the resolution to evaluate all aspects of the protein structure. To address the global emphasis, highly accurate methods employ local measures at the residue level, where the location of an amino acid is compared relatively to all other amino acids within the protein. Such residue-level measures give a notion of how accurate the location of an amino acid is in regards to others nearby. To evaluate the effectiveness of these local measures in the context of the complete protein, an aggregated score from

the local measures is produced, leading to the obscuring of the local scores. This leads to complications while trying to elucidate the mutational effects on the protein structure.

We propose investigating protein structures in a simplified manner, as to provide the optimal conditions to successfully predict the mutational effects. To elucidate the mutational effects on protein structure, we utilize secondary structure to focus solely on protein backbone atoms. Protein three-dimensional structure is inherently noisy due to the dynamic nature of atomic positions. Protein secondary structure helps mitigate the noisy atomic coordinates and atomic-level fluctuations by discretizing the atomic-level details, thereby elucidating mutational effects in the protein's architecture or backbone structure. This removal of confounding atomic-level variations introduced by experimental and physical factors can increase the reliability of observed structural changes.

The simplified conditions are the following:

- Mutations are solely considered through single amino acid mutations without indels.
- Protein structural changes through the aforementioned mutations are solely considered through secondary structure.

If through these conditions, prediction methods are unable to display the correct structural changes from a single amino acid mutation, we can infer a deficiency in their methodology. These deficiencies, if any arise, can provide clear pathways for the improvement of structure prediction methods. In the next chapters, we identify such deficiencies in state-of-the-art protein structure prediction methods, and create a strategy to address and rectify these mutational deficiencies.

3.2. Our Contributions

This section summarizes the contributions of this dissertation.

Contribution 1. *Obtaining a mutational dataset* with experimental data to evaluate state-of-the-art prediction methods. Experimental data is available for the different hierarchical protein structure levels. Mutational data is commonly available for primary structure, where potential functionality can be obtained through molecular binding testing. These assays include potential binding agents to test the bonding capabilities to certain molecules. This process cannot account for all possible molecules and result solely in the binding capacity of the protein. To obtain secondary and tertiary structure, additional experimental methods must be employed, such as X-ray crystallography. Secondary structure can be inferred from tertiary structure. Therefore, obtaining all possible tertiary structure for mutated proteins is necessary to obtain a wide breadth of mutations that cause local and global structural changes as required for RQ1 and RQ1.1.

Contribution 2. *Benchmarking a set of diverse structural prediction methods*, including state-of-the-art secondary and tertiary structure prediction models to analyze their behavior on mutation data. Their results can be simplified to backbone structural changes

utilizing secondary structure, allowing us to perceive local and global structural changes throughout protein backbone to answer RQ1.2 and RQ1.3. This simplification allow us to use secondary structure measures, as well as devise mutational measures as part of RQ2 and RQ2.1. These mutational measures can specifically calculate the SSE changes due to mutations by rephrasing and transforming the problem from direct structural comparison to the comparison of structural changes.

Contribution 3. *Providing insights into the deficiencies* of current state-of-the-art structural prediction methods by focusing on single amino acid changes. Comparing experimental data statistics against predicted ones, we can calculate distribution differences from their structural changes due to single amino acid mutations. Together with the benchmark results, this can allow us to check which mutations are degrading the performance of prediction methods. This contribution is thus linked to RQ2 and RQ3.

Contribution 4. *Proposing a novel secondary structure refinement strategy* that relies solely on single amino acid mutational data. We do this to answer RQ3.1, as protein structure prediction methods are still imperfect. Deficiencies in inaccurate predictions can be further improved by ensembling current prediction methods. Training this ensemble on mutational data, where deficiencies are found, can produce an ensemble that not only takes into consideration the predictions of each method of the ensemble, but also enhances their capabilities by removing highly improbable structural changes.

Contribution 5. *Implementation of Mut2Dens*, an ensemble model of methodologically diverse predictors following our refinement strategy, which mitigates individual predictor weaknesses while achieving reliable and consistent mutation-focused predictions. This contribution transforms the theoretical refinement strategy into a practical and useful tool that helps addressing RQ3.1.

Contribution 6. *Evaluation of Mut2Dens* using our refinement strategy on mutational and non-mutational protein datasets to assure consistent predictions across both types of protein data. By evaluating Mut2Dens and comparing it to current state-of-the-art prediction methods, we can obtain empirical results that gives further evidence for RQ3.1 by showing possible improvements in protein structure prediction.

Contribution 7. *Use of interpretable machine learning methodologies* to obtain biological insights in a human-readable format to guide researchers on their structure prediction endeavors. As current highly accurate prediction methods are difficult to interpret, the capabilities of interpretable methodologies in secondary structure can aid experts in exploring the capabilities and differences from these prediction methods. Furthermore this contribution helps address RQ4, as secondary structure knowledge can help improve tertiary and quaternary structure by refining possible locations where current methods exhibit uncertainty.

Contribution 8. *Providing all source code to replicate our findings*, as well as redevelop tools, e.g. measures calculation method, that have better performance than currently available tools. This effort will facilitate protein secondary structure research for experts. The field of protein secondary structure can be reinvigorated as an additional valuable asset to

support protein research at higher hierarchical structure levels, e.g. tertiary and quaternary, to produce significant improvements in personalized medicine.

3.3. Chapter Summary

This chapter presented an overview of the contributions achieved in the development of this dissertation. We started by identifying a simplified methodology to investigate protein structural changes due to single amino acid mutations. Based on this, we introduced and briefly described our contributions, namely: the creation of mutational dataset, the benchmarking of state-of-the-art prediction methods on the mutational dataset, elucidating mutational deficiencies in the prediction methods, and addressing the deficiencies through a refinement strategy, which materialized in a novel model for protein secondary structure prediction.

Chapter 4

Single amino acid mutations: Backbone structure positional effects

Contents

Correspondences in This Chapter

Addressed Research Question(s):

Q1— How can we collect experimental data for single amino acid mutations that cause both local and global structural changes in a protein?

Q2— How can we evaluate the performance of methods predicting backbone structural changes in proteins?

Q3— Are any of the selected structure prediction methods sufficiently precise to show the effect of single amino acid mutations on protein backbone structure?

Human diversity often manifests through single nucleotide polymorphisms (SNPs). Among these, SNPs that alter amino acids can modify a protein's three-dimensional (3D) structure. This impacts the protein function and can potentially elicit diseases or affect drug interactions. Thus, understanding protein single point mutations is crucial for precision medicine, as it helps tailor treatments based on individual genetic variations. As atomic locations can be susceptible to any number of changes that might or might not affect function, we focus on the secondary structure to provide concrete results on possible protein structural deformation that may occur from single amino acid mutations.

We assess state-of-the-art structure prediction methods regarding backbone deformations caused by single amino acid mutations. We categorize these deformations as **local**, **distant**, or **global** based on the proximity of structural changes to the mutation site. Our analysis utilizes a diverse dataset from the Protein Data Bank, comprising over 500 protein clusters with experimentally determined structures and documented mutations.

Our findings indicate that single amino acid mutations can significantly affect the accuracy of structure prediction methods. These mutations often lead to predicted structural changes even when the actual secondary structures remain unchanged, suggesting that current methods overestimate the impact of single amino acid mutations. This issue is particularly evident in advanced prediction algorithms, which struggle to accurately model proteins with stable mutations. We also found that the addition of low-performing prediction methods during structural analysis can positively impact the results on some proteins, particularly those with low homology. Furthermore, proteins that form complexes or bind ligands—such as membrane and transport proteins—are inaccurately predicted due to the absence of extra-molecular interaction data in the models, highlighting how single amino acid mutations can complicate accurate structure prediction.

4.1. Introduction

We investigate changes caused by single amino acid mutations at three distinct scales:

- *Local* changes, which occur in the immediate vicinity of the mutated residue.
- *Distant* changes, which arise beyond the immediate neighborhood of the mutation site, but remain structurally linked to that specific region.
- *Global* changes, which can manifest anywhere in the overall protein structure, regardless of the mutation's location.

We examine these distinct scales of backbone modification arising from a single amino acid mutation to deepen our understanding of its impact on protein structure. The different vicinity scales allow us to learn how single amino acid mutations affect the backbone structure within the experimental data, and highlight how prediction methods differ in determining the backbone structure after a mutation.

To the best of our knowledge, current structural prediction methods have not been evaluated on their ability to detect protein backbone structural changes induced by single amino acid mutations at the local, distant, and global levels. To assess these structural predictions, we use experimental data to determine how a mutation affects a protein's secondary structure. In this article, we examine the applicability and performance of nine state-of-the-art, methodologically diverse protein structure predictors and evaluate their capacity to distinguish local, distant, and global changes caused by single amino acid mutations, using Q8. This procedure is shown in Fig. 4.1.

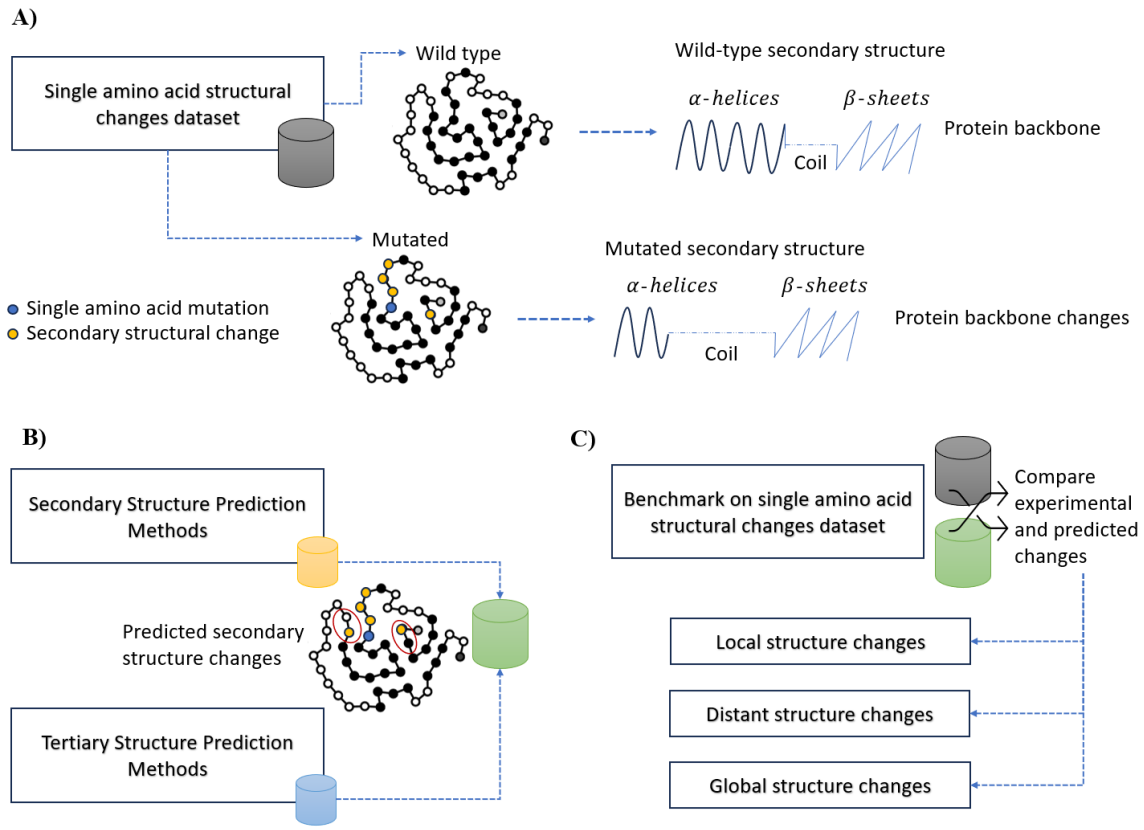


Figure 4.1.|: **Protein secondary structure assessment.** We assess structural changes occurring from single amino acid mutation for both secondary and tertiary structure prediction methods utilizing eight-state secondary structure. Benchmarking procedure is as follows. A) Collect experimental data containing mutations. An example of a secondary structural change during mutation is given, which shortens the α-helix. B) Collect predictions from both secondary and tertiary structure prediction methods on sequences from the previously collected experimental data. C) Comparison of experimental and prediction data to evaluate competency of structural prediction methods on backbone changes due to single amino acid mutations.

In addition, we extend our analysis to both secondary and tertiary structure prediction methods. Our focus on the protein backbone and its secondary structure elements is motivated by the strong correlation between backbone rigidity and protein thermostability [Gonzalez et al., 2022, Spassov et al., 2007], as well as by their capacity to characterize

the protein's tertiary conformation without introducing atomic-level noise.

4.2. Materials and Methods

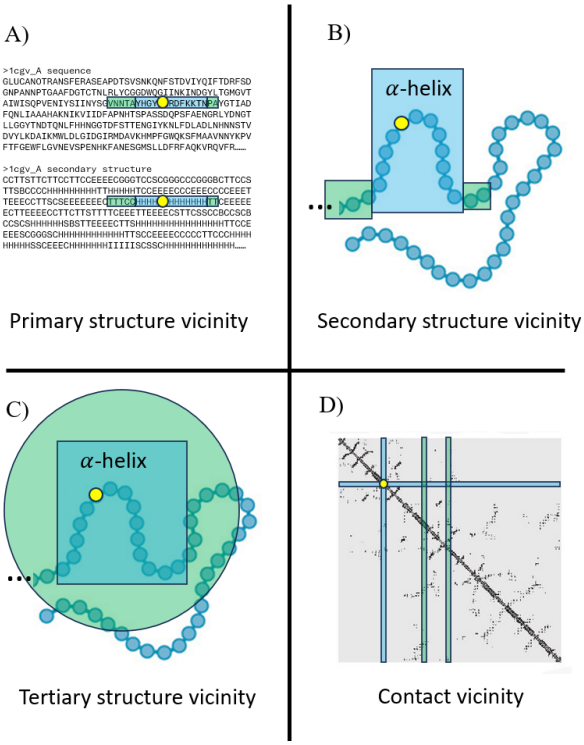


Figure 4.2.|: **Vicinity Measurements.** Examples of the different amino acid vicinities utilized in this study. The vicinity is shown for a single amino acid mutation location shown with a yellow circle. The single amino acid mutation is associated with an α-helix, shown in blue. A) Primary structure (1D) vicinity corresponds to a certain number of amino acids on each side of the target amino acid. Here, the threshold we consider is 9 amino acids. The 1D vicinity encompasses both blue and green boxes. B) Similarly, Secondary structure (2D) vicinity corresponds to a certain number threshold of amino acids on each side, but conditioned on the amino acids corresponding to the same secondary structure element as the immediate surrounding to the target amino acid. Here, the 2D vicinity only encompasses the blue box. C) Tertiary structure (3D) vicinity corresponds to the amino acids within a certain distance of the target amino acid through C_α atoms. Here, the 3D vicinity is comprised of the green circle. D) Similarly, contact vicinity corresponds to amino acids within a certain distance, but utilizing C_β atoms. As contact maps (shown here) are created using C_β atoms, the contact vicinity is comprised of the blue and green lines.

4.2.1. Terminology

Let $\mathbf{A} = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$ be a finite set containing the alphabet of standard amino acids as defined by residue or side chain.

A **protein sequence** over $\mathbf{A}$ is a finite sequence of amino acids $\mathbf{S} = [a_1, \ldots, a_n]$ of length n, where $a_i \in \mathbf{A}$, for all $1 \leq i \leq n$. a_1 is the N-terminus residue and a_n is the C-terminus residue. For each amino acid $a_i \in \mathbf{S}$ at position i in $\mathbf{S}$, there exists two 3D real-valued coordinates $p_i^{c\alpha} = (p_{i_x}^{c\alpha}, p_{i_y}^{c\alpha}, p_{i_z}^{c\alpha})$ and $p_i^{c\beta} = (p_{i_x}^{c\beta}, p_{i_y}^{c\beta}, p_{i_z}^{c\beta})$, where $p_i^{c\alpha}$ and $p_i^{c\beta}$ are associated to C_α-atoms and C_β-atoms respectively of the amino acid in Angstrom (Å) units. The list of coordinates for all amino acids in $\mathbf{S}$ is called the **3D structure** of $\mathbf{S}$.

For $1 \leq i \leq j \leq n$, $\mathbf{S}_{ij} = [a_i, a_{i+1}, \ldots, a_j]$ is called a **protein substring** of length $L = j - i + 1$ of $\mathbf{S}$ from the amino acid at position i to j. The lists of the corresponding C_α-coordinates and C_β-coordinates associated with the amino acids in sequence $\mathbf{S}_{ij}$ are called $P_{ij}^{c\alpha}$ and $P_{ij}^{c\beta}$, respectively.

For the protein sequence $\mathbf{S}$ each amino acid a_i is assigned a secondary structure r_i resulting in $\mathbf{R} = [r_1, r_2, \ldots, r_n]$. Similarly, for a substring $\mathbf{S}_{ij} = [a_i, a_{i+1}, \ldots, a_j]$ the secondary structure is then given by $\mathbf{R}_{ij} = [r_i, r_{i+1}, \ldots, r_j]$. As we investigate Q8 using DSSP, each SSE in $\mathbf{R}_{ij}$ must be an assignment $r_l \in \Upsilon_8$, where $\Upsilon_8 = \{\mathcal{C}, \mathcal{H}, \mathcal{E}, \mathcal{G}, \mathcal{I}, \mathcal{T}, \mathcal{S}, \mathcal{B}\}$ is the set of possible SSE classes in DSSP shown in Table A.1.

If the amino acid $a_m \in \mathbf{S}_{ij}, i \leq m \leq j$ is mutated to $\hat{a}_m$, this results in the modified sequence $\hat{S}_{ij}$ and the corresponding C_α-structure $\hat{P}_{ij}^{c\alpha}$ and C_β-structure $\hat{P}_{ij}^{c\beta}$.

For any amino acid $a_k \in \mathbf{S}$, we define its **primary structure vicinity** (1D vicinity) N_k^{1d} as the subsequence of amino acids surrounding a_k. Formally, $N_k^{1d} = \mathbf{S}_{jl}$ where $1 \leq j = k - \epsilon_{1d} \leq l = k + \epsilon_{1d} \leq n$ and $0 \leq \epsilon_{1d} \leq 20$. This range was chosen to reflect the average length of secondary structure elements, which most commonly span from 10 to 40 amino acids [Sitbon and Pietrokovski, 2007]. See Fig. 4.2.A for a visual representation of the 1D vicinity.

The **secondary structure vicinity** (2D vicinity) N_k^{2d} of an amino acid $a_k \in \mathbf{S}$ is defined as the concatenation of three subsequences: $N_k^{2d} = 2D_k^{nmax} 2D_k^{min} 2D_k^{cmax}$, where $2D_k^{min}$, the minimal 2D vicinity, $2D_k^{nmax}$, the N-terminus extension and $2D_k^{cmax}$, the C-terminus extension are defined as follows:

- The **minimal 2D vicinity** is defined as $2D_k^{min} = \mathbf{S}_{jl}$ where $1 \leq j = k - \epsilon_{2d} \leq l = k + \epsilon_{2d} \leq n$ with $0 \leq \epsilon_{2d} \leq 10$. This minimal vicinity includes a fixed number of amino acids on both sides of a_k. We extend this minimal vicinity to capture the largest possible contiguous secondary structure blocks at each terminus.
- **N-terminus extension** $2D_k^{nmax}$: Consider secondary structure r_j, assigned to the leftmost amino acid a_j in $\mathbf{S}_{jl}$. Then $R_{i(j-1)}^{nmax} = [r_i, r_{i+1}, \ldots, r_{j-1}]$ is called an $nmax$-block if

  1. $1 \leq i = k - \gamma_{2d} < j$,

2. $r_i = \cdots = r_{j-1} = r_j$, and
3. $r_{i-1} \neq r_j$.

Then, $2D_k^{nmax} = [a_i, \ldots, a_{j-1}]$ is the largest contiguous subsequence on the N-terminus side of $2D_k^{min}$ that has the same secondary structure as a_j.

- **C-terminus extension** $2D_l^{cmax}$: Similarly, consider secondary structure r_l, assigned to the rightmost amino acid a_l in $\mathbf{S}_{jl}$. Then $R_{(l+1)m}^{cmax} = [r_{l+1}, \ldots, r_m]$ is called $cmax$-block if

1. $l < m = k + \gamma_{2d} \leq n$,
2. $r_{l+1} = \cdots = r_m = r_l$, and
3. $r_{m+1} \neq r_l$.

Then, $2D_k^{cmax} = [a_{l+1}, \ldots, a_m]$ is the largest contiguous subsequence on the C-terminus side of $2D_k^{min}$ that has the same secondary structure as a_l.

We set $\gamma_{2d} = 30$, defining the maximum possible extension at each terminus.

Therefore, every amino acid in an extension must maintain the same secondary structure as their respective minimal vicinity boundary amino acids a_j or a_l; if not, the extension ceases where the structure changes. The resulting secondary structure vicinity is $N_k^{2d} = [a_{k-\gamma_{2d}}, \ldots, a_{k-\epsilon_{2d}}, \ldots, a_k, \ldots, a_{k+\epsilon_{2d}}, \ldots, a_{k+\gamma_{2d}}] = \mathbf{S}_{im}$. The length of N_k^{2d} is constrained by $\gamma_{2d} = 30$ to achieve vicinity lengths comparable to the other vicinity types. However, the maximum vicinity length for N_k^{2d} is rarely reached due to the typically small span of secondary structure elements. We also enforce a lower bound ϵ_{2d} to prevent the vicinity from becoming too small. See Fig. 4.2.B for a visual representation of the 2D vicinity.

For any amino acid $a_k \in \mathbf{S}$, its **tertiary structure vicinity** (3D vicinity) N_k^{3d} is a collection of subsequences of $\mathbf{S}$, defined by the set of positions of the amino acids in its 3d neighborhood $\{j \mid d_\alpha(k,j) \leq \epsilon_{3d}\}$, with $\epsilon_{3d} = 13$Å and d_α is a distance measure given by Eq 4.1. A visual representation can be seen in Fig. 4.2.C.

$$d_\alpha(k,j) = \sqrt{(p_{k_x}^{c\alpha} - p_{j_x}^{c\alpha})^2 + (p_{k_y}^{c\alpha} - p_{j_y}^{c\alpha})^2 + (p_{k_z}^{c\alpha} - p_{j_z}^{c\alpha})^2} \tag{4.1}$$

Analogously, we define **contact vicinity** $N_k^{contact}$ for any amino acid $a_k \in \mathbf{S}$ as the collection of subsequences of $\mathbf{S}$, defined by the set of positions of the amino acids in its contact vicinity $\{j \mid d_\beta(k,j) \leq \epsilon_{contact}\}$, where $\epsilon_{contact} = 8$Å and d_β is a distance measure given by Eq 4.2. A visual representation can be seen in Fig. 4.2.D.

$$d_\beta(k,j) = \sqrt{(p_{k_x}^{c\beta} - p_{j_x}^{c\beta})^2 + (p_{k_y}^{c\beta} - p_{j_y}^{c\beta})^2 + (p_{k_z}^{c\beta} - p_{j_z}^{c\beta})^2} \tag{4.2}$$

The maximum vicinity length thresholds for 3D and contact vicinities are set to match previous definitions. The threshold for N_k^{3d}, ϵ_{3d} is based on a previously established threshold by Alphafold2 [McBride et al., 2023], while $\epsilon_{contact}$ for $N_k^{contact}$ aligns with the accepted definition of contact prediction [Schaarschmidt et al., 2018].

A single amino acid mutation can trigger a structural change within a protein. We categorize the structural changes into three categories for each of the aforementioned vicinities. A **local** structural change occurs if a mutation at position l alters the secondary structure within its respective vicinity N_k^{1d}, N_k^{2d}, N_k^{3d}, or $N_k^{contact}$. A **distant** structural change occurs if a mutation at l modifies the structure outside its respective vicinity (i.e. N_k^{1d}, N_k^{2d}, N_k^{3d}, or $N_k^{contact}$) but not within the vicinity. Finally, a **global** structural change occurs if the altered secondary structure is neither confined within nor solely outside its respective vicinity.

When a mutation induces a structural change, we refer to it as *disruptive*. Consequently, all disruptive mutations cause a global change, which entails any possible structural mutation. Furthermore, a disruptive mutation can cause a local or distant change depending on the structural changes that occur and their distance to the single amino acid mutation. In contrast, if no secondary structural change occurs, the mutation is considered *stable*. The structural changes described above are illustrated in Fig 4.3.

4.2.2. Data acquisition and processing

Protein sequences (PDB_SEQRES.TXT) and their experimentally derived 3D structures (PROTEIN.CIF) were obtained from the Protein Data Bank (PDB) as of April 2023. We excluded non-protein sequences, duplicates and retained only proteins composed of the 20 standard amino acids. Consequently, any sequences containing ambiguous amino acids were also removed.

We clustered the protein sequences using CD-HIT [Fu et al., 2012] with a 99% sequence similarity threshold, producing groups of mutated proteins. Within each cluster, we performed multiple sequence alignments using Clustal Omega [Sievers and Higgins, 2014] to identify amino acid substitutions and discard incomplete sequences. After alignment and filtering, each cluster retained only full-length sequences of uniform length.

However, while retrieving the structural data files for these sequences, we sometimes encountered structures with missing amino acids, resulting in gaps where the atoms' locations are inconclusive. Such gaps can obscure the true effects of single amino acid mutations, as the 3D structures may not fully represent the corresponding protein sequence associated to the structure. To ensure that all relevant atomic positions are accounted for, we excluded any proteins whose structure files contained these gaps. This approach guarantees that the mutation effects that we analyze are accurately represented.

After this preprocessing step, we applied DSSP to assign secondary structures to each protein sequence using their corresponding experimental structures. DSSP is selected to provide a fair assessment of the prediction methods, as Q8 prediction methods are trained

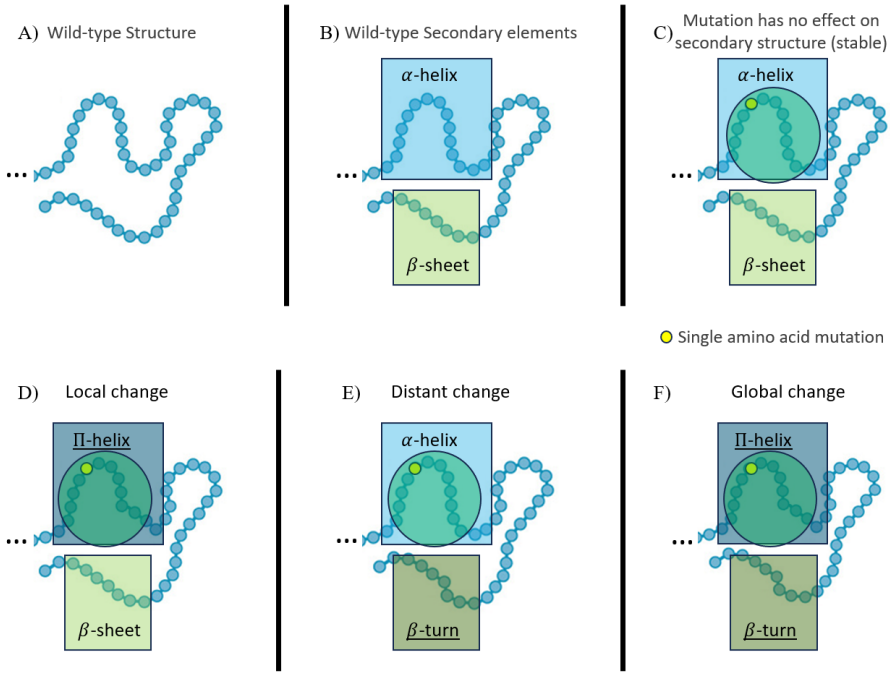


Figure 4.3.|: **Types of Backbone Changes.** Measuring backbone changes in proteins requires pinpointing specific locations within their secondary structures, relative to the mutation site. This allows us to observe how a mutation impacts the protein's backbone. Yellow circles indicate the amino acid mutation location. Blue regions show the secondary structure in the mutated region. Green regions contain the amino acids that are part of the vicinity. A) Original protein backbone structure. B) Secondary elements in the protein backbone. C) No structural change due to mutation. D) Local structural change due to mutation. E) Distant structural change outside the local structural vicinity of the mutation. F) Global structural change occurring anywhere in the protein backbone.

specifically on DSSP classifications. As resulting protein sequences had uniform length within a cluster and differ through single amino acid mutations, the proteins could be considered aligned. Wild-type and mutated sequences were identified through mutation extraction that is detailed in Supplementary section A.4. The identification procedure follows the same logic as Weblogos [Crooks et al., 2004], which display the most common amino acid as the largest symbol in a figure.

We ran each method on our dataset and normalized the results to ensure consistent Q8 predictions, as different methods may use distinct symbols for identical secondary structure classes. After completing these steps, we obtained the final preprocessed dataset used for evaluating the methods. A summary of this process is illustrated in Fig. 4.4.

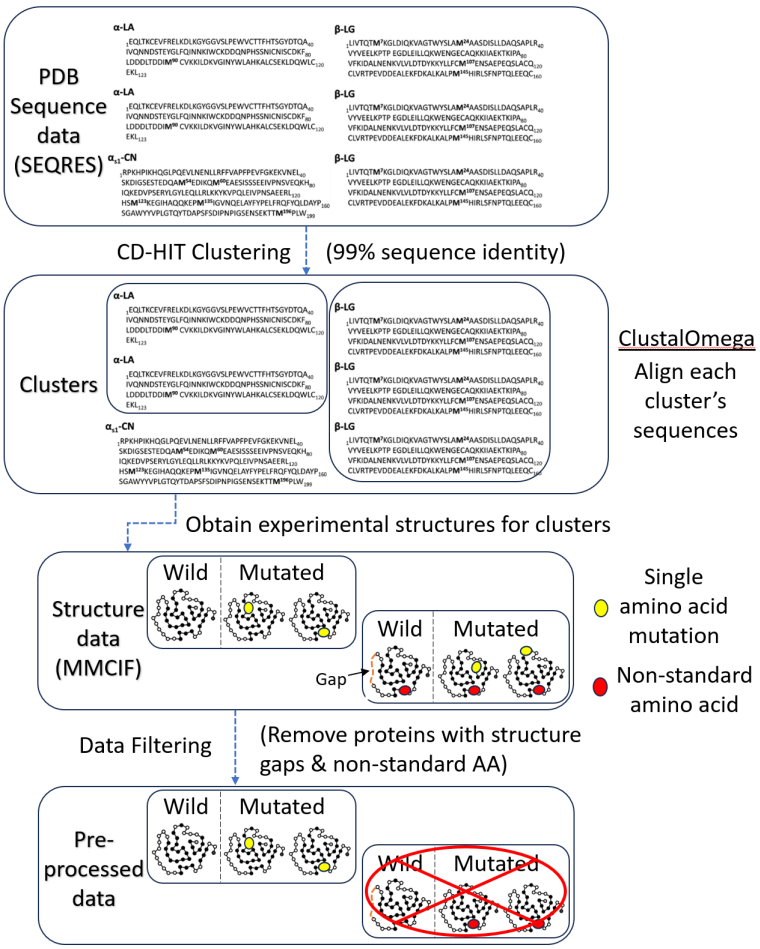


Figure 4.4.|: **Data processing.** Top to bottom: Begin with 'SEQRES' data containing PDB molecular sequences (keeping proteins only). Followed by sequence clustering using 99% sequence identity through CD-HIT. Then, we align each cluster's sequences using Clustal Omega to obtain clusters solely containing equal-length proteins. Afterwards, collect the experimental structural data relating to the protein sequences in the clusters. Finally, filter the data to ensure that all amino acids appear in their associated experimental structure files. This means that no amino acid atoms are missing from the structure, and the sequence and structure residues match one-to-one. We also remove proteins with non-standard amino acids to accommodate the prediction methods.

4.2.3. Dataset statistics

Our preprocessed dataset comprises 579 clusters encompassing $1,414$ proteins. It serves as a simple mutational dataset, where at most 1% of the amino acids are mutated and an average length of 238 amino acids. Further filtering of the dataset ensures a dataset containing single amino acid mutations only. The protein sequences range from 100 to 858 amino acids in length. Any proteins longer than $1,024$ amino acids were excluded, as several prediction methods cannot handle such length.

Each cluster may include multiple single amino acid mutations, depending on the number of proteins it contains. Approximately 60% of the clusters have just one mutation, while the remainder is evenly split—between those with two mutations and those with three or more. A wide range of sequence lengths is represented across all clusters, as illustrated in Fig. 4.5.

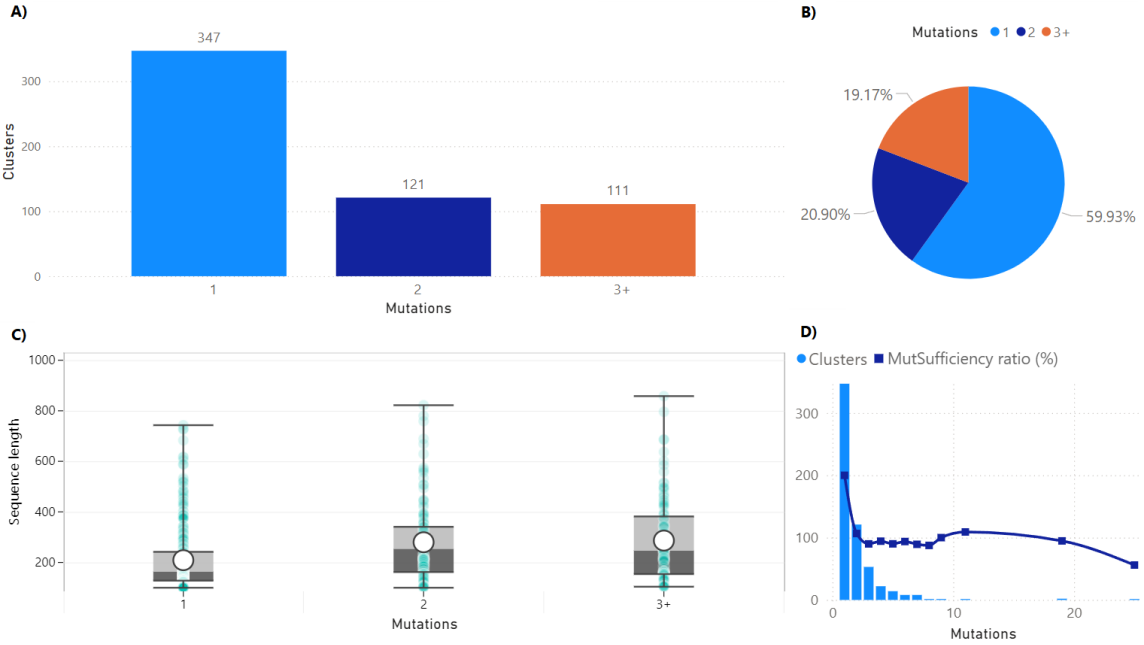


Figure 4.5.|: **Protein mutation statistics.** A) Number of clusters containing 1, 2, or 3+ protein mutations. B) Percentage of clusters in the dataset for certain mutations. C) Protein sequence lengths in clusters with 1, 2, or 3+ protein mutations. We can see an even spread of sequence lengths among all clusters regardless of mutations D) Number of clusters with a certain number of mutations and mutation sufficiency ratio as a percentage.

Because the clusters were formed at 99% sequence similarity, individual proteins within a cluster may carry more than one amino acid mutation. To identify clusters containing only a single unique mutation per protein, we applied the *Mutation Sufficiency* (`MutSufficiency`) measure defined in Eq. 4.3. This measure uses the ratio of the total number of mutations to the total number of proteins in a cluster (recall that each protein has at least one mutation).

$$\texttt{MutSufficiency}(c) = \begin{cases} 1, \frac{prot_c}{mut_c} \geq 100\% \\ 0, \frac{prot_c}{mut_c} < 100\% \end{cases} \tag{4.3}$$

where mut_c is the total number of mutations in cluster c and $prot_c$ is the total number of proteins in c. After applying the Mutation Sufficiency measure we obtain a set of clusters $\{c \mid \texttt{MutSufficiency}(c) \neq 0\}$. This set retained 542 clusters and $1,291$ proteins, with an average length of 226 amino acids. (See Fig. 4.6).

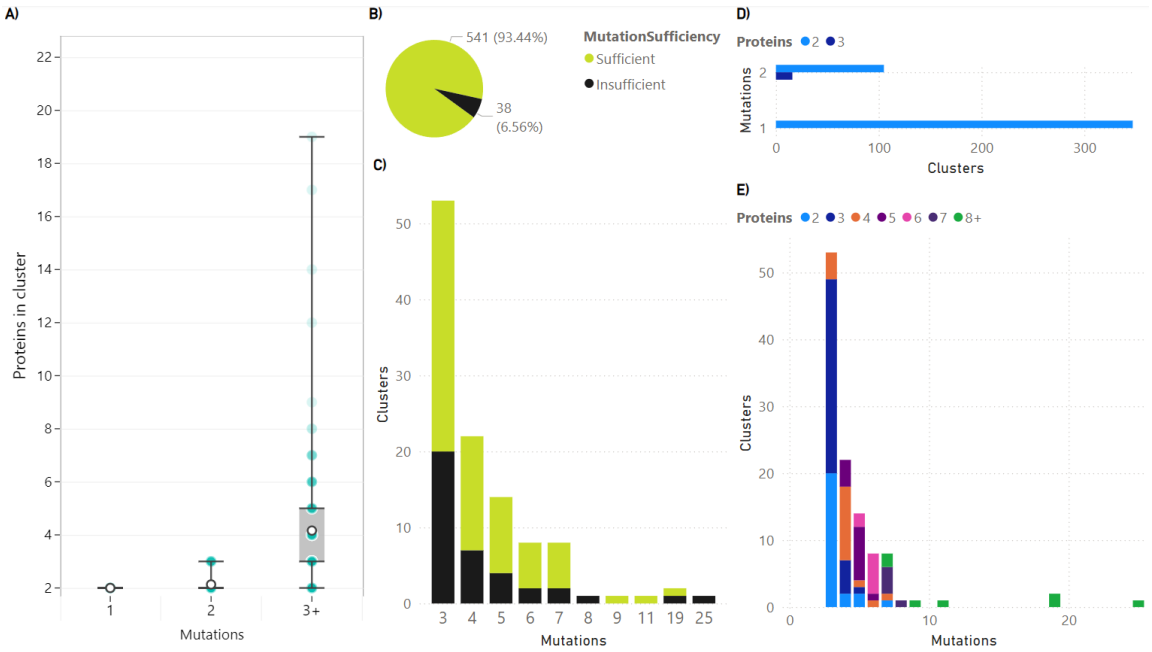


Figure 4.6.|: **Protein mutation sufficiency statistics.** All mutation insufficient clusters contain 3 or more mutations. Clusters were separated according to their number of proteins from 2 to 8 or more (8+). A) Whisker plot showing the mutations that occur for certain number of proteins in a cluster. White circle represent the mean value. The box indicates the 25th and 75th percentiles. The transparency of the gray circles indicate the number of clusters (more transparent, more clusters with that number of proteins). B) Percentage and values of mutation sufficient and insufficient clusters. C) Number of clusters containing a specific number of mutations. D) Clusters with 1 or 2 mutations (All of these clusters are mutation sufficient). E) Detailed graph of proteins and mutations for clusters.

Most of the removed clusters from the Mutation Sufficiency filtering were those with three or more mutations per cluster, as well as those with longer protein sequences and higher mutation rates. These changes are detailed in Supplementary section A.5. This filtration primarily removed mutational outliers, where long protein sequences carried an exceptionally high proportion of mutations. As we delve into single amino acid mutations, the Mutation Sufficiency filtering was applied to remove any clusters with an insufficient number of mutations arising from possible duplicate mutations.

We were also interested in any evolutionary structural information found in our protein dataset, as well as functional annotations that this evolutionary information can provide for the proteins. For this task, we utilized structural homology datasets, which contain a collections of proteins that are structurally similar due to shared ancestry. These datasets identify conserved regions, and predict the function of unknown protein sequences based on known ones from experimental data.

SCOP [Andreeva et al., 2020] (Structural Classification of Proteins) and CATH [Sillitoe et al., 2021] (Class, Architecture, Topology, Homologous superfamily)

are databases that categorize proteins based on their structural and evolutionary relationships. Applying these classifications to our dataset reveals that not all proteins have matching entries: SCOP lacks data for about 25% of our clustered proteins, and CATH is missing data for approximately 15%.

Among the proteins with SCOP data, more than 98% are globular, around 1.5% are membrane proteins, and about 0.5% are fibrous proteins. In the CATH classifications, roughly 43% feature a mixture of alpha and beta secondary structures, about 36% have predominantly beta (sheet) structures, around 20% primarily contain alpha (helical) structures, and the remaining 1% exhibit few secondary structures.

Properties shared by more than 50% of our proteins primarily include top-level SCOP and CATH classifications, as well as common protein functions such as 'binding'. The removal of these prevalent properties is done to minimize bias from frequently appearing characteristics in our subsequent analyses.

Each mutation vicinity differs in length due to the varying number of amino acids it encompasses. The contact vicinity is the longest, followed by the 1D and 3D vicinities. The 2D vicinity is the shortest, averaging at least 10 fewer amino acids than any other vicinity type.

These protein and mutation statistics are illustrated in Fig. 4.7.

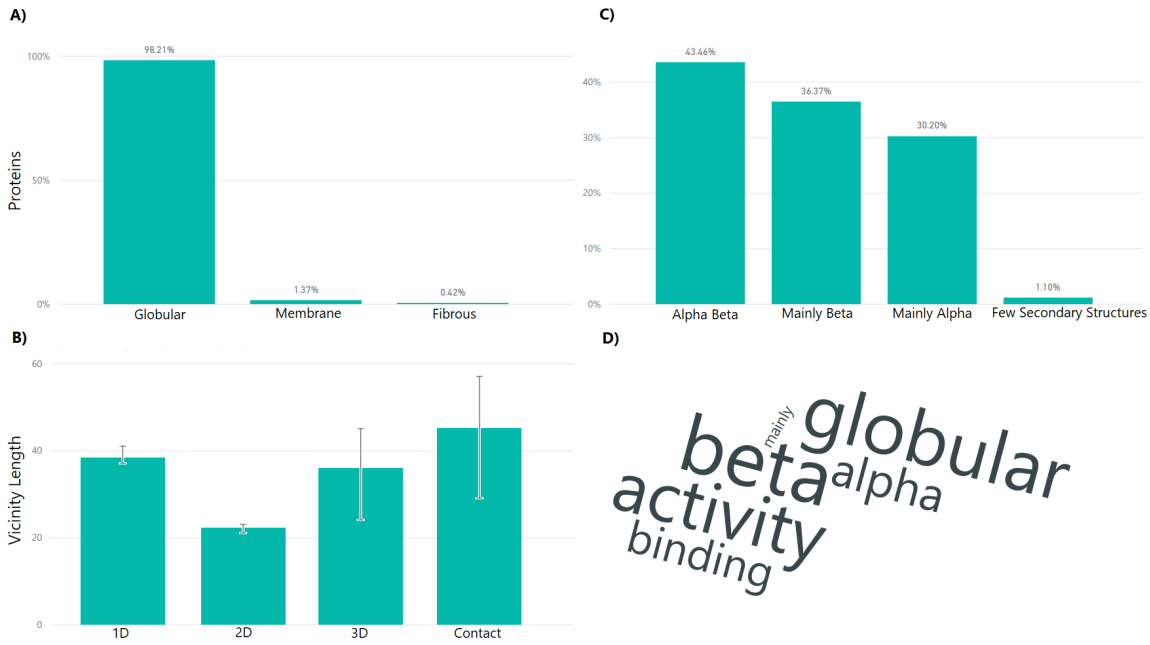


Figure 4.7.|: **Protein statistics.** Percentage proteins containing SCOP (A) and CATH (B) top level classifications in our data. C) Average vicinity length according to the mutation vicinity type. D) Most common protein properties in our dataset.

4.2.4. Dataset limitations

Some of the original training data used by these methods are included in our dataset, introducing a positive performance bias that cannot be fully mitigated due to the limited availability of experimentally obtained single-amino acid mutation structures. To gauge the extent of the data overlap for each method, even though most methods do not provide the respective exact training dataset, we assembled a dataset closely resembling their training data by following the published methodologies.

Raptor-X Property was originally trained using a dataset [Wang et al., 2016b], which included protein chains resolved at better than 2.5 Å, with less than 30% protein sequence identity, lengths between 50 and 700 residues, and no chain discontinuities. That dataset contained approximately 5,600 proteins.

To approximate these conditions, we selected the May 2012 CullPDB dataset ('cullpdb_pc30_res2.5_R1.0_d120428_chains9175'). We filtered it by length and excluded any proteins published after 2010, using publication dates obtained from the Protein Data Bank Japan (`https://pdbj.org/`). This process yielded $5,972$ proteins, among which 40 clusters and 40 individual proteins overlap with our dataset.

SSPro8 was originally trained on a CullPDB dataset from the PISCES [Wang and Dunbrack, 2003] server, utilized for culling sets of protein sequences from the Protein Data Bank (PDB) based on sequence identity and structural quality criteria. The resulting dataset (Cullpdb_pc30_res2.5_ R1.0_d100716'), contained roughly 8,000 high-resolution protein structures. Similar to Raptor-X Property, we used the 'cullpdb_pc30_res2.5_R1.0_d120428_chains9175' dataset as the closest available resource. After discarding protein sequences shorter than 50 residues or longer than 1,500 residues and removing proteins published after 2011, we obtained a set of $8,039$ proteins. Of these, 45 proteins—distributed across 45 clusters—overlap with our dataset.

SPOT-1D utilizes a dataset of around 10,000 proteins. This dataset was based on the data used for SPIDER3-Single, a method that precluded SPOT-1D, which its methodology was upgraded to become SPOT-1D. This dataset has 38 clusters and 38 proteins in common with our dataset.

SPOT-1D-Single and SPOT-1D-LM utilize the same dataset, which is publicly and readily available (`https://sparks-lab.org/server/spot-1d-lm/`). Their dataset has 39119 proteins, which has 114 clusters and 126 proteins in common with our dataset.

Alphafold2 and ColabFold use a training dataset derived from PDB with proteins published before 30 Apr. 2018. This dataset contains 139,417 proteins with 1174 proteins in common with our dataset covering 503 clusters.

Similarly, ESMFold uses a training dataset derived from PDB with proteins published before 30 Nov. 2022. This dataset contains 198,618 proteins which cover $1,371$ proteins and 570 clusters in our dataset.

RGN2 was trained on the 'astral-rapid-access-1.75.raf' dataset, which includes $87,061$ proteins. Within our dataset, this training set overlaps with 269 clusters and 609 proteins.

Additionally, RGN2 predictions omit the last two residues of each protein sequence. To align with the lengths used by other methods, we appended two coil secondary structures to the end of each prediction. Comparing results with and without this adjustment (Fig. 4.8) shows a slight overall performance increase, likely because most protein termini consist of coils.

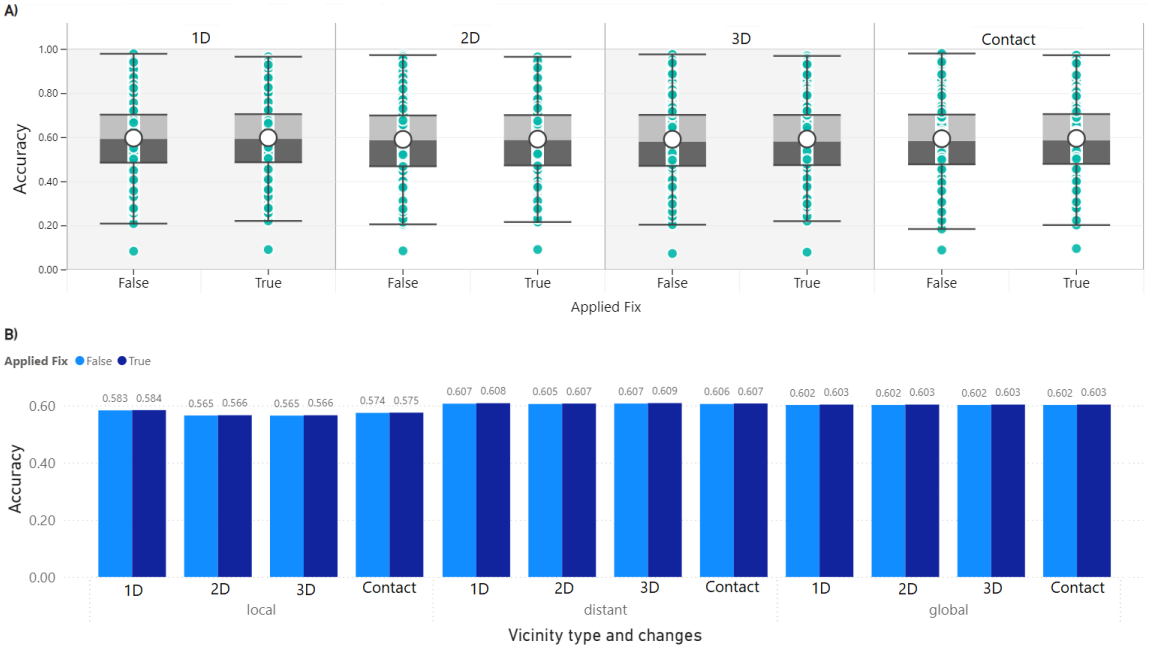


Figure 4.8.|: **Adjustment to the RGN2 secondary structure prediction.** Difference in performance with (True) and without (False) adjusting for the length. A) Box plot showing performance of RGN2 for each mutation vicinity with boxes spanning from 25 to 75 percentile, as well as whiskers of 1.5 IQR. B) Performance of RGN2 for each mutation vicinity and type of backbone change. These graphs show that the results remain unchanged after the adjustment was performed, thus the RGN2 results remaining valid.

Furthermore, all protein structure prediction methods draw on protein sequence homology from databases that include the majority of known protein sequences at the time of their release. Since these methods rely on databases published between 2013 and 2022, it is reasonable to assume that any sequence published prior to a method's release year may have been part of its training data.

Finally, our dataset focuses on single amino acid mutations where the experimental structure is fully known. Therefore the removal of structural gaps, along our other filtering techniques, might remove proteins with intrinsically disordered regions, insertions and deletions. Other genetic mutations that have downstream effects on protein, e.g. nonsense mutations or frameshift mutations, would also be removed from the final protein dataset.

4.2.5. Protein descriptors

We obtained protein descriptors from multiple sources, including gene ontology (GO) annotations, structural file descriptions, and structural homology classifications. GO annotations were sourced from UniProtKB [The UniProt Consortium, 2023]. Structural descriptions were extracted from each protein's mmCIF files in the PDB. Structural homology classifications were retrieved from the CATH [Sillitoe et al., 2021] and SCOP [Andreeva et al., 2020] databases. These descriptors were aggregated for each protein, as no single source included all the necessary information for our dataset.

In the Results and Discussion section, we mention the descriptors as properties related to a corresponding protein. These properties are obtained by identifying the most commonly found descriptors for the protein within our previously mentioned aggregated descriptors. These properties are filtered to eliminate recurrent descriptors found in all proteins to prevent non-specific descriptors for each protein. The process is done through word clouds, which are shown in Supplementary section A.7.

4.2.6. Secondary structure measures

To assess the performance of secondary structure prediction methods on Q8, three commonly used measures are employed: ACCURACY (Q^{Acc}), SEGMENT OVERLAP (SOV), and SOV_REFINE. As SOV has been improved since its inception, the most common version, SOV99 [Zemla et al., 1999], is the version that we refer to as SEGMENT OVERLAP. Its most recent modification is referred to as the refined version SOV_REFINE [Liu and Wang, 2018]. These measures are computed using our "Secondary Structure Measures Calculator" software (`https://github.com/ivanpmartell/SSMetrics`), which implements previously published secondary structure measures, such as ACCURACY, SOV_REFINE, SOV99, but also includes older SOV versions like SOV94 [Rost et al., 1994b].

The measures require two secondary structures of length n as input:

Reference structure: $\mathbf{R}^{ref} = [r_1, r_2, \ldots, r_n]$

Predicted structure: $\mathbf{R}^{pred} = [\widetilde{r}_1, \widetilde{r}_2, \ldots, \widetilde{r}_n]$

For each SSE class r, a *reference segment* (or *reference r-block*) in $\mathbf{R}^{ref}$ is defined as a contiguous substructure $\mathbf{B}^{ref}_{jk}(r) = [r_j, r_{j+1}, \ldots, r_k]$ of $\mathbf{R}^{ref}$ that satisfies the following conditions:

Uniform Structure: All residues in the substructure are of SSE class r (i.e., $r_j = r_{j+1} = \ldots = r_k = r$).

Boundary Conditions:
- $r_{j-1} \neq r$ and $r_{k+1} \neq r$, thus the SSEs immediately before r_j and after r_k must not be SSE class r.

Similarly, for each SSE class r, a *prediction r-block* in $\mathbf{R}^{pred}$ is defined as $\mathbf{B}_{jk}^{pred}(r)$ following the same criteria. The sets of all such blocks are defined as:

Reference Blocks:

$$B^{ref} = \left\{ \mathbf{B}_{jk}^{ref}(r) \mid r \in \Upsilon_{ref} \right\}$$

Prediction Blocks:

$$B^{pred} = \left\{ \mathbf{B}_{lm}^{pred}(r) \mid r \in \Upsilon_{ref} \right\}$$

Here, Υ_{ref} represents the set of all secondary structure element (SSE) classes present in the reference structure sequence $\mathbf{R}^{ref}$.

In this study, we evaluate Q8 prediction by calculating the following measures for each DSSP-assigned secondary structure class within $\Upsilon_8 = \{\mathcal{B}, \mathcal{C}, \mathcal{E}, \mathcal{G}, \mathcal{H}, \mathcal{I}, \mathcal{S}, \mathcal{T}\}$, where $\Upsilon_{ref} \subseteq \Upsilon_8$. Further details on DSSP assignments are given in Supplementary section A.1 and section A.2.

All secondary structure measures evaluate SSEs at each position of both reference and predicted structure sequences using the *identity* function,

$$I_i^r(\mathbf{R}^{ref}, \mathbf{R}^{pred}) = \begin{cases} 1, \text{ if } r_i = \widetilde{r}_i = r \\ 0, \text{ otherwise} \end{cases} \tag{4.4}$$

where i is the position of the SSE in both secondary structures $\mathbf{R}^{ref}$ and $\mathbf{R}^{pred}$, and r is the SSE class.

The ACCURACY measure is defined as the ratio of matching SSE pairs between the reference and predicted structures to the total number of SSE pairs. Since the reference and predicted structures have equal lengths, the total number of SSE pairs is equal to the length of the structure sequences, i.e., $\left|R^{\text{ref}}\right| = \left|R^{\text{pred}}\right| = n$. This is more formally defined as,

$$Q_{\Upsilon_8}^{Acc} = \frac{\sum\limits_{r \in \Upsilon_8} \sum\limits_{i=1}^{n} I_i^r(\mathbf{R}^{ref}, \mathbf{R}^{pred})}{n} \tag{4.5}$$

ACCURACY, which measures exact matches between two secondary structures at each position, may not be able to sufficiently capture the structural details of slightly misaligned secondary structure elements (SSEs) that extend across the structure sequence. To address this limitation, a more informative measure called SEGMENT OVERLAP (SOV) was introduced [Rost et al., 1994b] and consequently improved [Zemla et al., 1999].

The SEGMENT OVERLAP measure is a weighted sum over overlapping pairs of segment blocks for each SSE class $r \in \Upsilon_{ref}$, accounting for slight misalignments in SSEs. Formally, for r-blocks $\mathbf{B}_{jk}^{ref}(r)$ and $\mathbf{B}_{lm}^{pred}(r)$, let $BP_{lm}^{jk}(r) = (\mathbf{B}_{jk}^{ref}(r), \mathbf{B}_{lm}^{pred}(r))$ be an **overlapping segment block pair** for $\mathbf{R}^{ref}$ when $j \leq m$ and $l \leq k$. Then, let $BP_{lm}^{jk}(r)_{ref}$ denote the first element of $BP_{lm}^{jk}(r)$ (i.e., $\mathbf{B}_{jk}^{ref}(r)$) for that overlapping segment block pair.

We denote $O(r) = \{BP_{lm}^{jk}(r)|\ 1 \leq l \leq m \leq n \text{ and } 1 \leq j \leq k \leq n\}$ as the **set of all overlapping pairs** of r-blocks between $\mathbf{R}^{ref}$ and $\mathbf{R}^{pred}$ (i.e., the set of all tuples $(\mathbf{B}_{jk}^{ref}(r), \mathbf{B}_{lm}^{pred}(r))$ for all $r \in \Upsilon_{ref}$). If a reference r-block $\mathbf{B}_{jk}^{ref}(r)$ has no overlap with any predicted r-block in B^{pred}, we define $\overline{O}(r)$ as the set of non-overlapping segment blocks in $\mathbf{R}^{ref}$ for SSE class r:

$$\overline{O}(r) = \{\mathbf{B}_{jk}^{ref}(r) \mid \text{no } BP_{lm}^{jk}(r)_{ref} \in O(r)\}.$$

We now specify the functions that take part in the SOV definition.

$\texttt{Norm}(r)$ is the **normalization value** for SSE class r defined as,

$$\texttt{Norm}(r) = \sum_{O(r)} \left|\mathbf{B}_{jk}^{ref}(r)\right| + \sum_{\overline{O}(r)} \left|\mathbf{B}_{jk}^{ref}(r)\right| \tag{4.6}$$

where $\left|\mathbf{B}_{jk}^{ref}(r)\right| = k - j + 1$ is the length of the r-block. It is important to note that any particular reference r-block can appear multiple times across different block pairs in $O(r)$.

LenOv_r is the number of identical SSEs of class r for a pair of segments and defined as,

$$\text{LenOv}_r(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m) = \sum_{i=\max(j,l)}^{\min(k,m)} I_i^r(\mathbf{R}^{ref}, \mathbf{R}^{pred}) \tag{4.7}$$

where $\delta(\mathbf{R}_{jk}^{ref}, \mathbf{R}_{lm}^{pred})$ is the amount of **allowable misalignment** given to a pair of segments and defined as,

$$\delta_r(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m) = \min \begin{cases} \left|BP_{lm}^{jk}(r)\right| - \text{LenOv}_r(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m) \\ \text{LenOv}_r(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m) \\ \left\lfloor \left|\mathbf{B}_{jk}^{ref}(r)\right| / 2 \right\rfloor \\ \left\lfloor \left|\mathbf{B}_{lm}^{pred}(r)\right| / 2 \right\rfloor \end{cases} \tag{4.8}$$

where $\left|BP_{lm}^{jk}(r)\right| = \max\{k, m\} - \min\{j, l\} + 1$ is the combined overlap length of an overlapping pair of segment blocks. An example of the way overlapping segments can obtain the different δ_r values is shown in Fig. 4.9.

To define **Segment Overlap**, it is important to note that the set of all overlapping pairs of r-blocks $O(r)$ contain r-blocks with their respective starting and ending indices (e.g.,

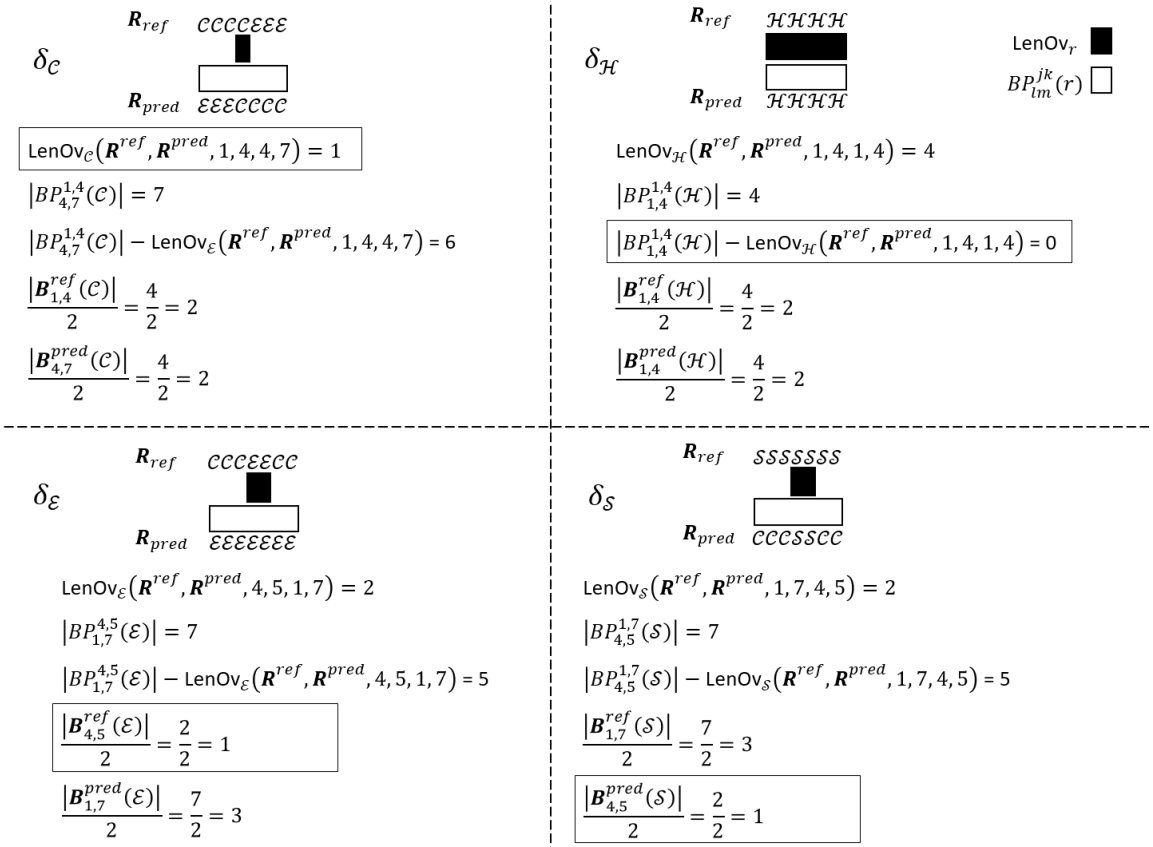


Figure 4.9.|: **Calculating δ_r.** Example shows all four possible minimum values depending on the overlapping segments. The resulting value is the misalignment allowed for the r-block pair.

$\mathbf{B}^{ref}_{jk}(r)$, where j is the starting index of the r-block for $\mathbf{R}^{ref}$, and k is the ending index). Then for SSE class r, we define SEGMENT OVERLAP $SOV(r)$,

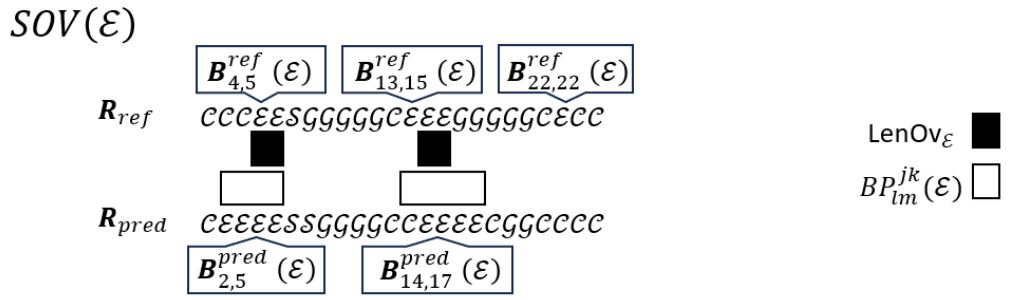
$$SOV(r) = \frac{\sum\limits_{(\mathbf{B}^{ref}_{jk}(r), \mathbf{B}^{pred}_{lm}(r)) \in O(r)} F}{\texttt{Norm}(r)} \tag{4.9}$$

$$F = \frac{\text{LenOv}_r\left(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m\right) + \delta_r\left(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m\right)}{\left|BP^{jk}_{lm}(r)\right|} \left|\mathbf{B}^{ref}_{jk}(r)\right|$$

Lastly the overall SOV score for all SSE classes is defined as,

$$SOV_{|\Upsilon_{ref}|} = \frac{\sum\limits_{r \in \Upsilon_{ref}} (SOV(r) \cdot \texttt{Norm}(r))}{\sum\limits_{r \in \Upsilon_{ref}} \texttt{Norm}(r)} \tag{4.10}$$

An example for calculating SEGMENT OVERLAP can be seen in Fig. 4.10.



$$O(\mathcal{E}) = \left\{\left(\boldsymbol{B}^{ref}_{4,5}(\mathcal{E}), \boldsymbol{B}^{pred}_{2,5}(\mathcal{E})\right), \left(\boldsymbol{B}^{ref}_{13,15}(\mathcal{E}), \boldsymbol{B}^{pred}_{14,17}(\mathcal{E})\right)\right\}$$

$$\text{LenOv}_{\mathcal{E}}\left(\boldsymbol{R}^{ref}, \boldsymbol{R}^{pred}, 4, 5, 2, 5\right) = 2 \qquad \text{LenOv}_{\mathcal{E}}\left(\boldsymbol{R}^{ref}, \boldsymbol{R}^{pred}, 13, 15, 14, 17\right) = 2$$

$$\left|BP^{4,5}_{2,5}(\mathcal{E})\right| = 4 \qquad \left|BP^{13,15}_{14,17}(\mathcal{E})\right| = 5$$

$$\left|\boldsymbol{B}^{ref}_{4,5}(\mathcal{E})\right| = 2 \qquad \left|\boldsymbol{B}^{ref}_{13,15}(\mathcal{E})\right| = 3$$

$$O'(\mathcal{E}) = \{\boldsymbol{B}^{ref}_{22,22}(\mathcal{E})\} \qquad Norm(\mathcal{E}) = \left|\boldsymbol{B}^{ref}_{4,5}(\mathcal{E})\right| + \left|\boldsymbol{B}^{ref}_{13,15}(\mathcal{E})\right| + \left|\boldsymbol{B}^{ref}_{22,22}(\mathcal{E})\right| = 6$$

$$\left|\boldsymbol{B}^{ref}_{22,22}(\mathcal{E})\right| = 1 \qquad SOV(\mathcal{E}) = \frac{1}{6} \times \left(\left(\frac{2+1}{4} \times 2\right) + \left(\frac{2+1}{5} \times 3\right)\right) = 0.55$$

Figure 4.10.|: **Calculating** $SOV(\mathcal{E})$**.** Example following the nomenclature from our definitions in the secondary structure measures section.

The refined version of SOV, termed as SOV_REFINE, changes the allowance function as follows,

$$\delta_r\left(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m\right) = \min \begin{cases} \delta(\mathbf{R}^{ref}) \dfrac{\left|\mathbf{R}^{ref}_{jk}\right|}{|\mathbf{R}^{ref}|} \cdot \dfrac{\text{LenOv}_r\left(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m\right)}{\left|BP^{jk}_{lm}(r)\right|} \\ \left|BP^{jk}_{lm}(r)\right| - \text{LenOv}_r\left(\mathbf{R}^{ref}, \mathbf{R}^{pred}, j, k, l, m\right) \end{cases} \tag{4.11}$$

where $\delta(\mathbf{R}^{ref})$ is the **total misalignment allowance** given for all segment blocks of the reference structure sequence as,

$$\delta(\mathbf{R}^{ref}) = \lambda \cdot \frac{|\Upsilon_{ref}|}{\sum\limits_{\mathbf{B}^{ref}_{jk}(r) \in B^{ref}} \left(\dfrac{\left|\mathbf{B}^{ref}_{jk}(r)\right|}{|\mathbf{R}^{ref}|}\right)^2} \tag{4.12}$$

where $\Upsilon_{ref} \subseteq \Upsilon_8$ and $|\Upsilon_{ref}|$ is the number of SSE classes that appear in the reference

structure sequence, and $\lambda \in \mathbb{R}, 0 \leq \lambda \leq 1$ is an adjustable scale hyper-parameter that is used to limit the range of $\delta(\mathbf{R}^{ref})$.

4.2.7. Measures calculation

Existing tools for calculating the previously defined measures are script-based and lack efficiency, particularly for large datasets or longer structure sequences. To address this, we developed a more efficient measure calculation tool that outperforms currently available options. Secondary structure measures for this project were calculated using our custom software, **SSMeasures**. The results were validated against the Perl script SOV_refine.pl developed by Liu and Wang [Liu and Wang, 2018]. The performance comparison between our SSMeasures and existing methods is shown in Fig. 4.11.

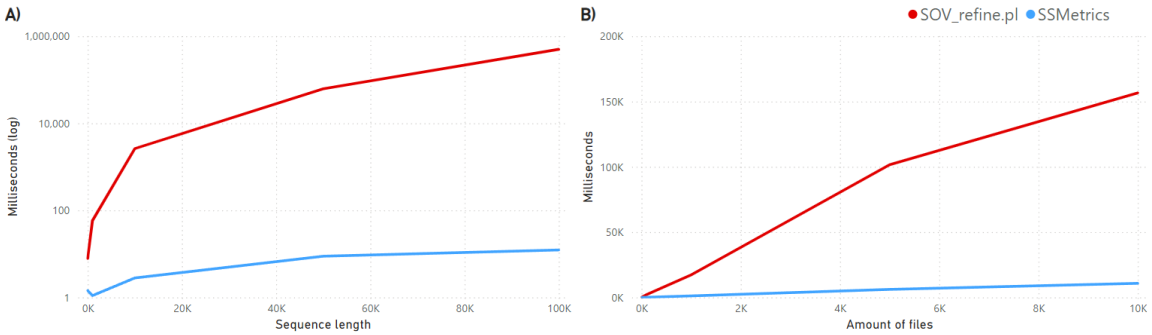


Figure 4.11.|: **Performance comparison between SSMeasures and SOV_refine.pl.** A) Performance measured with the "perf stat" profiling tool across varying structure sequence lengths. B) Performance measured with the "time" tool for runs involving 10, 100, 1,000, 5,000, and 10,000 files of 500 amino acids each.

4.2.8. Mutational measures

Mutational data clusters consist of a wild-type protein sequence along with its associated mutations. The previously discussed measures are specifically designed to compare two secondary structures: a reference structure and a predicted structure. For each protein, the reference structure is derived using DSSP, while the predicted structure is generated by a structure prediction method.

To evaluate the performance of prediction methods on mutational data, it is essential to account for mutational changes. To achieve this, we introduce three mutational measures, described below.

Mutational consistency measures the ACCURACY between mutational changes in the reference and predicted structures. To compute this, we:

1. compare the wild-type and mutated reference structures element-wise for each secondary structure element (SSE). This yields the **reference mutational change** sequence, indicating whether each SSE is preserved ('N') or changed ('C') after mutation.

2. perform the same comparison for the wild-type and mutated predicted structures, producing the **predicted mutational change** sequence.

3. calculate ACCURACY for the reference and predicted mutational change sequences, as described in the Secondary structure measures section, treating the two possible states—'preserved' and 'changed'—as binary classes.

By framing the problem as a binary classification task, additional binary classification statistics [Canbek et al., 2017] can be used to further evaluate mutational consistency.

Mutational accuracy measures ACCURACY for SSE mutations in the reference and predicted structures. The calculation involves the following steps:

1. Compare the wild-type and mutated reference structures element-wise for each secondary structure element (SSE). This generates the **reference SSE mutation** sequence, indicating the type of change, e.g., 'EE' for no change in a β-strand or 'EI' for a β-strand changing into a π-helix.

2. Perform the same comparison for the wild-type and mutated predicted structures to produce the **predicted SSE mutation** sequence.

3. Calculate ACCURACY for the reference and predicted SSE mutation sequences, as outlined in the Secondary structure measures section, with the SSE mutation change types replacing the standard secondary structure classes.

For Q8, there are 64 possible SSE mutation classes, as each of the eight secondary structure classes can either remain the same or change into any other class after mutation.

Mutational precision is calculated as ACCURACY, SEGMENT OVERLAP, or SOV_REFINE between interlaced SSE sequences of the wild-type and predicted secondary structures. The process involves:

1. Interlacing the equal-length wild-type and mutated structures element-wise for each SSE. For a wild-type secondary structure $\mathbf{R}^{rep} = [r_1, r_2, \ldots, r_n]$ and a mutated structure $\mathbf{R}^{mut} = [\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n]$, the **interlaced SSE sequence** is defined as:

$$\mathbf{R}^{\sim} = [r_1, \hat{r}_1, r_2, \hat{r}_2, \ldots, r_n, \hat{r}_n].$$

2. This procedure is applied to both the reference and predicted structures, yielding the **reference interlaced SSE** and **predicted interlaced SSE**.

3. Finally, ACCURACY, SEGMENT OVERLAP, or SOV_REFINE is computed between the reference and predicted interlaced SSE sequences, as described in the Secondary structure measures section.

4.2.9. Tertiary structure assessment

Tertiary structure prediction methods typically output results in 'PDB' format, though some, like AlphaFold2, also support mmCIF format. For consistency, we used the 'PDB' output for all methods. To align with the process used for experimental structures, we converted the predictions to mmCIF format using MAXIT (`https://sw-tools.rcsb.org/apps/MAXIT`). We then applied DSSP to assign secondary structures to the mmCIF-formatted predictions. We normalized the output as outlined in the Secondary structure measures section to produce secondary structure predictions from tertiary structure predictions.

Similar to secondary structure prediction methods, tertiary structure predictions were evaluated based on local, distant, and global structural changes. Mutation types were determined uniformly across all methods, using the protein sequence most similar to the consensus (wild-type sequence) within each cluster.

As Alphafold2 took a considerably longer amount of time than other tertiary structure prediction methods, we utilized batch processing to increase its throughput. The details on batch processing are detailed in Supplementary section A.9.

RGN2 was originally developed as a Colab notebook, which is designed to be used online. We transformed the online scripts to be utilized locally for our experiments and for better ease of access to others. Details on the changes can be seen in Supplementary section A.10.

4.3. Results and Discussion

For each structure prediction method, we evaluated their performance using the ACCURACY, SEGMENT OVERLAP, and SOV_REFINE measures. These measures were calculated for different structural and mutation vicinities. The methods evaluated include: AlphaFold2 ('af2'), ColabFold ('colabfold'), ESMFold ('esmfold'), RGN2 ('rgn2'), SSPro8 ('sspro8'), Raptor-X Property ('raptorx'), SPOT-1D ('spot_1d'), SPOT-1D-Single ('spot_1d_single'), and SPOT-1D-LM ('spot_1d_lm').

Measures were analyzed for the following backbone vicinities: primary structure ('1d'), secondary structure ('2d'), tertiary structure ('3d'), and contact/distance vicinity ('contact'). Additionally, results were grouped by mutation vicinity types: local, distant, and global.

Performance results for top-, average-, and low-performing methods are provided as tables in Supplementary section A.6. These categories are based on performance trends, as shown in Fig. 4.12. In the figure, the error bars represent the 1st to 99th percentiles. This reveals the performance spread of the methods, indicating that even top-performing methods struggle with some proteins. Further details of the performance spread can be seen as a box plot in Supplementary section A.8.

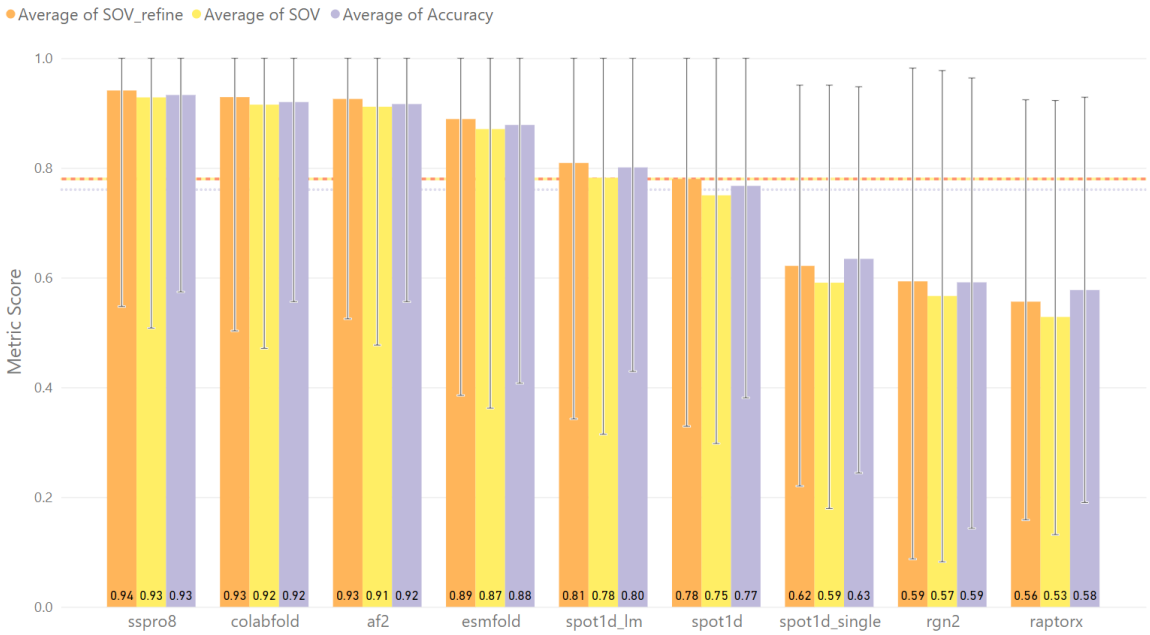


Figure 4.12.|: **Performance of each structure prediction method.** SSPro8, ColabFold, Alphafold2, and ESMFold perform higher than average. SPOT-1D-LM and SPOT-1D perform close to average. SPOT-1D-Single, RGN2, and Raptor-X Property perform below average. Therefore, methods are categorized according to their performance as 'Top', 'Avg', or 'Low' respectively.

4.3.1. Mutational measures

Since mutational consistency reduces structural changes to a two-class problem, we were expecting high scores across all prediction methods. Surprisingly, low-performing methods achieve similar scores to average- and top-performing methods. The exception is RGN2, which performs lower overall but still has higher minimum scores compared to other methods.

When binary classification statistics are applied, treating structural change ('C') as the positive class and structural preservation ('N') as the negative class, mutational consistency scores reveal more variation. However, since structural changes are rare, the classification problem is highly imbalanced.

Fig. 4.13 reveals that all methods exhibit high False Discovery Rates and False Negative Rates, along with low Positive Predictive Values and Sensitivity. This indicates that, regardless of their overall mutational consistency scores, the methods struggle to accurately predict mutational changes, often failing to detect structural changes when they occur. The high mutational consistency scores are largely due to the rarity of structural changes, resulting in a strong bias toward preserved structures.

Mutational accuracy follows a scoring trend similar to SOV_REFINE benchmark results, which calculate secondary structure measures on a protein-by-protein basis without focusing on mutations. However, mutational accuracy reduces scores across all methods by

approximately 4%. For example, SSPro8 achieves a mean mutational accuracy of 91%, compared to 95% in the protein-by-protein SOV_REFINE benchmark, providing a more realistic assessment of predictive performance. The mutational consistency and accuracy results are shown in Fig. 4.13.

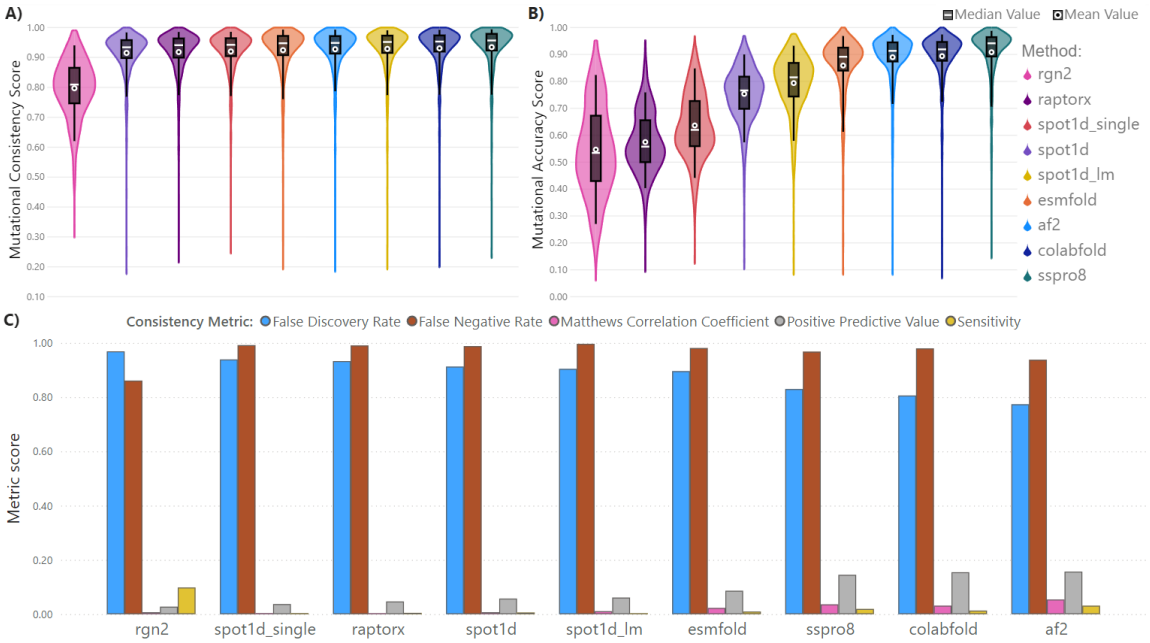


Figure 4.13.|: **Mutational consistency and Mutational accuracy.** Violin plots display the mutational consistency (A) and mutational accuracy (B) results for each structure prediction method. C) A bar graph presents the binary classification measures for mutational consistency across all prediction methods. This shows that all methods have deficiencies predicting if and when a mutational change will occur. The high scores in A and B come from the data imbalance of very few mutational secondary structure changes occurring.

Mutational precision using ACCURACY follows a similar trend to SOV_REFINE in Fig. 4.12, with Raptor-X Property scoring lowest, followed by RGN2, SPOT-1D-Single, SPOT-1D, and SPOT-1D-LM. When using SEGMENT OVERLAP or SOV_REFINE, the order changes, with RGN2 scoring lowest, followed by Raptor-X Property. Additionally, most measures show lower mean values when calculated with a Segment Overlap measure. Notably, mutational accuracy strongly correlates with mutational precision when using a Segment Overlap measure. The mutational precision results are shown in Fig. 4.14.

4.3.2. Mutation stability

Single amino acid mutations can produce stable results (no structural change), or disruptive (inducing structural change) results. In our mutational dataset, containing experimental structures from the PDB, stable mutations account for 128 out of 915 mutated proteins. Disruptive mutations were considered by their 1D vicinity: local and distant. Local disruptive mutations constitute 429 of the 787 total disruptive mutations. Distant disruptive mu-

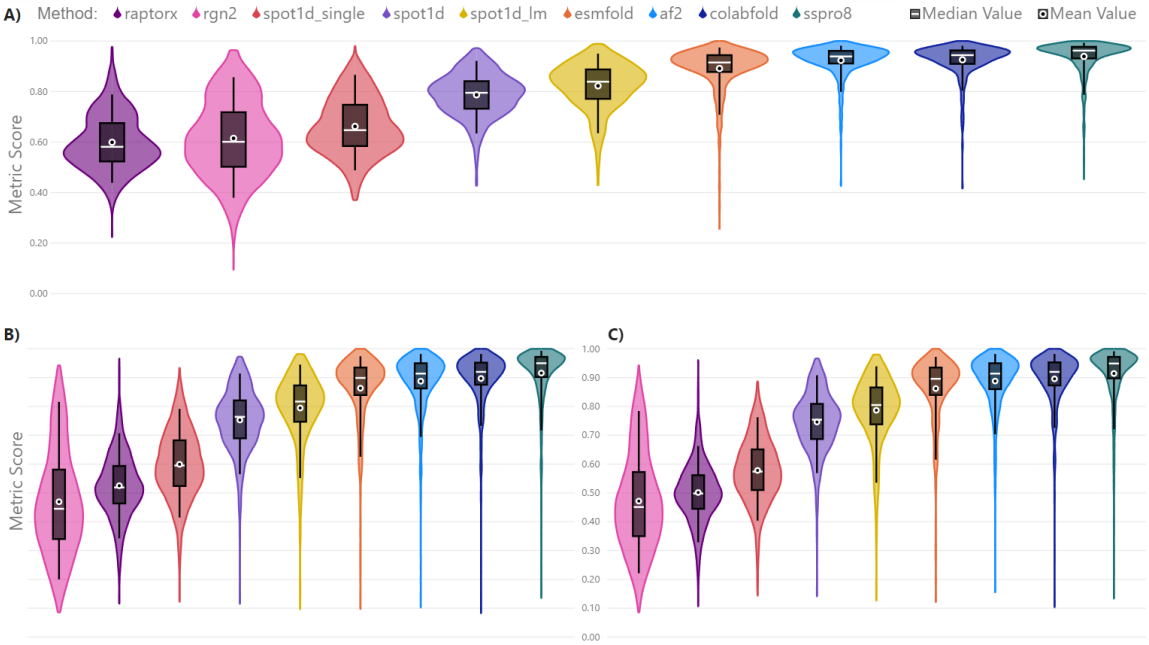


Figure 4.14.|: **Mutational precision.** Violin plots showing mutational precision for each prediction method using three different secondary structure measures: A) ACCURACY, B) SEGMENT OVERLAP, and C) SOV_REFINE. Results for mutational precision measures are very similar to their respective individual protein secondary structure measures.

tations are found in 698 mutated proteins. Finally, 340 disruptive mutations caused both local and distant structural changes. In contrast to the experimental data, structure prediction methods tend to infer a disruptive mutation for almost all proteins in our dataset. This can be of interest to structure prediction researchers, as even the best methods are still lacking on predicting stable mutations. In the PDB data, stable mutations are found mostly in transport proteins. This transport property of a protein is also found in mutations that cause a structural change in its distant vicinity, but not in its local vicinity. These results might follow from evolutionary pressure on keeping important functional structures, e.g. transport and complex-forming capabilities, as they affect downstream processes in the cell. From our selected prediction methods, SSPro8 is the most capable method for stable mutations. It is able to predict stable mutations in transport proteins, but struggles with disruptive mutations that only affect its local vicinity. This type of disruptive mutation deficiency for SSPro8 can be found in complex-forming proteins. This discrepancy may arise because some experimental PDB structures are resolved in their ligand-bound state, potentially biasing the data toward bound conformations, which may differ from unbound structures. Prediction methods, however, do not account for ligands and are designed to predict unbound structures. It is also possible for the extraordinary results from SSPro8 to be caused by its use of templates that contain most of our mutational dataset. The results for mutation stability data can be seen in Fig 4.15.

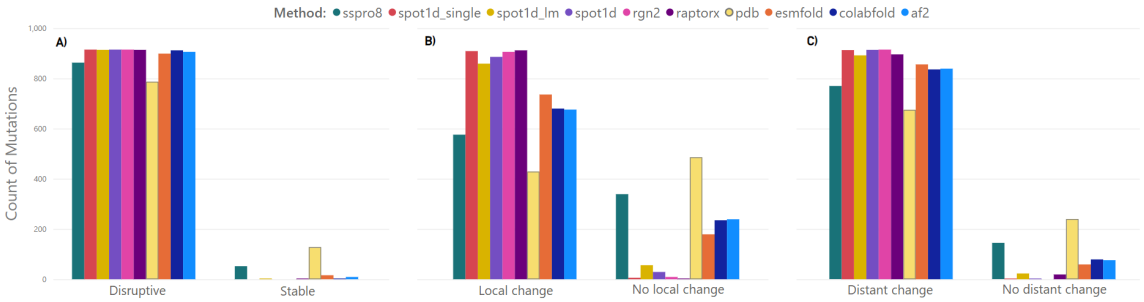


Figure 4.15.|: **Mutation stability results.** Number of mutations, for all prediction methods and experimental PDB data, with a disruptive or stable result in the A) complete secondary structure, B) Local vicinity, C) Distant vicinity of a mutation. Stable mutations occur more often in PDB data than in prediction methods, as the latter almost always predicts destabilizing mutations. The exception is SSPro8 while still missing two thirds of stabilizing mutations. PDB data also show that the local vicinity is more stable than not when a mutation occurs.

4.3.3. Mutation vicinity

Examining the vicinity results for each type of backbone change reveals that methods struggle more with accurately predicting local changes. The shorter maximum length of local changes compared to distant and global changes likely contributes to greater variability in performance. The lower average measures for local changes result from all methods performing slightly worse on these changes, as reflected in Fig. 4.16.

Focusing on disruptive mutations, PDB data reveals a few common single amino acid mutations that cause secondary structure changes, such as Asparagine (polar) to Histidine (positively charged), Serine (polar) to Threonine (polar), and Alanine (non-polar) to Leucine (non-polar). In contrast, many mutations rarely caused more than one secondary structure change, as shown in Fig. 4.17.

As previously seen in mutation stability results, secondary structural changes due to single amino acid mutations occur less frequently in experimental PDB data than in predicted structures. The SSE classes that produce the previously mentioned discrepancies can be seen in detail in Fig. 4.18. In our dataset, π-helices commonly transition into α-helices, loosening the helix structure. Less frequently, π-helices dissolve into hydrogen-bonded turns or bends. They never transition into β-sheets, bridges, 3-10 helices, or coils, as such changes would require significant energy, likely exceeding what a single mutation can induce. This thermodynamic constraint appears to be absent in structure prediction methods, limiting their ability to produce more realistic predictions.

4.3.4. Prediction difficulty for methods

Most structure prediction methods struggle to predict stable mutations (i.e. single amino acid mutations that do not cause a change in the structure), even though such mutations

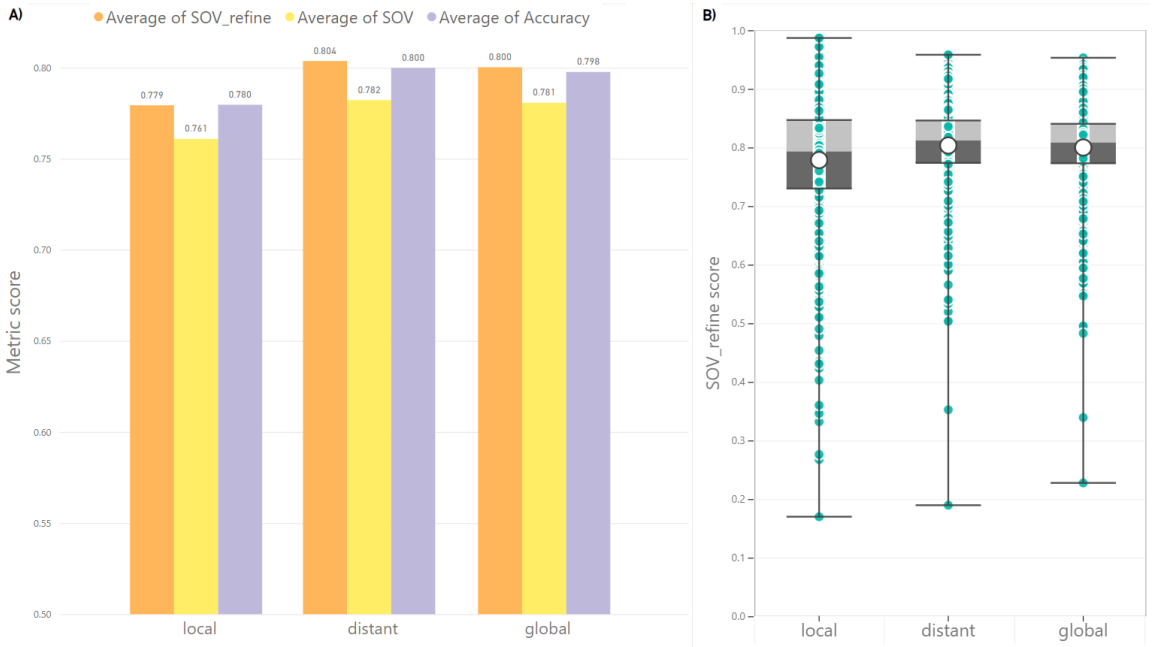


Figure 4.16.|: **Statistics for Type of Backbone changes.** ACCURACY, SEGMENT OVERLAP and SOV_REFINE measures for each type of backbone change. A) Bar graph showing the average performance across the three measures, highlighting lower accuracy in predicting local changes. B) Box plot of SOV_REFINE values for each type of backbone change, illustrating a wider spread in local change predictions, ranging from the highest to the lowest overall results.

occur in reality. Transport proteins are a notable exception, as these proteins are inherently stable, and minor prediction errors do not significantly affect their structure. This stability allows transport proteins to often be correctly predicted, aligning with their actual behavior. However, transport proteins also appear among incorrectly predicted structures, highlighting inconsistencies in prediction accuracy for these proteins.

Incorrectly predicted proteins frequently include membrane and structural proteins, which often form complexes and contain β-sandwich structures characterized by anti-parallel β-sheets. These features are also common in proteins with stable mutations. Therefore, the beta structures found in these types of proteins could be the factor leading to inaccurate predictions for stable mutations.

These findings are illustrated in Fig. 4.19.

4.3.5. Method comparisons

In this section, we analyze the similarities and differences between methods in each performance category by examining their best- and worst-predicted proteins using three secondary structure prediction measures.

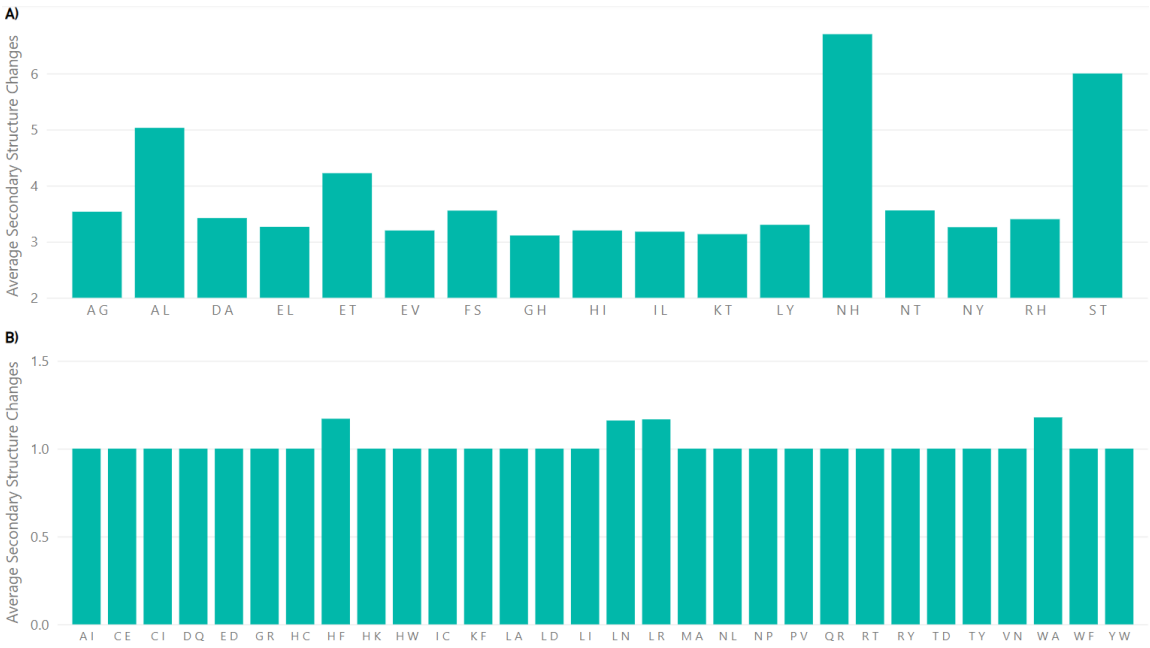


Figure 4.17.|: **Amino acid mutations results.** This data shows mutations in two letter codes. The first letter is the wild-type amino acid and the second letter is the mutated amino acid. A) Most common disruptive mutations in our dataset. B) Least common disruptive mutations that appear in our dataset.

Among the top-performing methods, the best-predicted protein chains often include immunoglobulin-like β-sandwich domains [Clarke et al., 1999], one of the most common structural motifs. These domains are present in a wide variety of proteins, including those in the extracellular matrix, muscle proteins, immune system proteins, cell-surface receptors, and enzymes.

The best-predicted protein chains for average-performing methods often include transport proteins involved in nuclear and cytoplasmic transfer. Some also relate to lipid binding, suggesting an association with the lipocalin family. The lipocalin family [Flower et al., 2000], characterized by an antiparallel β-barrel structure surrounding its binding site, transports small hydrophobic molecules such as lipids and binds to complexed iron molecules and heme.

The best-predicted protein properties for low-performing methods include heme binding and periplasmic activity, suggesting an association with periplasmic heme-binding proteins. In bacteria, these proteins are part of the heme acquisition system, transferring heme across the periplasmic space from the outer membrane for energy acquisition. These results are shown in Fig. 4.20.

The worst-predicted proteins for top performing methods are similar to the overall mispredicted proteins. These include membrane and structural proteins with a connection to RNA, suggesting an association with membrane-associated RNA-binding proteins. These proteins play a role in organelle-coupled translation, facilitating efficient protein localization within the cell.

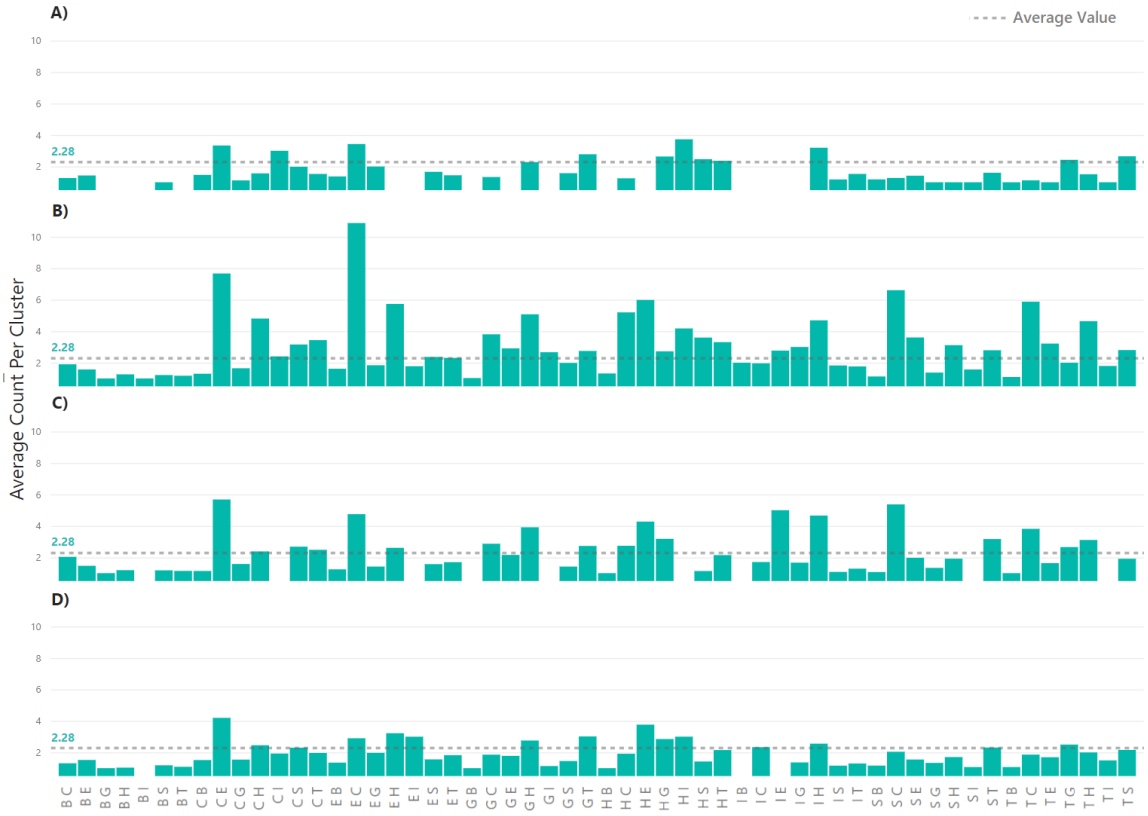


Figure 4.18.|: **Secondary structure mutations results.** All bars show the average number of times a secondary structure mutation occurred in a cluster. Secondary structure mutations follow a two letter code. First letter is the secondary structure assigned to the wild-type amino acid, while the second letter is the secondary structure assigned to the mutated amino acid. A) PDB data from experimentally obtained structures. B) 'Low' performing structure prediction methods. C) 'Average' performing structure prediction methods. D) 'Top' performing structure prediction methods.

The worst-predicted proteins for average performing methods are primarily associated with viral proteins and some membrane activity, suggesting a link to surface proteins critical for viral infection. The prediction challenges may stem from the diverse binding capabilities of these proteins. Additionally, the receptor-binding site flexibility of surface proteins, which is not captured in the static, crystallized structures within the PDB, could further contribute to the increased difficulty in prediction.

The worst-predicted proteins for low performing methods also include transport and metal ion binding properties, such as heme binding. Interestingly, these properties were also present in the best-predicted proteins, indicating inconsistency in the performance of low-performing methods when predicting certain protein types. As a result, no clear association between prediction performance and specific protein types could be identified. While low-performing methods exhibit some overlapping protein properties, no significant patterns were observed. The worst-predicted proteins for different method categories are

Figure 4.19.|: **Overall protein structure prediction results.** Structural properties are an agglomeration of protein descriptors from CATH, SCOP, and PDB. The proteins are named in the following manner: PDB ID (underscore) Protein chain. A) Best overall predicted proteins and their structural properties. B) Worst overall predicted proteins and their structural properties.

shown in Fig. 4.21.

4.3.6. Methods strengths and weaknesses

The structure prediction methods analyzed in this work employ distinct methodologies, resulting in varying performance outcomes. This section highlights the protein properties that posed challenges for each method.

AlphaFold2 struggles with the same protein properties observed in the overall worst-predicted proteins for top performing methods. This is unsurprising, as AlphaFold2 and ColabFold produce similar results due to their closely related algorithms. Their similarity biases the top performing category by contributing a disproportionate number of similar results.

Both AlphaFold2 and ColabFold perform well overall, but their predictions for mutation stability are notably weak, underscoring the need for further research on stable mutation prediction. Between the two, ColabFold is optimized for efficiency, making it the more favorable choice for this study.

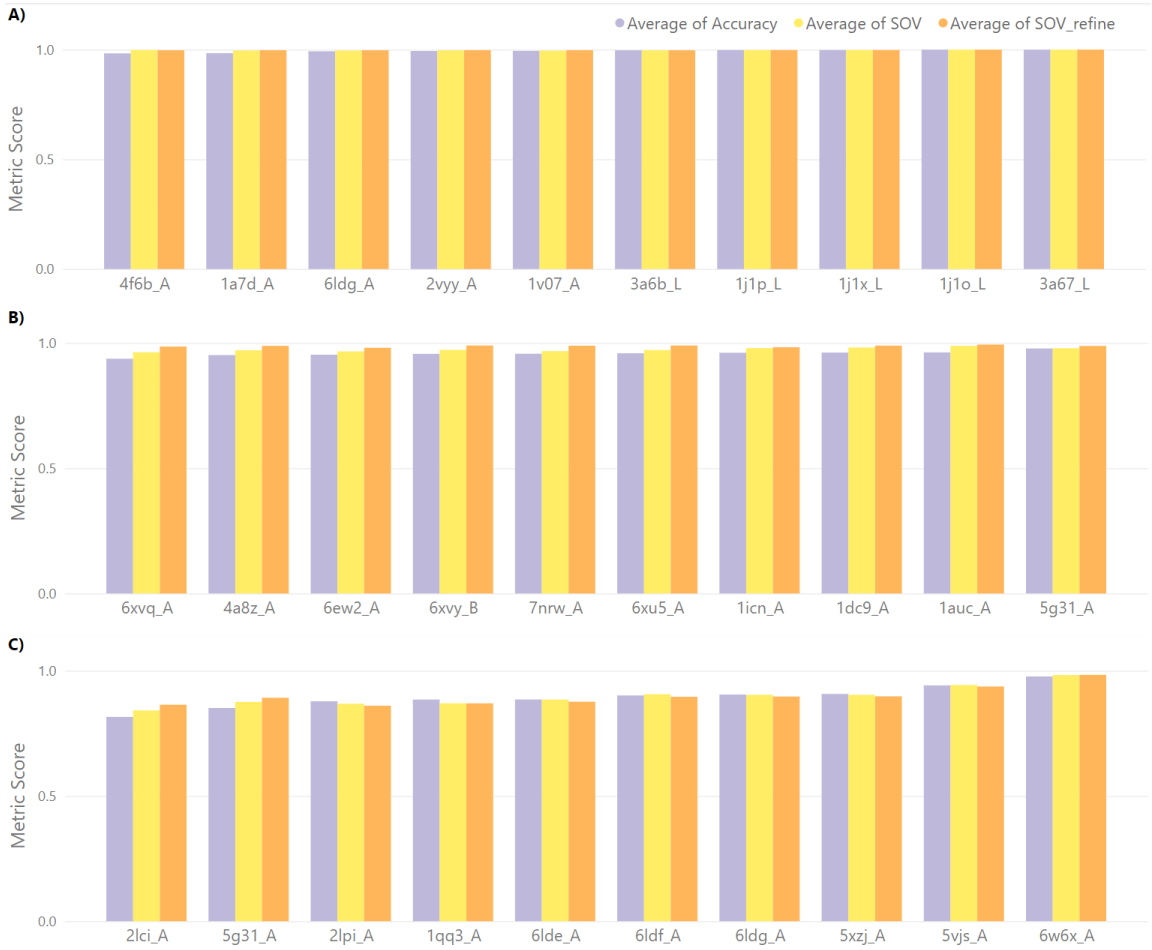


Figure 4.20.|: **Best results per method category.** Best predicted proteins for each method performance category. There are only a few proteins in multiple performance categories. A) Top performing methods. B) Average performing methods. C) Low performing methods.

ESMFold, like AlphaFold2 and ColabFold, struggles with transport proteins. However, as a language model, ESMFold does not require MSAs, significantly reducing its prediction time by an order of magnitude. While its overall performance is comparable to AlphaFold2 and ColabFold, ESMFold has a slightly lower average accuracy. Nevertheless, its faster prediction speed makes it a strong alternative for handling higher workloads.

SSPro8 stands out as the only top performing method specifically designed for secondary structure prediction. Unsurprisingly, it achieves the best overall performance for our task. Analyzing its worst predictions reveals no clear pattern, suggesting that SSPro8 is a robust solution without bias toward specific protein types. The results for each top performing method are shown in Fig. 4.22.

The average performing methods, SPOT-1D and SPOT-1D-LM, struggle with predicting proteins related to viruses and RNA binding. Additionally, SPOT-1D has difficulty with metal ion binding proteins, a challenge also observed in top performing methods. Both

Figure 4.21.|: **Worst-predicted proteins per method category.** The worst-predicted proteins along with their properties for the different method performance categories. Many proteins are equally incorrectly predicted among all performance categories. A) Top performing methods. B) Average performing methods. C) Low performing methods.

are secondary structure prediction methods that underperform compared to their tertiary structure counterparts. As a language model, SPOT-1D-LM achieves higher accuracy than SPOT-1D by not relying on MSA evolutionary information, while SPOT-1D's reliance on MSAs mirrors the approach of AlphaFold2 and ColabFold. The results for these methods are shown in Fig. 4.23 A and Fig. 4.23 B.

Low performing methods lack a performance consensus, as their varied methodologies lead to differing results.

Raptor-X Property struggles with virus-binding proteins, but no consistent pattern was found among other poorly predicted proteins. Despite its low performance, it is the fastest prediction method among all analyzed.

SPOT-1D-Single is arguably the best among the low performing methods, with slightly higher overall performance than RGN2 and Raptor-X Property. However, it has difficulty

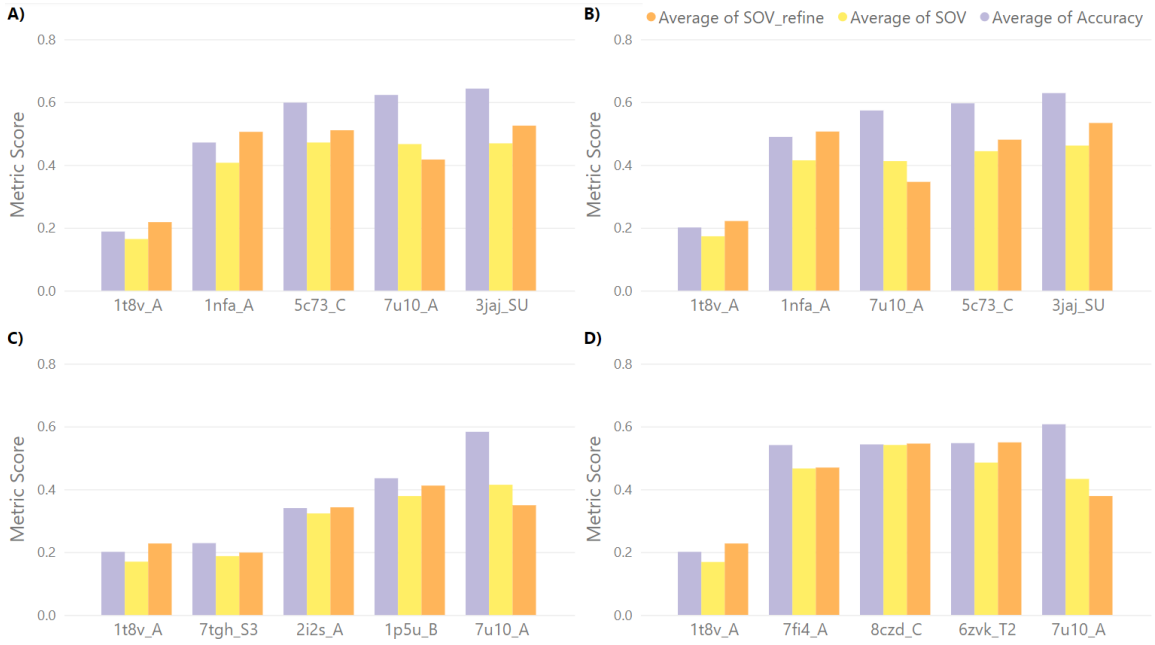


Figure 4.22.|: **Limitations on Top performing methods.** Worst performing proteins for each of the top performing methods. Alphafold2 and Colabfold have very similar performance and thus perform the same prediction mistakes. ESMFold and SSPro8 have very different methodologies to the other two methods and thus perform differently. '1t8v_A' is commonly predicted incorrectly across all methods. Prediction methods: A) AlphaFold2, B) ColabFold, C) ESMFold, D) SSPro8.

predicting membrane and lipid transport proteins, where average-performing methods excel.

RGN2, a 3D structure prediction method, is fast but ranks low in performance. Its speed depends on a structure relaxation process, which can take longer for poorly predicted structures. RGN2 struggles with proteins related to regulation, metabolism, and metal ion binding.

The results for each low performing method are presented in Fig. 4.23 C, Fig. 4.23 D, and Fig. 4.23 E.

An exceptional case in our dataset involves a protein (PDB ID 1t8v, a fatty-acid binding protein) that posed significant challenges for all structure prediction methods but was predicted more accurately by RGN2, a low-performing method. As shown in Fig. 4.24, RGN2 outperformed all other methods for chain A of this protein. Notably, this protein has few homologous sequences, with only 14 entries exceeding 90% similarity in our non-redundant dataset aligned using protein-BLAST.

Figure 4.23.|: **Limitations on Average and Low performing methods.** Worst performing proteins for each of the 'average' and 'low' performing methods. As with top performing methods, '1t8v_A' is commonly predicted incorrectly. Prediction methods: A) SPOT-1D, B) SPOT-1D-LM, C) SPOT-1D-Single, D) Raptor-X Property, E) RGN2.

4.3.7. Temperature factor and confidence results.

The Temperature factor [Sun et al., 2019] (TF) in crystallographic data measures the attenuation of the X-rays by thermal motion due to atomic vibrations. This can lead to inaccuracies in the crystallographic data as the atoms' positions become difficult to discern.

Here, we investigate possible correlations with the TF of proteins within the PDB, and the confidence scores produced by tertiary structure prediction methods. This is done by comparing the TF and confidence score in relation to the single amino acid mutation position in the protein. Unsurprisingly, no correlation could be found for the TF, as it is prone to bias from experimental conditions, e.g., thermal motion, or crystal purity.

Confidence scores for each predicted secondary structure, or location of C_α-atom for each amino acid, also did not provide a strong correlation to the single amino acid mutation location. In some cases, the variance of the confidence score can become the maximum value near the mutation location; but in general, the confidence score of the method, which is present where the TF would be in PDB data, is not predictive of mutation location. These results can be visualized in Fig. 4.25.

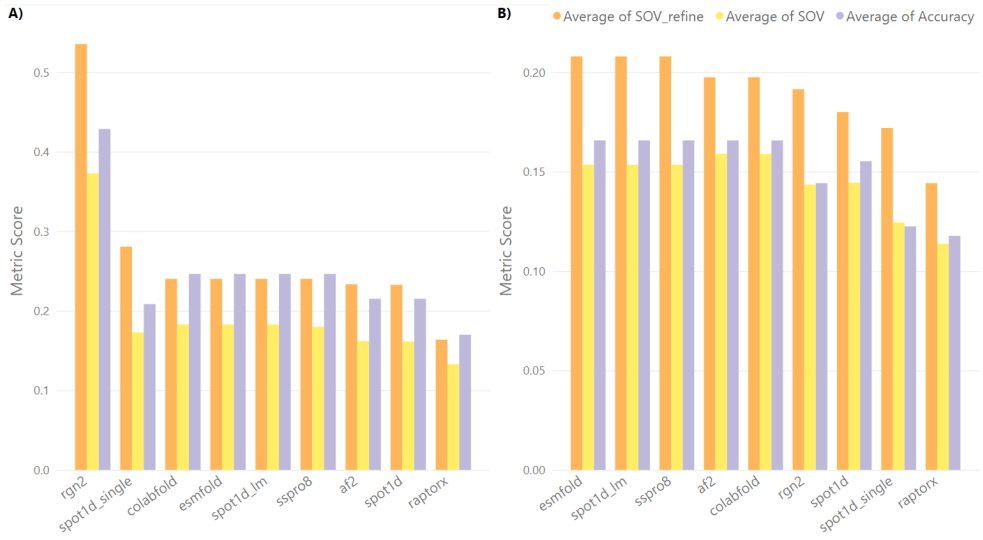


Figure 4.24.|: **Exceptional Case: RGN2** A challenging protein to predict for all methods (PDB ID 1t8v_A), where the low performing method RGN2 outperforms all others. A) Performance difference from all other methods on local vicinity. B) No difference or very low difference to other prediction methods for distant vicinity.

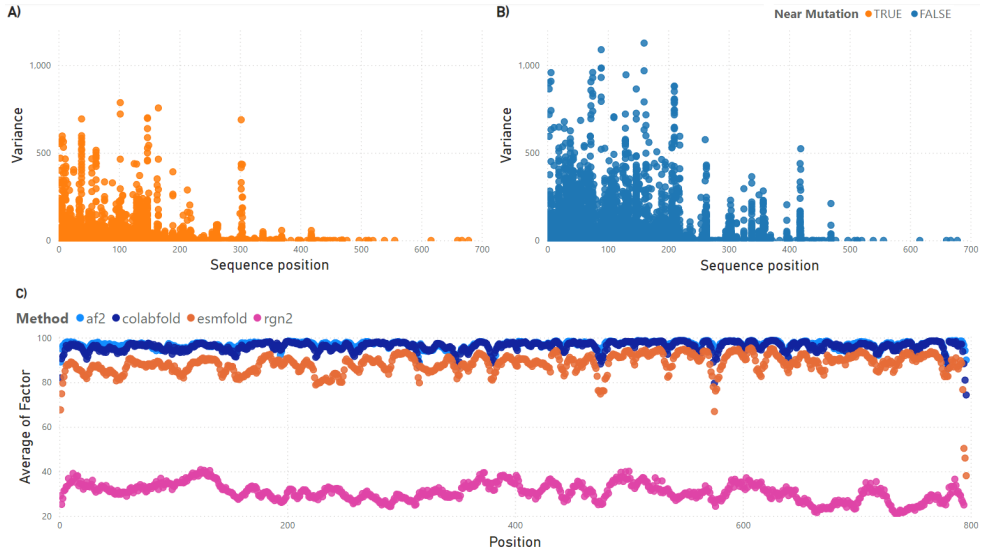


Figure 4.25.|: **Temperature factor and confidence results.** No significant correlation to single amino acid mutations found for both Temperature factors in PDB data and confidence scores in predictions. A) Variance value in Temperature factor when a mutation is near the sequence location. B) Variance value in Temperature factor when a mutation is far from the sequence location. C) Confidence values for all tertiary structure prediction methods in mutations around position 260. This position have low variance when a mutation is near, but high variance when a mutation is not near. As seen from the figure, there is no indication that a mutation has taken place around position 260 from the confidence scores.

4.4. Conclusions

State-of-the-art protein structure prediction methods were evaluated on predicting backbone secondary structure changes caused by single amino acid mutations. To our knowledge, this is the first evaluation of protein prediction capabilities at different mutation vicinity levels. For this purpose, we created a dataset of single amino acid mutations containing primary, secondary, and tertiary structures derived from experimental data. The evaluation includes five secondary structure prediction methods and four tertiary structure prediction methods, each employing vastly different methodologies.

Our analysis reveals that all methods struggle to predict stable mutations—those that do not cause structural changes—often favoring secondary structure element changes even when experimental data does not support them. Experimental data shows logical patterns for secondary structural changes, such as 3-10 helices transforming only into α-helices, bends, or turns. In contrast, prediction methods lack the capability to infer the improbability of drastic changes, such as a 3-10 helix turning into a β-sheet from a single amino acid mutation.

Although the benchmarking dataset is limited by the small amount of available experimental mutation data, it remains a reliable evaluation tool since models generally avoid training on high-homology sequences. This ensures that the dataset effectively tests how models handle proteins with single amino acid mutations they have not encountered during training.

The dataset provided in this work can aid in the training and testing of future models for single amino acid mutations. Our results demonstrate that while current prediction models achieve high accuracy, they exhibit weaknesses that must be addressed when working with mutational data. With this knowledge, it should be possible to refine existing models using our dataset to develop a more stable, mutation-aware secondary structure prediction method.

Additionally, protein chains involved in complexes or requiring ligand binding to adopt their crystallized structure, as observed in PDB data, highlight the need for models that account for bound molecules in their predictions. Membrane and transport proteins in our dataset illustrate these challenges, as even top performing methods struggle to predict their structures accurately.

Finally, the prediction of low-homology proteins may benefit from an ensemble of structure prediction methods with varying homology requirements. This approach could leverage methods that do not rely on homology, which have shown the potential to outperform top-performing methods in cases where high homology information is unavailable.

Chapter 5

Single amino acid mutation knowledge can decrease prediction inaccuracies on protein secondary structure

Contents

Correspondences in This Chapter

Addressed Research Question(s):

Q3— Are any of the selected structure prediction methods sufficiently precise to show the effect of single amino acid mutations on protein backbone structure?

Q3.1— If not sufficiently precise, what are possible improvements that can be made to these methods regarding single amino acid mutations?

Q4— Are secondary structure prediction methods redundant now that tertiary structure prediction methods have reached comparable performance to the former?

Protein tertiary structure prediction models like AlphaFold2 have revolutionized the field with unprecedented accuracy. Yet predicting structural changes arising from single

amino acid mutations remains a challenge. The complexity introduced by these mutations calls for models that can incorporate mutational information into their predictions. We propose a novel refinement strategy for protein secondary structure prediction that leverages single amino acid mutational data. As part of this strategy, we introduce *Mut2Dens*, a model that not only yields improved consistency of predictions for mutational data, but also maintains robust predictive performance on non-mutational datasets. Mut2Dens takes multiple predicted secondary structures and generates a mutation-aware secondary structure. This awareness comes from our mutational dataset, learning to avoid common mistakes in prediction methods after a single amino acid mutation occurs. In particular, Mut2Dens employs the extremely randomized trees (ExtraTree) algorithm to avoid overfitting and makes effective use of the limited mutational data available from experimentally determined three-dimensional structures. By combining predictions from highly accurate structure prediction models, we create an ensemble that integrates their strengths while enhancing mutational capabilities. This refinement strategy also improves the non-mutational performance of state-of-the-art methods by addressing their most inaccurate and least confident predictions. Moreover, it reduces improbable outcomes in mutated protein structures—such as transforming π-helices into β-sheets—that can still occur in current prediction models. Finally, by using interpretable machine learning algorithms (e.g., ExtraTree), we can reveal the underlying biological knowledge from the refinement model; the insights gained from Mut2Dens can be corroborated with known mutational outcomes, helping users pinpoint discrepancies across structure prediction models and make more informed decisions regarding the predicted structures.

5.1. Introduction

The three-dimensional structure of a protein is directly correlated with its function and is partly defined by its amino acid sequence [Kuhlman and Bradley, 2019]. Recently, protein tertiary structure prediction models have become highly accurate [Jumper et al., 2021] for proteins with known homology. Despite these achievements, it is an open question on whether these models are able to correctly predict structural outcomes caused by single amino acid mutations [McBride et al., 2023, Keskin Karakoyun et al., 2023].

Protein three-dimensional structure is inherently noisy due to the dynamic nature of atomic positions. This includes atomic vibrations, environmental conditions during experimentation, and the inherent flexibility of protein structures. This noise can make it challenging to obtain a precise and stable representation of the protein's architecture. Protein secondary structure helps mitigate these inconsistencies and variations introduced by noisy atomic coordinates and atomic-level fluctuations by discretizing the atomic-level details, thereby elucidating mutational effects in the protein's architecture or backbone structure. Specifically, the distances between backbone atoms are used to infer backbone bonds, which can then be classified into secondary structure elements (SSEs) by secondary structure assignment algorithms such as DSSP [Kabsch and Sander, 1983a]. This removal of

confounding atomic-level variations introduced by experimental and physical factors can increase the reliability of observed structural changes.

Recent studies [McBride et al., 2023, Keskin Karakoyun et al., 2023] evaluated highly performing structure prediction models on single amino acid mutations. These evaluations, however, only focused on backbone information that utilize *three-state secondary structure* (Q3). As the name implies, Q3 classifies all SSE into three classes characterized by their secondary structure—α-helix, β-sheet, and coil. However, as documented in the literature [Kabsch and Sander, 1983a, Magnan and Baldi, 2014], Q3 is insufficient to account for the complete structural information of the protein backbone. Moreover, as existing methods for predicting secondary structures near the theoretical accuracy limit of nearly 90% for three-state predictions [Yang et al., 2018], attention has turned to the more complex challenge of predicting *eight-state secondary structure* (Q8). While the theoretical limit of Q8 accuracy is not well established, current template-less methods achieve an accuracy of about 75% [Sidi and Keasar, 2020].

Evaluating secondary structure prediction requires a consistent secondary structure assignment to serve as a *gold standard*. However, most secondary structure assignment algorithms produce varying results [Antony et al., 2021], especially in proteins with irregular conformations [Zhang and Sagui, 2015]. We selected DSSP [Kabsch and Sander, 1983a], a pioneering algorithm for secondary structure assignment, as the ground truth for machine learning (ML) purposes because it has been extensively tested and widely used.

Although the overall predicted secondary structure remains accurate, these models often locate mutational changes incorrectly, rendering them unsuitable for mutational prediction. Therefore, we propose a refinement strategy for protein secondary structure prediction that enhances already accurate models for mutational data and further benefits their non-mutational predictions. This approach utilizes an ensemble model, Mut2Dens, which is trained on single amino acid mutation data, and increases the prediction scores of secondary structures by at least 20% in low-scoring proteins. Moreover, Mut2Dens demonstrates substantial improvements in maintaining mutational consistency for altered secondary structure elements.

5.2. Materials and Methods

As in chapter 4, this investigation utilized the mutational dataset detailed in Data acquisition and processing. The prediction methods evaluated in this chapter utilize the measures in Secondary structure measures, as well as the mutational measures proposed in Mutational measures. These measures were calculated using our calculation tool described in Measures calculation for its calculation efficiency over existing tools.

5.2.1. Input data

We investigated multiple ways in which to represent the data used as input to our ensemble model. These input representations differ due to the requirements of the type of machine learning algorithm. For neural-type, the data was organized into a three-dimensional $(i \times j \times k)$-matrix containing one-hot encoded vectors. Here, i denotes the number of secondary structure classes (which is eight for Q8 prediction), j denotes the sequence length and k denotes the number of structure prediction methods.

Machine learning frameworks for designing tree-type algorithms, e.g. Scikit-learn [Pedregosa et al., 2011], require each sample to be a one-dimensional vector to ensure consistency of the feature representation where each element in the vector corresponds to a single feature from the sample. Therefore, tree-type algorithms were limited to a one-dimensional vector of features, which could be organized into different representations. We represented sample vectors in the following manners:

- Nominal data: $(n \cdot m)$-vector, where n denotes the sequence length and m denotes the number of predictors. See Fig. 5.1 C for an example of this representation with 3 predictors and a sequence length of 9.
- Windowed nominal data: Same as nominal data but the sequence is truncated to a specific window size. The full sequence is processed through a sliding window with a step size of 1, where the centre location of the window has the amino acid of interest. See Fig. 5.1 D for an example containing a window size of 7.

Because the input sequence length can vary, we use a window-based approach that allows the user to specify the maximum length. We tested prime numbers (starting at 3) for the window lengths as the window sizes need to be odd to place the amino acid of interest at the center. Prime numbers were specifically selected to follow the distribution from the prime number theorem [Newman, 1980] — prime numbers become less common as they become larger. Window sizes in our data seem to affect outcomes more sensitively the shorter they were, and thus we surveyed shorter sized windows more thoroughly than longer ones. As a first step, we used the maximum sequence length (1024) supported by some predictors (e.g., ESMFold), resulting in a maximum window length of 1021. We then tested progressively smaller window lengths down to a minimum of 3, yielding 171 different window sizes overall to evaluate model performance. To make sure we did not omit any amino acids on longer window sizes, we utilized zero-padding on both edges of the sequences. This also allowed the model to know when the start and end of the sequence occurred.

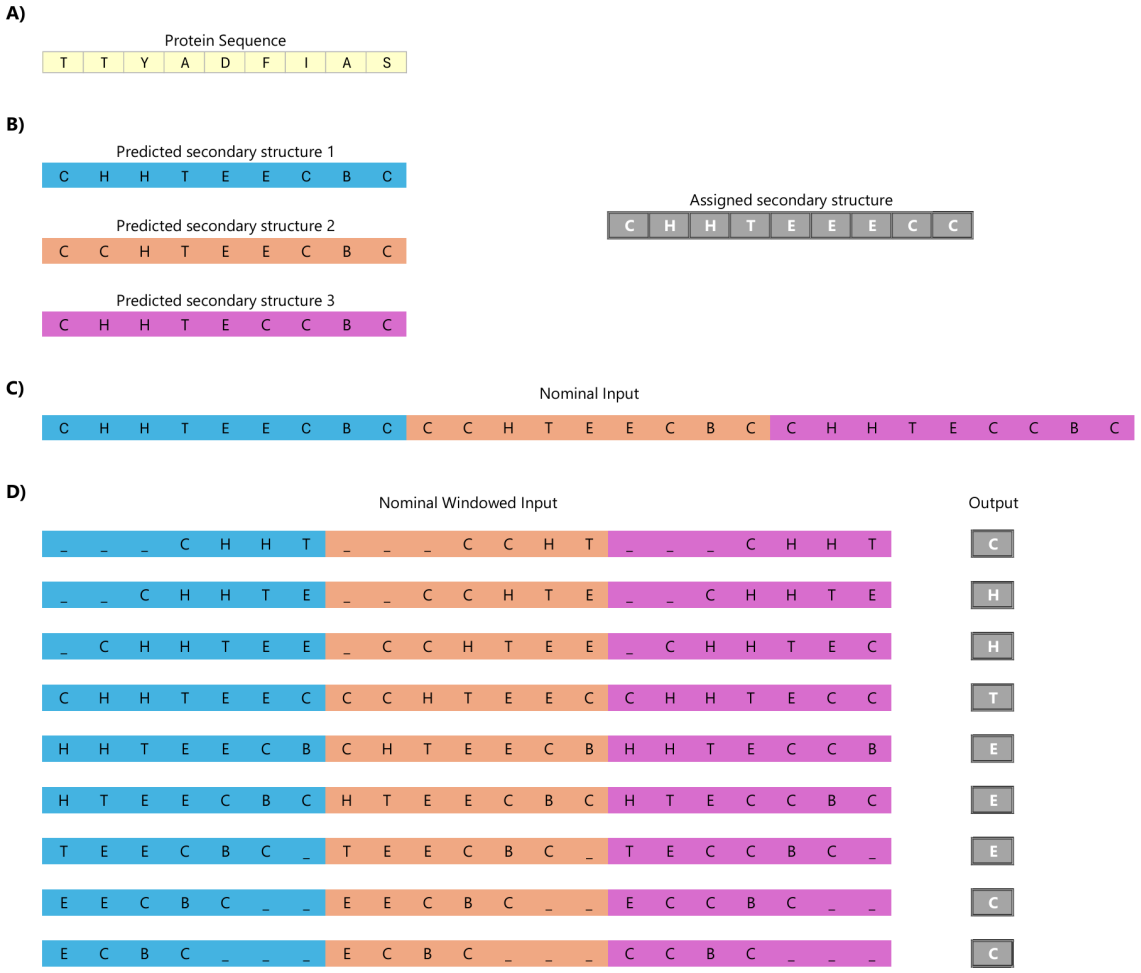


Figure 5.1.|: **Data representation.** Secondary structure representation as input features for machine learning and feature selection procedures. A) Example of a protein sequence of length 9. B) Output from three structure predictors and the DSSP-assigned secondary structure to the protein sequence. The assigned structure is utilised as the truth label or expected outcome. C) Nominal data representation. This representation concatenates all predictions with their full length. Therefore, the complete sequences for the predicted and assigned secondary structures are utilised. D) Windowed nominal data representation. Window size of 7. To differentiate parts of the sequence, the data is padded with empty spaces. Each row represents an input with a position of interest. This position is located in the middle of the window, and also contains the expected output or label.

5.2.2. Prediction performance spread

Prediction methods can also be assessed by the distribution spread of their performance measure values along different secondary structure measures. To compare different prediction methods, their evaluation must be measured on a common set of protein sequences $\mathbf{S}^*$. For our purposes, this dataset should contain protein sequences for which a prediction method m will predict a secondary structure. The secondary structure assignment to these sequences is also required, and are obtained through DSSP.

Formally, let $\mathbf{A} = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$ be a finite set containing the alphabet of standard amino acids as defined by residue or side chain. A **protein sequence** over $\mathbf{A}$ is a finite sequence of amino acids $\mathbf{S} = [a_1, \ldots, a_n]$ of length n, where $a_i \in \mathbf{A}$, for all $1 \leq i \leq n$. a_1 is the N-terminus residue and a_n is the C-terminus residue. For a protein sequence $\mathbf{S}$ each amino acid a_i is assigned to a secondary structure element r_i resulting in the complete protein secondary structure $\mathbf{R} = [r_1, r_2, \ldots, r_n]$. Then, every secondary structure element must be assigned a DSSP class $r_i \in \Upsilon_8$, where $\Upsilon_8 = \{\mathcal{C}, \mathcal{H}, \mathcal{E}, \mathcal{G}, \mathcal{I}, \mathcal{T}, \mathcal{S}, \mathcal{B}\}$ is the set of possible SSE classes in DSSP.

A prediction method m is defined in Eq. 5.1 as a learnable function that maps the protein sequence $\mathbf{S}$ to a predicted secondary structure sequence $\hat{\mathbf{R}} = [\hat{r}_1, \ldots, \hat{r}_n]$, where $\hat{r}_i \in \Upsilon_8$.

$$m : \mathbf{S} \rightarrow \hat{\mathbf{R}} \tag{5.1}$$

The function m can be represented as any machine learning model that takes a protein sequence $\mathbf{S}$ as input and outputs a predicted secondary structure sequence $\hat{\mathbf{R}}$.

Let the results dataset $\mathbf{D}_m^{\mathbf{S}^*} = \{(\mathbf{R}, \hat{\mathbf{R}})\}$ for prediction method m be a set of tuples containing assigned and predicted sequences for each protein sequence $\mathbf{S} \in \mathbf{S}^*$, for $\mathbf{S}^* = \{\mathbf{S}_1, \ldots, \mathbf{S}_z\}$. Then, each tuple can be scored by any secondary structure measure defined above, where $\mathbf{R}^{ref} = \mathbf{R}$ and $\mathbf{R}^{pred} = \hat{\mathbf{R}}$.

The standard deviation is a conventional measure of spread for a distribution when its mean is the most appropriate measure of the distribution centre. Simply using standard deviation as a measure of distribution spread does not account for the performance of the prediction method. This can lead to low performing prediction methods with a low standard deviation to be considered superior to higher performing prediction methods. To account for this, we define two types of spread for a results dataset $\mathbf{D}_m^{\mathbf{S}^*}$. We start with Extreme spread, $XTSpread$, defined as:

$$XTSpread(\mathbf{D}_m^{\mathbf{S}^*}, t) = \max(Score_t(\mathbf{D}_m^{\mathbf{S}^*})) - \min(Score_t(\mathbf{D}_m^{\mathbf{S}^*})) - \overline{Score_t(\mathbf{D}_m^{\mathbf{S}^*})} \tag{5.2}$$

where the $Score_t$ function returns a list of scores calculated from a secondary structure measure t for every tuple in $\mathbf{D}_m^{\mathbf{S}^*}$. As the dataset contains multiple proteins and a prediction for each protein, the secondary structure measures result in a distribution of scores over $\mathbf{D}_m^{\mathbf{S}^*}$. Thus, we can obtain the maximum value $\max(Score(\mathbf{D}_m^{\mathbf{S}^*}))$ and minimum value $\min(Score(\mathbf{D}_m^{\mathbf{S}^*}))$ for the distribution, along any distribution statistic such as its mean value, $\overline{Score(\mathbf{D}_m^{\mathbf{S}^*})}$, and standard deviation, $\sigma(Score(\mathbf{D}_m^{\mathbf{S}^*}))$. $XTSpread$ is defined as the difference between the distribution's range and its mean. This measure decreases when the spread of the data is narrower and the mean is higher. It captures the most extreme values—the best and worst predictions—making it particularly useful for assessing low-confidence predictions and quantifying the potential impact of large errors.

Similarly, we define Standard spread, $STSpread$, as:

$$STSpread(\mathbf{D}_m^{\mathbf{S}^*}, t) = \sigma(Score_t(\mathbf{D}_m^{\mathbf{S}^*})) - \overline{Score_t(\mathbf{D}_m^{\mathbf{S}^*})} \tag{5.3}$$

which, like the $XTSpread$, returns a lower value when the spread is narrower and the mean is higher. However, because $STSpread$ does not account for the entire distribution range, it excludes extreme values. Consequently, $STSpread$ captures the overall performance of a prediction method without considering outcomes distributed outside a single standard deviation of its mean.

5.3. Results and Discussion

This study initially examines each top-, average-, and low-performing predictor, grouped according to their significance in three different feature selection algorithms. In chapter 4, each predictor's performance was determined based on its mutational capabilities, which confirmed the importance of each group. Specifically, the top-performing predictors include AlphaFold2, ColabFold, ESMFold, and SSPro8; the average-performing predictors include SPOT1D and SPOT1D-LM; and the low-performing predictors include SPOT1D-Single, RGN2, and Raptor-X Property.

In addition, we compare different machine learning algorithms that can be used during our refinement strategy. These comparisons justify our choice of algorithm for creating the proof-of-concept ensemble model, Mut2Dens. The creation of Mut2Dens is intertwined with our refinement strategy, and as such we describe this strategy alongside Mut2Dens. We then show the performance results on test data using the best-performing ML algorithm and input representation combination that formed our ensemble model, along different versions of Mut2Dens depending on its input predictors. The multiple versions of Mut2Dens allow us to investigate how different types of predictors can affect the refinement output. Finally, we provide an in-depth analysis of the ensemble approach, illustrating how the ensemble model refines the secondary structures and its potential benefits and drawbacks.

5.3.1. Feature selection

Feature selection algorithms are highly dependent on the data representation they receive. As with tree-type ML algorithms, the inputs can be structured in various ways. In our initial strategy, we used nominal data with the full sequence length for each prediction method. Each method's feature importance was calculated by summing the contribution of all secondary structure elements (SSEs) it assigned, effectively weighting each method by its cumulative SSE impact. However, as shown in Fig 5.2 A, the algorithms had difficulty discerning feature significance under this approach, and no clear consensus emerged across different feature selection methods.

To address this issue, we adopted a windowed input approach. This yielded more cohesive results, with the top predictors—AlphaFold2, ColabFold, ESMFold, and SSPro8—being consistently identified by all three feature selection algorithms (Fig 5.2 B), a finding that aligns with our earlier analysis in chapter 4. Meanwhile, Raptor-X Property ranked among the least significant methods in both χ^2 and Mutual Information tests, yet it appeared highly significant in the full-sequence analysis for ANOVA and Mutual Information. Consequently, we still included Raptor-X Property in our ensemble model to see if adding a lower-performing prediction method with minimal feature importance could nevertheless improve performance.

The feature selection process identified several key predictors that significantly influence the ensemble model's performance. Unsurprisingly, their significance generally aligns with their performance scores, making the top-performing predictors the most important. Meanwhile, Mutual Information-based feature selection diverged considerably when using windowed sequence lengths, and ANOVA also varied with changes in data representation. Interestingly, when Mutual Information was excluded from the nominal data approach, the results mirrored the windowed nominal data findings, with top-performing predictors emerging as the most significant and low-performing predictors as the least significant.

5.3.2. Machine learning algorithms

We compared the performance of tree-type models and neural-type models in predicting secondary structure. Details of the tree-type models and neural-type models are given in Supplementary section A.13. Results can be seen in Fig. 5.3. The tree-type models, such as decision trees and random forests demonstrated superior performance with higher average SOV_REFINE scores and narrower confidence intervals, indicating more consistent predictions. Specifically, the tree-type models achieved an average SOV_REFINE test score of 95%. In contrast, the neural-type models, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), showed lower average SOV_REFINE scores and wider confidence intervals, reflecting greater variability in their predictions. The best neural-type model (transformer) achieved a similar average SOV_REFINE score of 93% to tree-type models. While neural-type models have the potential for improvement through extensive hyper-parameter tuning and improved architectural designs, tree-type models outperform them in terms of both accuracy and consistency for this limited mutational dataset.

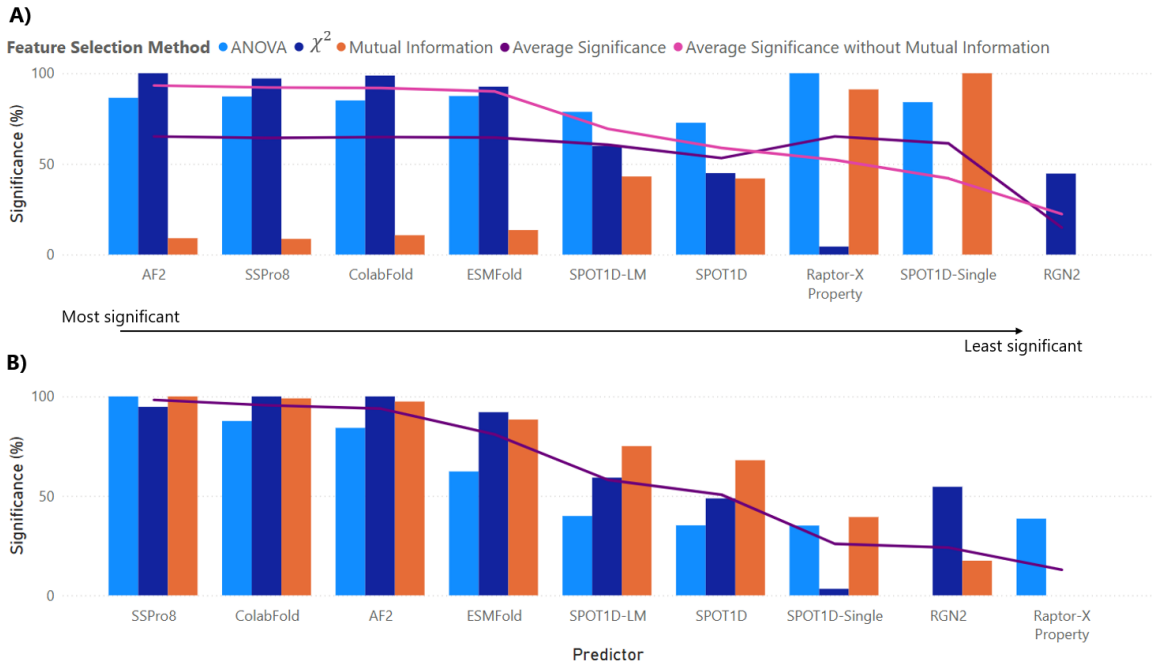


Figure 5.2.|: **Feature selection.** Score percentage for three differing feature selection algorithms: ANOVA, χ^2, and Mutual Information. Both graphs show the most significant predictors from left to the least significant predictors on the right. A) Results from nominal data. The purple line shows the average significance for all three algorithms for a specific predictor. The pink line shows the average for ANOVA and χ^2. Removing Mutual Information gives similar results for both windowed and full sequence nominal data where top, avg, and low performing methods follow the same trend to their significance. B) Results obtained from windowed nominal data. The line shows the average significance for all three algorithms for a specific predictor.

5.3.3. Tree ensemble results

As previously demonstrated, tree-type models showed superior performance on our mutational dataset. We therefore focused on tree-type algorithms — specifically extremely randomized trees — because of their robustness to overfitting. To assess the reliability and generalizability of these models, we performed a 7-fold cross-validation on the mutational dataset using a leave-one-out approach; the number seven was chosen to ensure folds were of uniform size. The results from this cross-validation are shown in Fig. 5.4.

We excluded tree-type algorithms that relied on a nominal data representation because they underperformed, possibly due to the discrepancy observed in the Mutual Information-based feature selection. It appears that absolute positional information imposed by nominal data does not yield meaningful insights into secondary structure. Consequently, all subsequent tree-type results presented here use a windowed nominal data representation.

An extremely randomized tree ensemble demonstrated consistent performance across all folds, achieving an average accuracy of 98% across all predictors when using longer

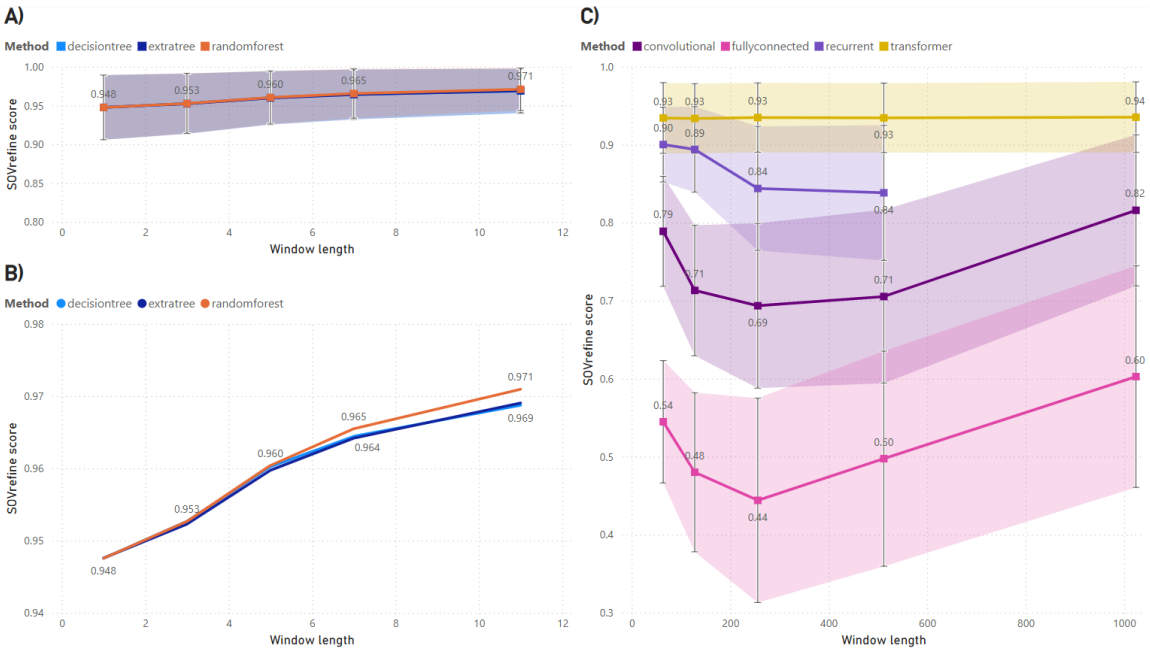


Figure 5.3.|: **Comparison of Tree-type and Neural-type trained models.** A) Results from tree-type models showing their average SOV_REFINE score and their confidence intervals using the 25th and 75th percentiles. B) A magnified look into the tree-type models for differing window lengths. C) Results from neural-type models, showing their average SOV_REFINE scores and confidence intervals with 33rd and 66th percentiles to reduce the interval overlap in the visual. Although tighter percentiles are used in neural-type models, confidence intervals are wider than tree-type models. Clearly, tree-type models outperform network-type models for this dataset. Further improvements to neural-type models should be possible but would require large amounts of hyper-parameter tuning and design considerations. It is clear that window length has an effect in the performance of the models. The recurrent model was not trained on the highest window length for memory limitations.

window lengths. Interestingly, as predictors improve in performance and gain higher feature significance, shorter window lengths can sometimes help avoid misclassifications of secondary structures. It is worth noting that overall accuracy tends to be high because the mutational dataset consists of proteins differing by only a single amino acid.

5.3.4. Mut2Dens

Here we describe the trained model Mut2Dens, which is built upon the knowledge gained during feature selection and model determination. As the feature selection algorithms suggest, we chose the most significant predictors, with an average significance percentage of 75 or higher, as the input for Mut2Dens. We also added Raptor-X Property for its ambiguous importance during feature selection. Its low significance on the windowed representation

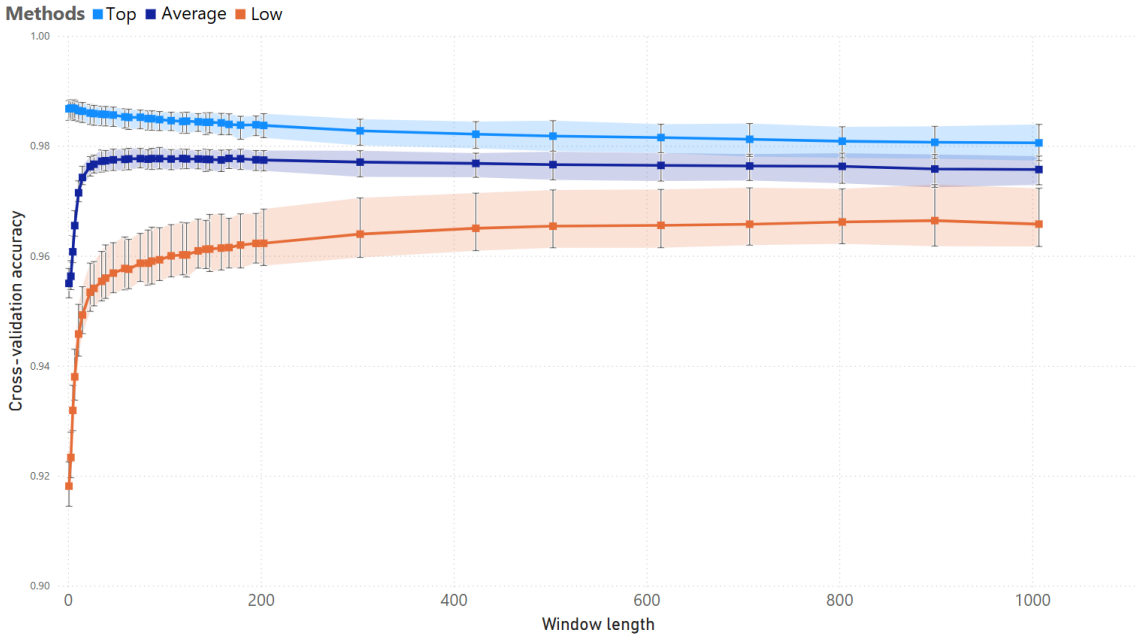


Figure 5.4.|: **Cross-validation results.** Further tree-type results using 7-fold cross validation with longer window lengths. The results are given for different input predictors: Top-performing, Average-performing, and Low-performing. Models created from Top-performing predictors show a slight decline in performance as window length increases, while the others improve as the window length increases.

can give insights into ensemble results from adding less significant predictors. We opt for ColabFold in place of AlphaFold2, as ColabFold is a computationally efficient version of AlphaFold2 while maintaining a high correlation between their predictions. Therefore, Mut2Dens is tailored to be computationally efficient with its mixed use of fast and accurate predictors. Further details of Mut2Dens are given in Supplementary section A.11. The final list of predictors are as follows,

1. SSPro8
2. ColabFold
3. ESMFold
4. Raptor-X Property

The predictor outputs are converted into nominal windowed data and concatenated into a single input vector. This input vector is passed to the ExtraTree model, which returns the refined output one amino acid at a time. Finally, the refined outputs are concatenated to form the complete refined secondary structure for the given sequence. A depiction of this process is shown in Fig 5.5.

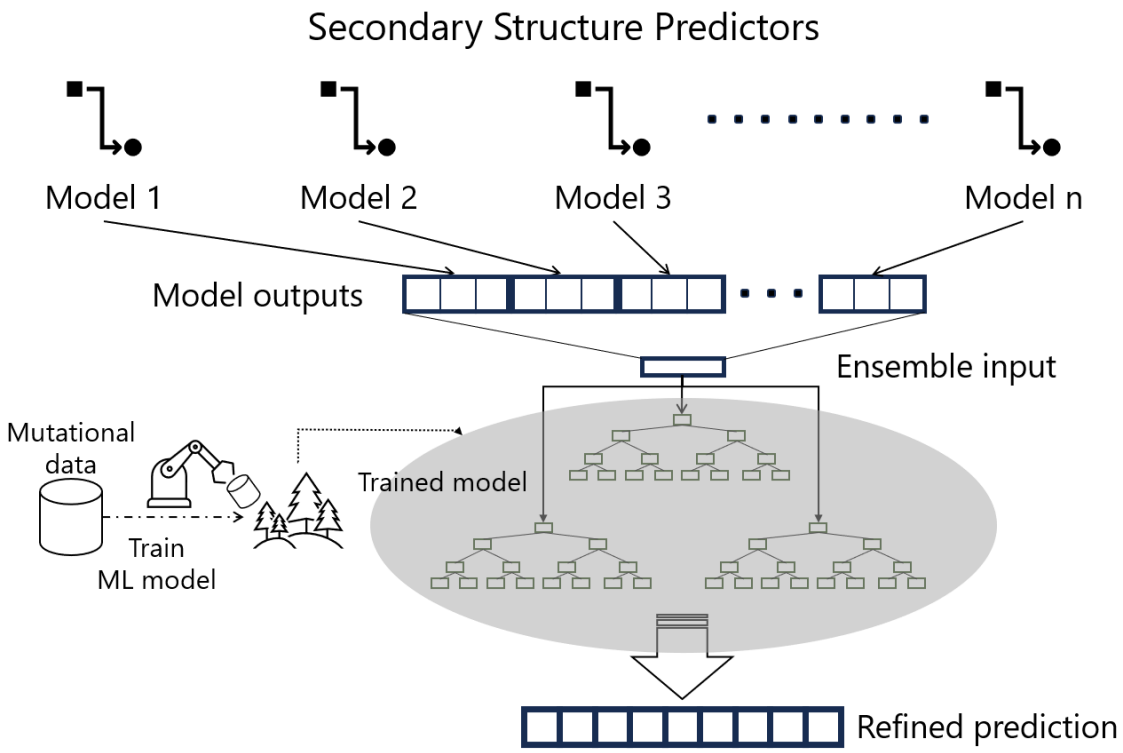


Figure 5.5.|: **Refinement strategy.** Diagram depicting the creation process of an ensemble model using our refinement strategy. First, the selected predictors are used to predict secondary structure for a given protein sequence. The predictors' outputs are concatenated and used as input for a trained tree-ensemble model of extremely randomized trees. The trained model, Mut2Dens, outputs a refined prediction of the secondary structure, which takes into account its mutation-specific training.

5.3.5. Mutational data results

Mutational capabilities of the models were evaluated using the mutational dataset along the mutational measures described previously. The mutational precision and accuracy results, depicted in Fig. 5.6 A and B respectively, show that most models consistently achieved high scores across all measures. These high results arise from the high degree of overlap between the mutational dataset and the training of these models. To minimize any potential overlap of the results and training data for Mut2Dens, results are shown for its testing dataset. While their mean results are high, the extreme values indicate a wide spread where low-confidence predicted proteins result in low predictive performance for most models. Our refinement strategy manages to reduce this spread, indicating the value of mutational refinement.

Mutational consistency converts the secondary structure classification problem into a two-class mutational classification task. For each amino acid location in the protein sequence, we ask whether the mutation caused a structural change or the structure remained stable. Fig. 5.6 C shows a high false negative rate for all predictors, including its majority

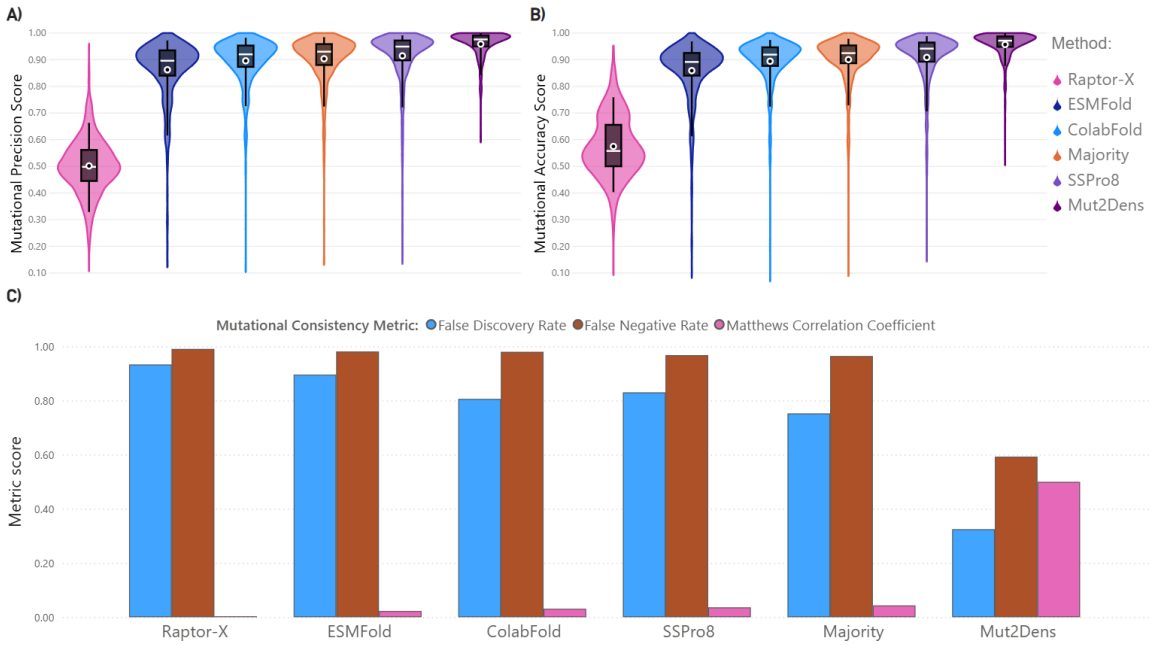


Figure 5.6.|: **Mutational dataset results.** Graphs showing mutational measure results for predictors, a majority agreement model, and Mut2Dens. Results show a narrower spread in the performance distribution, reducing the number of highly incorrect predictions for Mut2Dens for the following measures: A) Mutational precision, and B) Mutational accuracy. C) Results for mutational consistency measures: False Discovery Rate (FDR), False Negative Rate (FNR), and Matthews Correlation Coefficient (MCC). Mutational consistency scores indicate whether the structural mutation occurs in the correct place. High values of FDR and FNR indicate poor performance in the model predicting the correct structural change location. MCC indicates the overall performance of the model, where higher is better.

consensus model. The false negative rate in this context denotes an incorrect prediction of a change when no such structural change occurs. Likewise, the high false discovery rates indicate that out of all structural changes predicted, most structural changes did not actually occur. Mut2Dens produces lower false discovery rates and false negative rates than the predictors, demonstrating the usefulness of mutational refinement. The low Matthews Correlation coefficient scores indicate low predictive performance of the model regarding the mutational classification task. We can clearly see that our refinement strategy can enhance the overall mutational predictive capabilities of predictors, in contrast to a simple combination like a majority agreement of predictors or the predictors themselves.

5.3.6. Non-mutational data benchmarks

Mut2Dens was evaluated and compared to secondary structure predictors and a majority agreement model of the predictors. We also include an ablation study of the selected predictors for Mut2Dens that contained the best predictors for the test datasets, ColabFold

and SSPro8. Therefore, multiple versions of Mut2Dens with different input predictors are shown below.

Performance of the Mut2Dens model was evaluated using the CB513 dataset, a widely recognized benchmark for secondary structure prediction for its non-homologous proteins, with a sequence similarity of less than 25% between all proteins. The results, depicted in Fig. 5.7, demonstrate that Mut2Dens achieved high SOV_REFINE scores, indicating its ability to predict secondary structures accurately. Specifically, the model outperformed all other predictors, including SSPro8, in terms of extreme value predictions, as shown by the XTSpread measure. When looking at standard (near the mean) protein predictions, Mut2Dens exhibited a small decrease of performance of less than 1% compared to SSPro8, as indicated by the STSpread measure. Overall, the CB513 test benchmark results highlight the robustness and effectiveness of Mut2Dens in handling a diverse set of proteins. The capabilities of Mut2Dens are comparable to the best predictor for the dataset, while increasing the performance of the most inaccurate predictions for other predictors.

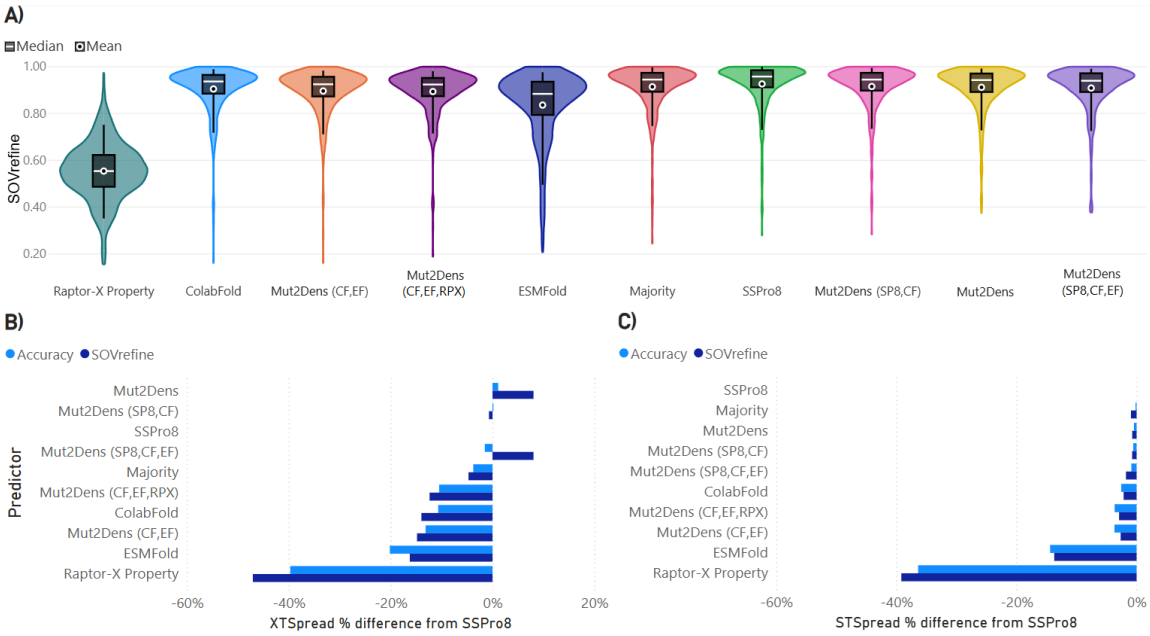


Figure 5.7.|: **Testing models on CB513.** This dataset has been previously utilized as a testing benchmark for many studies. Predictors utilized for ensemble models include ColabFold (CF), SSPro8 (SP8), ESMFold (EF), and Raptor-X Property (RPX). A) SOV_REFINE score results. The high scores result from most models utilizing this dataset. B) XTSpread difference to the best performing non-ensemble predictor for this dataset, SSPro8. C) STSpread difference to SSPro8. Taking the extreme values into account with XTSpread, we can see our ensemble model is capable of outperforming all others. Conversely, our ensemble model has a slightly lower performance than SSPro8 when focusing on non-extreme (very low performing) proteins.

From the different versions of Mut2Dens, we can see that the addition of predictors with low significance do increase the outcome slightly. It is likely that additional inclusion of uncorrelated predictors with differing methodologies will increase the outcome, although

marginally. Therefore, the inclusion of such predictors will depend on a cost-effectiveness evaluation as each additional predictor increases processing time in a non-trivial manner.

The performance of Mut2Dens was also evaluated using the CASP15 dataset. The results for this benchmark dataset are shown in Fig. 5.8. Mut2Dens predictive capabilities increased the extreme values by a wide margin of almost 30%. This increase in low-performing predicted proteins come at a slight cost where the STSpread accuracy score of Mut2Dens decreases by about 5%. Most inaccurate predictions by secondary structure predictors have low confidence values. Mut2Dens is suggested to be utilized for these instances as it can increase the quality of the predicted structures for such proteins and simultaneously provide insights from multiple predictors.

Figure 5.8.|: **Testing models on CASP15.** Most recent dataset with proteins that have not been included in the training of any model. Predictors utilized for ensemble models include ColabFold (CF), SSPro8 (SP8), ESMFold (EF), and Raptor-X Property (RPX). A) Performance of the models is more realistic than CB513 with a maximum mean SOV_REFINE of 88% by ColabFold. B) XTSpread and C) STSpread difference to the best performing non-ensemble model for this dataset, ColabFold. Similarly to CB513, the ensemble models outperform others when extreme values are taken into consideration, but perform slightly lower for non-extreme values.

5.3.7. Knowledge-based model

Examining how tree-based models use our mutational data to refine secondary structure shows that their decisions align with findings from chapter 4, indicating that single amino acid mutations rarely convert helices into sheets (or vice versa). This insight is drawn primarily from crystallographic data, which necessitates that proteins form ordered crystal

structures. Consequently, the data may be incomplete, yet the frequency of secondary structural interchanges observed in predictions remains higher than what current experimental evidence suggests.

Through the training procedure of tree-type models, we can compute a confusion matrix of the resulting SSE classes separated by every final decision (leaf) within the tree. This allow us to see the amount of weight any SSE class has when the model is expecting to predict a certain SSE. Our tree-type models contain over 1000 such prediction decisions and as such we aggregate them to obtain a complete picture of the weight each SSE class gives for an expected outcome. Therefore, we obtained decisions from 10 tree-type models to obtain a generalized average weight for each SSE class during an expected decision outcome. We use a simplified ensemble model with a window length of 1 to isolate the decisions taken for each model at a specific amino acid. Analyzing these weights, we obtain the results shown in Table 5.1. This square matrix contains possible secondary structure classes, where rows contain the expected or 'true' SSE and columns contain the SSE classes decided by the tree-type model. These values can be normalized to obtain the probability of the tree deciding a specific SSE class for each of the assigned SSE classes. The diagonal represents the secondary classes that are correctly decided, while the non-diagonal values represent overlapping decisions with other classes. We can see that π-helices have the most confounding values with other classes as it is selected 84.4% when it is expected. Interestingly, π-helices are never confounded with β-sheets, isolated β-bridges, or coils. The model is able to predict a π-helix for an amino acid only when it is certain that β-sheets, isolated β-bridges, or coils are not feasible for that amino acid. This follows the knowledge-based rule that π-helix never turn into β-sheets, bridges, $3-10$ helices, or coils within our mutational data.

Likewise, we can obtain all decision thresholds within the tree created during their training process. Aggregating these decisions thresholds allows us to obtain meaning from the decisions that the tree uses to obtain a certain outcome. The resulting human-readable rules, which simplifies the complexities of the tree-like structure, resemble the following example wording: "When ColabFold and ESMFold predict an α-helix, and SSpro8 predicts a π-helix, the ensemble outcome is π-helix."

The overall influence that a feature, the secondary structure outcome for a certain amino acid from a prediction method, has in Mut2Dens refinement outcomes can be obtained through the amount of reduction in Gini impurity achieved by the splitting of nodes for that particular feature. This procedure can be done throughout the training of the model. Otherwise, obtaining the importance of a prediction method for a particular protein is possible by back-tracing the decision nodes for the outcome. For multiple trees, these decisions can be averaged to obtain influential features that are common to most trees.

The advantage of using an ensemble model is the inclusion of multiple predictor outputs. Having different predictions allow us to compare and check which predictors have generated incorrect SSEs. We can also take these rules to see where the model generally thinks certain predictors require an adjustment. Furthermore, each individual protein prediction can also be compared to each predictor's outcome for a more detailed view of the

protein. Utilizing visualization tools such as 2dss (http://genome.lcqb.upmc.fr/2dss/), although limited to Q3 visuals, we can more easily see potential weaknesses of each predictor and have a better understanding of the resulting prediction from our ensemble model. An example of such a visualization is given in Fig. 5.9 comparing the DSSP-assigned structure, and predicted structures from Mut2Dens and the highly accurate ColabFold predictor. The numbers in the figure represent the location in the amino acid sequence, while the colors for the query describe the properties of each amino acid.

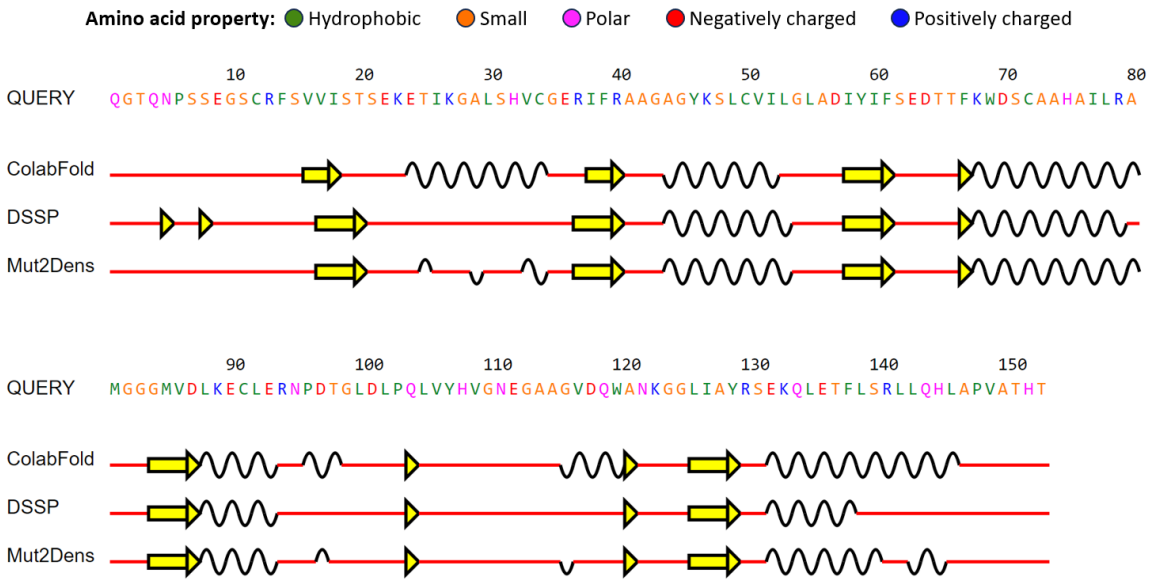


Figure 5.9.|: **Structure refinement comparison.** Visualization of predicted and assigned secondary structures. The secondary structure is simplified into Q3 for visualization purposes. For each amino acid, a line represents a coil, the yellow arrow represents a β-sheet, and the wavy line represents an α-helix. For this protein, ColabFold achieves 61% accuracy, while Mut2Dens achieves 82%. ColabFold predicts α-helices in several places that do not occur in the actual assigned structure by DSSP. While not perfect, Mut2Dens tries to correct these structures by removing most of the helical SSEs that are not part of the actual structure.

DSSP-assigned	Tree-decided							
SSE class	**Isolated β-bridge**	**Coil**	**β-sheet**	**$3-10$ helix**	**α-helix**	**π-helix**	**Bend**	**Turn**
Isolated β-bridge	2622	132	102	1	1	0	39	8
Coil	128	53743	777	98	90	2	1259	442
β-sheet	305	1861	68521	8	1	0	291	46
$3-10$ helix	0	109	7	10076	298	1	218	770
α-helix	2	202	18	377	85050	166	313	1610
π-helix	0	0	0	7	239	1768	11	71
Bend	53	1221	208	99	73	11	23210	674
Turn	15	313	32	606	859	22	1405	29978

Table 5.1.: **Tree knowledge.** Decision overlap from simplified tree-type models. It is clearly shown that π-helices only overlap for helices and bends and turns. This coincides with rules obtained from our previous study, where π-helices do not transition into β-sheets, β-bridges, or coils. Other rules also become evident, such as β-sheets not transitioning into α-helices or π-helices.

5.4. Conclusions

Although recent advances have led to highly accurate structure prediction methods, incorrect or low-confidence predictions are still possible — particularly when homology data are scarce. In such cases, these models may yield unreliable topologies. Furthermore, previous findings in chapter 4 show that current prediction methods often fail to predict the structural effects of single amino acid mutations accurately, sometimes producing highly improbable outcomes compared to experimental data.

To address these limitations, we developed a novel refinement strategy that relies solely on single amino acid mutational data, requiring no additional information. By creating an ensemble of methodologically diverse predictors, we mitigate individual weaknesses while achieving reliable and consistent mutation-focused predictions. Our approach, implemented in the Mut2Dens model, integrates the strengths of selected predictors, thereby improving performance on mutational datasets without compromising results on non-mutational datasets.

Moreover, our refinement strategy enables the exploration or validation of biological insights through interpretable machine learning algorithms, such as tree-based models. Future work could extend these findings to tertiary structure predictions by leveraging knowledge gained from inconsistencies observed in secondary structure predictions. Additionally, we include the use of visualization tools that highlight discrepancies among predictors, revealing which secondary structures have been modified during refinement—thereby guiding researchers to regions where the predicted protein structure may be unstable or unreliable.

It is important to note that this refinement strategy is not intended to replace current prediction models. Instead, it serves as an auxiliary tool to verify and enhance the integrity of structure predictions, especially when predictor confidence is low. Incorporating mutational data not only refines existing models but also improves low-scoring predictions for non-mutational data, addressing a significant shortcoming of current prediction methods. For instance, in cases where ColabFold produces low-performing predictions, Mut2Dens often yields more reliable secondary structure representations. The ability to improve low-confidence predictions also allows meaningful comparisons across various structure prediction methods.

Ultimately, our work narrows the gap in low-scoring protein structure predictions, offering a more accurate and reliable framework that benefits biomedical research and drug design. By enhancing the consistency of protein structure predictions, we provide researchers with a comprehensive toolkit for making informed decisions about predicted structures. Moreover, secondary structure-focused applications, such as protein functional prediction methods [Song et al., 2024], can directly benefit from these more reliable secondary structure predictions.

Part III

Conclusion

Chapter 6

Conclusions

Contents

In this chapter, we present a dissertation summary by revisiting the research challenges and goals we addressed, as well as the corresponding contributions. Afterwards, this chapter concludes this dissertation with a discussion of future research opportunities.

6.1. Dissertation Summary

In this dissertation, we have addressed and solved several challenges in protein structure prediction. We focus on the underexplored area of single amino acid mutations, since it did not have much experimental data available. Although still limited, given the increased availability of current mutational data, we assess the performance of state-of-the-art structure prediction methods on such data. Our findings indicate that single amino acid mutations can significantly affect the accuracy of structure prediction methods.

The research goal addressed in this dissertation was to contribute on the improvement to protein structure prediction capabilities. To this end, we propose a novel refinement strategy for protein secondary structure prediction that leverages single amino acid mutational data. As part of this strategy, we introduce Mut2Dens, a model that not only yields more consistent predictions for mutational data but also maintains robust predictive performance on non-mutational datasets.

6.1.1. Addressed Challenges and Goals

We constrained the research problem by stating a set of research challenges. In the following, we summarized how we addressed the challenges stated for this dissertation.

CH1: *There is a limited amount of mutational structure data for proteins.* Through publicly available databases, we created a dataset comprising 541 groups of mutated proteins. Although this subset of protein conformations is limited, significant differences between predicted and experimental data can be obtained. This can lead to improvements in current prediction methods for, at the very least, a subset of proteins.

CH2: *Crystallographic data might not represent the breadth of mutated structures.* While the public structural data obtained for our research is mostly created through crystallographic means, the data applies to a subset of protein conformations. Therefore, this data does not represent a complete selection of protein conformations, but state-of-the-art prediction methods should be capable of correctly predicting such a subset. Failure to predict this data subset implies a need for improvement in the prediction methods.

CH3: *Simplification of protein structure into solely secondary structure might not capture mutational effects on the protein.* While tertiary structure gives more detail into the protein shape, secondary structure simplifies the prediction task of the state-of-the-art prediction methods. As with the previous challenges, failure to correctly predict secondary structure changes due to single amino acid mutations indicate opportunities for improvement.

CH4: *Improvement on secondary structure prediction might be unfeasible.* While secondary structure and tertiary structure prediction methods have been deemed sufficiently robust for their prediction tasks, their capabilities are not perfect for every protein and conformation possible. Improvement can be performed on the proteins where performance is still not sufficiently accurate.

In light of these challenges, we pursued the general goal of assessing state-of-the-art protein structure prediction methods and improving their performance from deficiencies in their mutational capabilities. We refined this goal by adopting four specific goals, as follows:

Assessment

G1: Obtain a mutational dataset with a wide breadth of mutational effects to evaluate state-of-the-art prediction methods.

G2: Evaluate the performance of structure prediction methods on basic mutational conditions to streamline the analysis process confounding prediction artifacts.

Improvement

G3: Analyse and investigate deficiencies on evaluated prediction methods from the mutational dataset.

G4: Propose a strategy and a model implementation of this strategy to improve current state-of-the-art prediction methods capabilities on single amino acid mutation data.

In the following section, we discuss our achieved contributions with respect to these goals and challenges.

6.1.2. Contributions

We present our main contributions according to our single amino acid mutation conceptualization, where simplified conditions are given to provide optimal conditions for the prediction of mutational effects on protein structure. We summarize these contributions, as follows:

C1: Addressing G1 Obtain a mutational dataset of experimental data containing primary, secondary, and tertiary protein structures and their respective mutational changes to evaluate prediction methods.

C2: Addressing G2 and G3 Benchmarking a set of diverse state-of-the-art structural prediction methods on single amino acid mutation data to provide insights into their prediction deficiencies

C3: Addressing G3 Proposing a novel secondary structure refinement strategy that relies solely on single amino acid mutational data, and implementing this strategy through a model that achieves reliable and consistent mutation-focused predictions.

C4: Addressing G3 and G4 Evaluation of models using our refinement strategy on mutational and non-mutational protein datasets to assure consistent predictions across both types of protein data.

6.1.3. Contributions Significance

Throughout the previous decade, protein structure prediction research has been focusing on tertiary structure. Prior to that, secondary structure prediction performance had steadily been improving, reaching a plateau at the start of the 21^{st} century. Now tertiary structure prediction has reached performance levels similar to secondary structure. Still, secondary structure is useful for the refinement of protein tertiary structures as their prediction methods are not perfect. It is not uncommon to find protein regions where tertiary structure prediction methods have low confidence, resulting in incorrect structural predictions. We address this issue by investigating possible deficiencies in their methodology. An area that is resurging due to increases in data availability is the prediction of structural effects from single amino acid mutations. While data is still limited, testing the capabilities of current state-of-the-art prediction methods on mutational data is possible. Our assessment has lead to the discovery of deficiencies in mutational capabilities of these prediction methods. We reaffirm these deficiencies by proposing and implementing a refinement strategy for these mutational deficiencies, which leads to an increase in accuracy for incorrectly predicted protein structures from current methods.

Structure prediction research must contend with mutational changes to proteins. Current personalized medicine will require understanding of these structure mutational changes to produce improved outcomes in treatment for individuals through their unique genome and mutations. As we have demonstrated, all structural data is essential for the success of this endeavor, thus being able to predict protein secondary structures is crucial for this field.

6.2. Future Work

This dissertation concludes with a presentation of selected future work opportunities emerging from our research.

6.2.1. Protein structure synergistic integration

Significant advances have been achieved in recent times for protein tertiary structure prediction. The performance of these predictions methods is not perfect and can exhibit inconsistencies in low-confidence protein regions. As we have shown in this dissertation, utilizing secondary structure prediction methodologies, in conjunction with tertiary structure prediction methods, can increase the performance of incorrectly predicted structures. Further work in bringing secondary and tertiary structure prediction methods into a synergistic method should and can be achieved with the technological advances currently available.

The creation of a deep learning model that predicts both secondary structure and tertiary structure can bring such a synergistic prediction. The secondary structure prediction component can directly transfer secondary structure knowledge into the tertiary structure component. As AlphaFold2 does through recycling, the tertiary structure can also help stabilize and correct the secondary structure prediction, forming a synergistic loop of different structure hierarchical levels.

The stability that secondary structure provides for the protein structure through backbone bonds helps contrast the noisy environment from atomic locations. Current leading tertiary structure prediction methods, such as AlphaFold2, depend on homology and protein templates to output protein structures of excellent quality. Research is currently ongoing for template-less prediction methods without homology integration. These methods are highly desired, but their performance remains lower than homology- and template-based methods.

The addition of a highly performing and robust secondary structure prediction component in combination with co-evolutionary information could give an indication of structural homology that the tertiary structure prediction component can utilize to refine its structure. Likewise, the geometrical constraints of the backbone angles from the tertiary structure could increase the performance of the secondary structure component.

An additional benefit that such synergistic integration might offer is the ability to predict tertiary structure given a primary and secondary structure of a protein. Imposing secondary structure restraints to the protein might be important for a protein's functionality. Furthermore, specifically trained models with primary, secondary, and tertiary structure knowledge might be able to translate between the different structural hierarchies, leading to more flexibility in the design process of a protein.

6.2.2. Protein function

Protein function is inferred through its structure. Prediction methods for functional classification can benefit from secondary structure as shown in work by Song et al. [Song et al., 2024]. Utilizing a synergistic approach as previously mentioned, functional prediction could be improved. Secondary structure in conjunction with tertiary structure can be utilized to predict a protein's function. We believe the usage of more robust secondary structure prediction can also benefit protein function prediction. This is especially the case as function is directly correlated to the protein tertiary structure.

A model which employs secondary and functional constraints to predict tertiary structure might bring protein design improvements by allowing the translation of different structure hierarchies into protein functionality. We envision this type of model to require more data than is currently available. Therefore, additional data acquisition might be required to create a general protein design model that utilizes primary, secondary and tertiary structure, alongside functionality constraints of a protein.

6.2.3. Personalized medicine

As mentioned previously, personalized medicine can greatly benefit from the advances incurred from progress in structure prediction. Secondary structure prediction methods can benefit this field as it can directly benefit protein structure prediction as shown in this dissertation. Mutational changes to the proteome of an individual can be identified and taken into consideration utilizing the previously mentioned hierarchical structure synergistic approach. Deciphering the structural changes that have occurred due to mutations will help knowing which molecules or drugs can bind effectively to the target protein. Such a synergistic approach can also help in discovering possible side effects to drugs from undesired interactions throughout the body.

Ultimately, protein function is not only dependent on the protein structure, but also on the protein's environment. Newer models that take into consideration biomolecular interactions and multimeric proteins, composed of multiple chain subunits, are being developed [Abramson et al., 2024]. We believe that the previously mentioned synergistic approach in these models might lead to improved drug design by allowing nuanced changes to a protein's functionality. Therefore, future models should also consider functional changes that differ from complete activation or deactivation of the protein.

Acronyms

2D VICINITY	Secondary structure vicinity
3D	Three-dimensional
3D VICINITY	Tertiary structure vicinity
AA	Amino acid
AFP	Aligned fragment pair
ANOVA	Analysis of variance
BRNN	Bidirectional recursive neural network
CASP	Critical assessment of techniques for protein structure prediction
CATH	Class, architecture, topology, homologous superfamily
CNN	Convolutional neural network
CRYO-EM	Cryogenic-electron microscopy
DNA	Deoxyribonucleic acid
DP	Dynamic programming
DSSP	Dictionary of Secondary Structure in Proteins
EBI	European bioinformatics institute
ECOD	Evolutionary Classification of protein Domains
EMBL	European molecular biology laboratory
EXTRATREE	Extremely randomized trees
FCNN	Fully-connected neural network
GDT	Global distance test
GNN	Graph neural network
GO	Gene ontology

HMM	Hidden Markov model
LCS	Longest continuous segment
MACHINE LEARNING	Machine learning
MMCIF	Macromolecular Crystallographic Information File
MRNA	Messenger RNA
MSA	Multiple sequence alignment
NCBI	National center for biotechnology information
NGS	Next generation sequencing
NIH	National institute of health
NMR	Nuclear magnetic resonance
NS-SNP	non-synonymous SNP
PDB	Protein data bank
PDBX	PDB Exchange
PSSM	Position specific scoring matrix
Q3	Three-state secondary structure
Q8	Eight-state secondary structure
RCSB	Research Collaboratory for Structural Bioinformatics
RNA	Ribonucleic acid
RNN	Recurrent neural network
SCOP	Structural classification of proteins
SNP	Single nucleotide polymorphism
SSE	Secondary structure element
TNN	Transformer neural network
TRNA	Transfer RNA

Glossary

Å	Angstrom
$\mathbf{A} = \{A, C, D, ..., V, W, Y\}$	Set of 20 standard amino acids
$\mathbf{S} = [a_1, ..., a_n]$, $a_i \in \mathbf{A}$	Protein sequence over A_{20}, a_1 is N-terminus, a_n is C-terminus
$\mathbf{S}_{ij} = [a_i, a_{i+1}, ..., a_j]$	Subsequence of $\mathbf{S}$ for $1 \leq i \leq j \leq n$ of length $L = j - i + 1$
$p_i^{C\alpha} = (p_{i_x}^{C\alpha}, p_{i_y}^{C\alpha}, p_{i_z}^{C\alpha})$	Coordinates of $C\alpha$ atom corresponding to AA a_i
$p_i^{C\beta} = (p_{i_x}^{C\beta}, p_{i_y}^{C\beta}, p_{i_z}^{C\beta})$	Coordinates of $C\beta$ atom corresponding to AA a_i
$\mathbf{P} = [p_1^{C\alpha}, p_1^{C\beta}, ..., p_n^{C\alpha}, p_n^{C\beta}]$	3D backbone structure for $\mathbf{S}$
$\mathbf{P}^{C\alpha} = [p_1^{C\alpha}, ..., p_n^{C\alpha}]$	3D $C\alpha$ structure for $\mathbf{S}$
$\mathbf{P}^{C\beta} = [p_1^{C\beta}, ..., p_n^{C\beta}]$	3D $C\beta$ structure for $\mathbf{S}$
$\mathbf{P}_{ij} = [p_i^{C\alpha}, p_i^{C\beta}, ..., p_j^{C\alpha}, p_j^{C\beta}]$	3D backbone structure for $\mathbf{S}_{ij}$
$\mathbf{P}_{ij}^{C\alpha} = [p_i^{C\alpha}, ..., p_j^{C\alpha}]$	3D $C\alpha$ structure for $\mathbf{S}_{ij}$
$\mathbf{P}_{ij}^{C\beta} = [p_i^{C\beta}, ..., p_j^{C\beta}]$	3D $C\beta$ structure for $\mathbf{S}_{ij}$
$\Upsilon_8 = \{\mathcal{C}, \mathcal{H}, \mathcal{E}, \mathcal{G}, \mathcal{I}, \mathcal{T}, \mathcal{S}, \mathcal{B}\}$	Set of eight-state secondary structure elements assigned by DSSP
$\mathbf{R} = [r_1, ..., r_n]$, $r_i \in \Upsilon_8$	Secondary structure sequence assigned to $\mathbf{S}$ over Υ_8
$\mathbf{R}_{ij} = [r_i, r_{i+1}, ..., r_j]$	Secondary structure subsequence assigned to $\mathbf{S}_{ij}$
$\hat{\mathbf{S}}_{ij} = [a_i, ..., \hat{a}_m, ..., a_j]$	Sequence corresponding to a *single amino acid mutation* for $\mathbf{S}_{ij}$
$\hat{\mathbf{P}}_{ij} = [\hat{p}_i^{C\alpha}, \hat{p}_i^{C\beta}, ..., \hat{p}_j^{C\alpha}, \hat{p}_j^{C\beta}]$	3D backbone structure corresponding to $\hat{\mathbf{S}}_{ij}$
$\hat{\mathbf{R}}_{ij} = [\hat{r}_i, ..., \hat{r}_j]$	Secondary structure subsequence assigned to $\hat{\mathbf{S}}_{ij}$

Bibliography

[Abadi et al., 2016] Abadi, M., Barham, P., Chen, J., et al. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283.

[Abramson et al., 2024] Abramson, J., Adler, J., Dunger, J., et al. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, pages 1–3. Publisher: Nature Publishing Group.

[Adams et al., 2019] Adams, P. D., Afonine, P. V., Baskaran, K., et al. (2019). Announcing mandatory submission of PDBx/mmCIF format files for crystallographic depositions to the Protein Data Bank (PDB). *Acta Crystallographica. Section D, Structural Biology*, 75(Pt 4):451–454.

[Ahdritz et al., 2024] Ahdritz, G., Bouatta, N., Floristean, C., et al. (2024). OpenFold: retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature Methods*, 21(8):1514–1524. Publisher: Nature Publishing Group.

[Alford et al., 2017] Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., et al. (2017). The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048. Publisher: American Chemical Society.

[Aloy et al., 2003] Aloy, P., Stark, A., Hadley, C., and Russell, R. B. (2003). Predictions without templates: New folds, secondary structure, and contacts in CASP5. *Proteins: Structure, Function, and Bioinformatics*, 53(S6):436–456. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.10546.

[AlQuraishi, 2019] AlQuraishi, M. (2019). End-to-End Differentiable Learning of Protein Structure. *Cell Systems*, 8(4):292–301.e3.

[Andreeva et al., 2020] Andreeva, A., Kulesha, E., Gough, J., and Murzin, A. G. (2020). The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic Acids Research*, 48(D1):D376–D382.

[Antony et al., 2021] Antony, J. V., Madhu, P., Balakrishnan, J. P., and Yadav, H. (2021). Assigning secondary structure in proteins using AI. *Journal of Molecular Modeling*, 27(9):252.

[Atkins and Gesteland, 2002] Atkins, J. F. and Gesteland, R. (2002). The 22nd Amino Acid. *Science*, 296(5572):1409–1410. Publisher: American Association for the Advancement of Science.

[Auton et al., 2015] Auton, A., Abecasis, G. R., Altshuler, D. M., et al. (2015). A global reference for human genetic variation. *Nature*, 526(7571):68–74. Number: 7571 Publisher: Nature Publishing Group.

[Baek et al., 2021] Baek, M., DiMaio, F., Anishchenko, I., et al. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876. Publisher: American Association for the Advancement of Science.

[Bairoch and Apweiler, 1996] Bairoch, A. and Apweiler, R. (1996). The SWISS-PROT Protein Sequence Data Bank and Its New Supplement TREMBL. *Nucleic Acids Research*, 24(1):21–25.

[Bairoch and Boeckmann, 1994] Bairoch, A. and Boeckmann, B. (1994). The SWISS-PROT protein sequence data bank: current status. *Nucleic Acids Research*, 22(17):3578–3580.

[Baldwin and Rose, 1999a] Baldwin, R. L. and Rose, G. D. (1999a). Is protein folding hierarchic? I. Local structure and peptide folding. *Trends in Biochemical Sciences*, 24(1):26–33. Publisher: Elsevier.

[Baldwin and Rose, 1999b] Baldwin, R. L. and Rose, G. D. (1999b). Is protein folding hierarchic? II. Folding intermediates and transition states. *Trends in Biochemical Sciences*, 24(2):77–83. Publisher: Elsevier.

[Banfield et al., 2007] Banfield, R. E., Hall, L. O., Bowyer, K. W., and Kegelmeyer, W. (2007). A Comparison of Decision Tree Ensemble Creation Techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):173–180.

[Berkowitz et al., 1968] Berkowitz, D., Hushon, J. M., Whitfield, H. J., et al. (1968). Procedure for Identifying Nonsense Mutations. *Journal of Bacteriology*, 96(1):215–220. Publisher: American Society for Microbiology.

[Bernardi and Bruni, 2019] Bernardi, L. and Bruni, A. C. (2019). Mutations in Prion Protein Gene: Pathogenic Mechanisms in C-Terminal vs. N-Terminal Domain, a Review. *International Journal of Molecular Sciences*, 20(14):3606. Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.

[Bettella et al., 2012] Bettella, F., Rasinski, D., and Knapp, E. W. (2012). Protein Secondary Structure Prediction with SPARROW. *Journal of Chemical Information and Modeling*, 52(2):545–556. Publisher: American Chemical Society.

[Blout et al., 1960] Blout, E. R., de Lozé, C., Bloom, S. M., and Fasman, G. D. (1960). THE DEPENDENCE OF THE CONFORMATIONS OF SYNTHETIC POLYPEPTIDES ON AMINO ACID COMPOSITION1,2. *Journal of the American Chemical Society*, 82(14):3787–3789. Publisher: American Chemical Society.

[Bokor and Tantos, 2021] Bokor, M. and Tantos, A. (2021). Secondary Structures of Proteins: A Comparison of Models and Experimental Results. *Journal of Proteome Research*, 20(3):1802–1808. Publisher: American Chemical Society.

[Breiman, 2001] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

[Brinkjost et al., 2020] Brinkjost, T., Ehrt, C., Koch, O., and Mutzel, P. (2020). SCOT: Rethinking the classification of secondary structure elements. *Bioinformatics*, 36(8):2417–2428.

[Brooks et al., 2009] Brooks, B. R., Brooks III, C. L., Mackerell Jr., A. D., et al. (2009). CHARMM: The biomolecular simulation program. *Journal of Computational Chemistry*, 30(10):1545–1614. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.21287.

[Buchan and Jones, 2019] Buchan, D. W. A. and Jones, D. T. (2019). The PSIPRED Protein Analysis Workbench: 20 years on. *Nucleic Acids Research*, 47(W1):W402–W407.

[Burgess and Scheraga, 1975] Burgess, A. W. and Scheraga, H. A. (1975). Assessment of some problems associated with prediction of the three-dimensional structure of a protein from its amino-acid sequence. *Proceedings of the National Academy of Sciences*, 72(4):1221–1225.

[Burley et al., 2017] Burley, S. K., Berman, H. M., Kleywegt, G. J., et al. (2017). Protein Data Bank (PDB): The Single Global Macromolecular Structure Archive. In Wlodawer, A., Dauter, Z., and Jaskolski, M., editors, *Protein Crystallography: Methods and Protocols*, Methods in Molecular Biology, pages 627–641. Springer, New York, NY.

[Burman, 1990] Burman, P. (1990). Estimation of Optimal Transformations Using v-Fold Cross Validation and Repeated Learning-Testing Methods. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 52(3):314–345.

[Bustamante et al., 2011] Bustamante, C., Cheng, W., and Mejia, Y. X. (2011). Revisiting the Central Dogma One Molecule at a Time. *Cell*, 144(4):480–497. Publisher: Elsevier.

[Canbek et al., 2017] Canbek, G., Sagiroglu, S., Temizel, T. T., and Baykal, N. (2017). Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 821–826.

[Cao et al., 2016] Cao, C., Wang, G., Liu, A., et al. (2016). A New Secondary Structure Assignment Algorithm Using Cα Backbone Fragments. *International Journal of Molecular Sciences*, 17(3):333. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[Carter et al., 2003] Carter, P., Andersen, C. A. F., and Rost, B. (2003). DSSPcont: continuous secondary structure assignments for proteins. *Nucleic Acids Research*, 31(13):3293–3295.

[Castelvecchi, 2024] Castelvecchi, D. (2024). 'A truly remarkable breakthrough': Google's new quantum chip achieves accuracy milestone. *Nature*, 636(8043):527–528.

[Chandonia et al., 2019] Chandonia, J.-M., Fox, N. K., and Brenner, S. E. (2019). SCOPe: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic Acids Research*, 47(D1):D475–D481.

[Chebrek et al., 2014] Chebrek, R., Leonard, S., de Brevern, A. G., and Gelly, J.-C. (2014). PolyprOnline: polyproline helix II and secondary structure assignment database. *Database*, 2014:bau102.

[Chitty-Venkata et al., 2022] Chitty-Venkata, K. T., Emani, M., Vishwanath, V., and Somani, A. K. (2022). Neural Architecture Search for Transformers: A Survey. *IEEE Access*, 10:108374–108412. Conference Name: IEEE Access.

[Chou and Fasman, 1979] Chou, P. Y. and Fasman, G. D. (1979). Prediction of the secondary structure of proteins from their amino acid sequence. *Advances in enzymology and related areas of molecular biology*, 47:45–148. Publisher: Wiley Online Library.

[Chowdhury et al., 2022] Chowdhury, R., Bouatta, N., Biswas, S., et al. (2022). Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology*, 40(11):1617–1623. Number: 11 Publisher: Nature Publishing Group.

[Clarke et al., 1999] Clarke, J., Cota, E., Fowler, S. B., and Hamill, S. J. (1999). Folding studies of immunoglobulin-like β-sandwich proteins suggest that they share a common folding pathway. *Structure*, 7(9):1145–1153. Publisher: Elsevier.

[Colloc'h et al., 1993] Colloc'h, N., Etchebest, C., Thoreau, E., et al. (1993). Comparison of three algorithms for the assignment of secondary structure in proteins: the advantages of a consensus assignment. *Protein Engineering, Design and Selection*, 6(4):377–382.

[Coluccio et al., 2015] Coluccio, M. L., Gentile, F., Das, G., et al. (2015). Detection of single amino acid mutation in human breast cancer by disordered plasmonic self-similar chain. *Science Advances*, 1(8):e1500487. Publisher: American Association for the Advancement of Science.

[Conte et al., 1999] Conte, L. L., Chothia, C., and Janin, J. (1999). The atomic structure of protein-protein recognition sites1. *Journal of Molecular Biology*, 285(5):2177–2198.

[Cowan and McGAVIN, 1955] Cowan, P. M. and McGAVIN, S. (1955). Structure of Poly-L-Proline. *Nature*, 176(4480):501–503. Publisher: Nature Publishing Group.

[Crooks et al., 2004] Crooks, G. E., Hon, G., Chandonia, J.-M., and Brenner, S. E. (2004). WebLogo: A Sequence Logo Generator. *Genome Research*, 14(6):1188–1190. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab.

[Cubellis et al., 2005] Cubellis, M. V., Cailliez, F., and Lovell, S. C. (2005). Secondary structure assignment that accurately reflects physical and evolutionary characteristics. *BMC Bioinformatics*, 6(4):S8.

[Cutting, 2015] Cutting, G. R. (2015). Cystic fibrosis genetics: from molecular understanding to clinical application. *Nature Reviews Genetics*, 16(1):45–56. Number: 1 Publisher: Nature Publishing Group.

[David and Sternberg, 2023] David, A. and Sternberg, M. J. E. (2023). Protein structure-based evaluation of missense variants: Resources, challenges and future directions. *Current Opinion in Structural Biology*, 80:102600.

[Deiana et al., 2019] Deiana, A., Forcelloni, S., Porrello, A., and Giansanti, A. (2019). Intrinsically disordered proteins and structured proteins with intrinsically disordered regions have different functional roles in the cell. *PLOS ONE*, 14(8):e0217889. Publisher: Public Library of Science.

[Dong et al., 2018] Dong, R., Peng, Z., Zhang, Y., and Yang, J. (2018). mTM-align: an algorithm for fast and accurate multiple protein structure alignment. *Bioinformatics*, 34(10):1719–1725.

[D.R., 1964] D.R., D. (1964). A CORRELATION BETWEEN AMINO ACID COMPOSITION AND PROTEIN STRUCTURE. *Journal of molecular biology*, 9:605–609.

[Drew and Janes, 2019] Drew, E. D. and Janes, R. W. (2019). 2StrucCompare: a webserver for visualizing small but noteworthy differences between protein tertiary structures through interrogation of the secondary structure content. *Nucleic Acids Research*, 47(W1):W477–W481.

[Drozdetskiy et al., 2015] Drozdetskiy, A., Cole, C., Procter, J., and Barton, G. J. (2015). JPred4: a protein secondary structure prediction server. *Nucleic Acids Research*, 43(W1):W389–W394.

[Du et al., 2021] Du, Z., Su, H., Wang, W., et al. (2021). The trRosetta server for fast and accurate protein structure prediction. *Nature Protocols*, 16(12):5634–5651. Number: 12 Publisher: Nature Publishing Group.

[Dupuis et al., 2004] Dupuis, F., Sadoc, J.-F., and Mornon, J.-P. (2004). Protein secondary structure assignment through Voronoï tessellation. *Proteins: Structure, Function, and Bioinformatics*, 55(3):519–528. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.10566.

[Egli, 2016] Egli, M. (2016). Diffraction Techniques in Structural Biology. *Current Protocols in Nucleic Acid Chemistry*, 65(1):7.13.1–7.13.41. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpnc.4.

[Eisenberg, 2003] Eisenberg, D. (2003). The discovery of the alpha-helix and beta-sheet, the principal structural features of proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 100(20):11207–11210.

[Eisenhaber et al., 1995] Eisenhaber, F., , Bengt, P., , and Argos, P. (1995). Protein Structure Prediction: Recognition of Primary, Secondary, and Tertiary Structural Features from Amino Acid Sequence. *Critical Reviews in Biochemistry and Molecular Biology*, 30(1):1–94. Publisher: Taylor & Francis _eprint: https://doi.org/10.3109/10409239509085139.

[Fang et al., 2018] Fang, C., Shang, Y., and Xu, D. (2018). MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 86(5):592–598. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.25487.

[Faraggi et al., 2012] Faraggi, E., Zhang, T., Yang, Y., et al. (2012). SPINE X: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of Computational Chemistry*, 33(3):259–267. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.21968.

[Flower et al., 2000] Flower, D. R., North, A. C. T., and Sansom, C. E. (2000). The lipocalin protein family: structural and sequence overview. *Biochimica et Biophysica Acta (BBA) - Protein Structure and Molecular Enzymology*, 1482(1):9–24.

[Fodje and Al-Karadaghi, 2002] Fodje, M. and Al-Karadaghi, S. (2002). Occurrence, conformational features and amino acid propensities for the π-helix. *Protein Engineering, Design and Selection*, 15(5):353–358.

[Frenet, 1852] Frenet, F. (1852). Sur les courbes à double courbure. *Journal de Mathématiques Pures et Appliquées*, 17:437–447.

[Frishman and Argos, 1995] Frishman, D. and Argos, P. (1995). Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Bioinformatics*, 23(4):566–579. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.340230412.

[Fu et al., 2012] Fu, L., Niu, B., Zhu, Z., et al. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152.

[Geourjon and Deléage, 1995] Geourjon, C. and Deléage, G. (1995). SOPMA: significant improvements in protein secondary structure prediction by consensus prediction from multiple alignments. *Bioinformatics*, 11(6):681–684.

[Geurts et al., 2006] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42.

[Giardina et al., 1995] Giardina, B., , Irene, M., , Roberto, S., et al. (1995). The Multiple Functions of Hemoglobin. *Critical Reviews in Biochemistry and Molecular Biology*, 30(3):165–196. Publisher: Taylor & Francis _eprint: https://doi.org/10.3109/10409239509085142.

[Gonzalez et al., 2022] Gonzalez, N. A., Li, B. A., and McCully, M. E. (2022). The stability and dynamics of computationally designed proteins. *Protein Engineering, Design and Selection*, 35:gzac001.

[Graves, 2012] Graves, A. (2012). Long Short-Term Memory. In Graves, A., editor, *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 37–45. Springer, Berlin, Heidelberg.

[Guerler and Knapp, 2008] Guerler, A. and Knapp, E.-W. (2008). Novel protein folds and their nonsequential structural analogs. *Protein Science*, 17(8):1374–1382. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1110/ps.035469.108.

[Hanson et al., 2019] Hanson, J., Paliwal, K., Litfin, T., et al. (2019). Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks. *Bioinformatics*, 35(14):2403–2410.

[Havsteen, 1966] Havsteen, B. (1966). A study of the correlation between the amino acid composition and the helical content of proteins. *Journal of Theoretical Biology*, 10(1):1–10. Publisher: Elsevier.

[Heffernan et al., 2018] Heffernan, R., Paliwal, K., Lyons, J., et al. (2018). Single-sequence-based prediction of protein secondary structures and solvent accessibility by deep whole-sequence learning. *Journal of Computational Chemistry*, 39(26):2210–2216. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.25534.

[Heffernan et al., 2017] Heffernan, R., Yang, Y., Paliwal, K., and Zhou, Y. (2017). Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics*, 33(18):2842–2849.

[Ho et al., 2018] Ho, B., Baryshnikova, A., and Brown, G. W. (2018). Unification of Protein Abundance Datasets Yields a Quantitative Saccharomyces cerevisiae Proteome. *Cell Systems*, 6(2):192–205.e3. Publisher: Elsevier.

[Ho et al., 2021] Ho, C.-T., Huang, Y.-W., Chen, T.-R., et al. (2021). Discovering the Ultimate Limits of Protein Secondary Structure Prediction. *Biomolecules*, 11(11):1627. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[Holm, 2020] Holm, L. (2020). DALI and the persistence of protein shape. *Protein Science*, 29(1):128–140. 345 citations (Crossref) [2022-08-05] _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pro.3749.

[Hosseini et al., 2008] Hosseini, S.-R., Sadeghi, M., Pezeshk, H., et al. (2008). PROSIGN: A method for protein secondary structure assignment based on three-dimensional coordinates of consecutive Cα atoms. *Computational Biology and Chemistry*, 32(6):406–411.

[Hu et al., 2021] Hu, T., Chitnis, N., Monos, D., and Dinh, A. (2021). Next-generation sequencing technologies: An overview. *Human Immunology*, 82(11):801–811.

[Huang et al., 2014] Huang, Y. J., Mao, B., Aramini, J. M., and Montelione, G. T. (2014). Assessment of template-based protein structure predictions in CASP10. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):43–56. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.24488.

[Hubbard et al., 2002] Hubbard, T., Barker, D., Birney, E., et al. (2002). The Ensembl genome database project. *Nucleic Acids Research*, 30(1):38–41.

[Hunt et al., 1986] Hunt, D. F., Yates, J. R., Shabanowitz, J., et al. (1986). Protein sequencing by tandem mass spectrometry. *Proceedings of the National Academy of Sciences*, 83(17):6233–6237. Publisher: Proceedings of the National Academy of Sciences.

[Hutchinson and Thornton, 1996] Hutchinson, E. G. and Thornton, J. M. (1996). PROMOTIF—A program to identify and analyze structural motifs in proteins. *Protein Science*, 5(2):212–220. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pro.5560050204.

[Ilari and Savino, 2008] Ilari, A. and Savino, C. (2008). Protein Structure Determination by X-Ray Crystallography. In Keith, J. M., editor, *Bioinformatics: Data, Sequence Analysis and Evolution*, Methods in Molecular Biology™, pages 63–87. Humana Press, Totowa, NJ.

[Ingraham et al., 2019] Ingraham, J., Riesselman, A., Sander, C., and Marks, D. (2019). Learning Protein Structure with a Differentiable Simulator. In *Proceedings of the Seventh International Conference on Learning Representations*.

[Jiang et al., 2017] Jiang, Q., Jin, X., Lee, S.-J., and Yao, S. (2017). Protein secondary structure prediction: A survey of the state of the art. *Journal of Molecular Graphics and Modelling*, 76:379–402.

[Jones, 1999] Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices11Edited by G. Von Heijne. *Journal of Molecular Biology*, 292(2):195–202.

[Jones et al., 2025] Jones, M. S., Khanna, S., and Ferguson, A. L. (2025). FlowBack: A Generalized Flow-Matching Approach for Biomolecular Backmapping. *Journal of Chemical Information and Modeling*, 65(2):672–692. Publisher: American Chemical Society.

[Jumper et al., 2021] Jumper, J., Evans, R., Pritzel, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589. Number: 7873 Publisher: Nature Publishing Group.

[J. Miles et al., 2021] J. Miles, A., W. Janes, R., and A. Wallace, B. (2021). Tools and methods for circular dichroism spectroscopy of proteins: a tutorial review. *Chemical Society Reviews*, 50(15):8400–8413. Publisher: Royal Society of Chemistry.

[Kabsch, 1976] Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923. Number: 5 Publisher: International Union of Crystallography.

[Kabsch and Sander, 1983a] Kabsch, W. and Sander, C. (1983a). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bip.360221211.

[Kabsch and Sander, 1983b] Kabsch, W. and Sander, C. (1983b). How good are predictions of protein secondary structure? *FEBS letters*, 155(2):179–182. Publisher: Wiley Online Library.

[Kendrew et al., 1958] Kendrew, J. C., Bodo, G., Dintzis, H. M., et al. (1958). A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. *Nature*, 181(4610):662–666. Publisher: Nature Publishing Group.

[Keskin Karakoyun et al., 2023] Keskin Karakoyun, H., Yüksel, S. K., Amanoglu, I., et al. (2023). Evaluation of AlphaFold structure-based protein stability prediction on missense variations in cancer. *Frontiers in Genetics*, 14.

[Khoruddin et al., 2021] Khoruddin, N. A., Noorizhab, M. N., Teh, L. K., et al. (2021). Pathogenic nsSNPs that increase the risks of cancers among the Orang Asli and Malays. *Scientific Reports*, 11(1):16158. Number: 1 Publisher: Nature Publishing Group.

[King and Johnson, 1999] King, S. M. and Johnson, W. C. (1999). Assigning secondary structure from protein coordinate data. *Proteins: Structure, Function, and Bioinformatics*, 35(3):313–320. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291097-0134%2819990515%2935%3A3%3C313%3A%3AAID-PROT5%3E3.0.CO%3B2-1.

[Kmiecik et al., 2007] Kmiecik, S., Gront, D., and Kolinski, A. (2007). Towards the high-resolution protein structure prediction. Fast refinement of reduced models with all-atom force field. *BMC Structural Biology*, 7(1):43.

[Kneller and Hinsen, 2015] Kneller, G. R. and Hinsen, K. (2015). Protein secondary-structure description with a coarse-grained model. *Acta Crystallographica Section D: Biological Crystallography*, 71(7):1411–1422. Publisher: International Union of Crystallography.

[Koch and Cole, 2011] Koch, O. and Cole, J. (2011). An automated method for consistent helix assignment using turn information. *Proteins: Structure, Function, and Bioinformatics*, 79(5):1416–1426. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.22968.

[Konagurthu et al., 2011] Konagurthu, A. S., Allison, L., Stuckey, P. J., and Lesk, A. M. (2011). Piecewise linear approximation of protein structures using the principle of minimum message length. *Bioinformatics*, 27(13):i43–i51.

[Konagurthu et al., 2012] Konagurthu, A. S., Lesk, A. M., and Allison, L. (2012). Minimum message length inference of secondary structure from protein coordinate data. *Bioinformatics*, 28(12):i97–i105.

[Konc and Janežič, 2010] Konc, J. and Janežič, D. (2010). ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics*, 26(9):1160–1168.

[Kotelchuck and Scheraga, 1969] Kotelchuck, D. and Scheraga, H. (1969). The influence of short-range interactions on protein conformation, II. A model for predicting the α-helical regions of proteins. *Proceedings of the National Academy of Sciences*, 62(1):14–21.

[Kriegeskorte and Golan, 2019] Kriegeskorte, N. and Golan, T. (2019). Neural network models and deep learning. *Current Biology*, 29(7):R231–R236. Publisher: Elsevier.

[Krigbaum and Knutton, 1973] Krigbaum, W. and Knutton, S. P. (1973). Prediction of the amount of secondary structure in a globular protein from its aminoacid composition. *Proceedings of the National Academy of Sciences*, 70(10):2809–2813.

[Kryshtafovych et al., 2014] Kryshtafovych, A., Monastyrskyy, B., and Fidelis, K. (2014). CASP prediction center infrastructure and evaluation measures in CASP10 and CASP ROLL. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):7–13. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.24399.

[Kryshtafovych et al., 2021] Kryshtafovych, A., Schwede, T., Topf, M., et al. (2021). Critical assessment of methods of protein structure prediction (CASP)—Round XIV. *Proteins: Structure, Function, and Bioinformatics*, 89(12):1607–1617. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26237.

[Kryshtafovych et al., 2023] Kryshtafovych, A., Schwede, T., Topf, M., et al. (2023). Critical assessment of methods of protein structure prediction (CASP)—Round XV. *Proteins: Structure, Function, and Bioinformatics*, 91(12):1539–1549. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26617.

[Kuhlman and Bradley, 2019] Kuhlman, B. and Bradley, P. (2019). Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, 20(11):681–697. Publisher: Nature Publishing Group.

[Kumar and Bansal, 2015] Kumar, P. and Bansal, M. (2015). Identification of local variations within secondary structures of proteins. *Acta Crystallographica Section D*, 71(5):1077–1086. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1107/S1399004715003144.

[Kumar et al., 2009] Kumar, P., Henikoff, S., and Ng, P. C. (2009). Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nature Protocols*, 4(7):1073–1081. Number: 7 Publisher: Nature Publishing Group.

[Labesse et al., 1997] Labesse, G., Colloc'h, N., Pothier, J., and Mornon, J.-P. (1997). P-SEA: a new efficient assignment of secondary structure from Cα trace of proteins. *Bioinformatics*, 13(3):291–295.

[Lamond and Earnshaw, 1998] Lamond, A. I. and Earnshaw, W. C. (1998). Structure and Function in the Nucleus. *Science*, 280(5363):547–553. Publisher: American Association for the Advancement of Science.

[Landis and Koch, 1977] Landis, J. R. and Koch, G. G. (1977). A One-Way Components of Variance Model for Categorical Data. *Biometrics*, 33(4):671–679. Publisher: International Biometric Society.

[Law et al., 2014] Law, S. M., Frank, A. T., and Brooks III, C. L. (2014). PCASSO: A fast and efficient Cα-based method for accurately assigning protein secondary structure elements. *Journal of Computational Chemistry*, 35(24):1757–1761. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.23683.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J., et al. (1989). Handwritten Digit Recognition with a Back-Propagation Network. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

[Legrain et al., 2001] Legrain, P., Wojcik, J., and Gauthier, J.-M. (2001). Protein–protein interaction maps: a lead towards cellular functions. *Trends in Genetics*, 17(6):346–352. Publisher: Elsevier.

[Leszczynski and Rose, 1986] Leszczynski, J. F. and Rose, G. D. (1986). Loops in Globular Proteins: A Novel Category of Secondary Structure. *Science*, 234(4778):849–855. Publisher: American Association for the Advancement of Science.

[Levitt and Greer, 1977] Levitt, M. and Greer, J. (1977). Automatic identification of secondary structure in globular proteins. *Journal of Molecular Biology*, 114(2):181–239.

[Li et al., 2020] Li, Z., Jaroszewski, L., Iyer, M., et al. (2020). FATCAT 2.0: towards a better understanding of the structural diversity of proteins. *Nucleic Acids Research*, 48(W1):W60–W64.

[Lim, 1974] Lim, V. I. (1974). Algorithms for prediction of α-helical and β-structural regions in globular proteins. *Journal of Molecular Biology*, 88(4):873–894.

[Lin et al., 2023] Lin, Z., Akin, H., Rao, R., et al. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130. Publisher: American Association for the Advancement of Science.

[Liu and Wang, 2018] Liu, T. and Wang, Z. (2018). SOV_refine: A further refined definition of segment overlap score and its significance for protein structure similarity. *Source Code for Biology and Medicine*, 13(1):1.

[Magnan and Baldi, 2014] Magnan, C. N. and Baldi, P. (2014). SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, 30(18):2592–2597.

[Majumdar et al., 2005] Majumdar, I., Krishna, S. S., and Grishin, N. V. (2005). PALSSE: A program to delineate linear secondary structural elements from protein structures. *BMC Bioinformatics*, 6:202.

[Mariani et al., 2013] Mariani, V., Biasini, M., Barbato, A., and Schwede, T. (2013). lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728.

[Martin et al., 2005] Martin, J., Letellier, G., Marin, A., et al. (2005). Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. *BMC Structural Biology*, 5(1):17.

[McBride et al., 2023] McBride, J. M., Polev, K., Abdirasulov, A., et al. (2023). AlphaFold2 Can Predict Single-Mutation Effects. *Physical Review Letters*, 131(21):218401. Publisher: American Physical Society.

[Mirdita et al., 2022] Mirdita, M., Schütze, K., Moriwaki, Y., et al. (2022). ColabFold: making protein folding accessible to all. *Nature Methods*, 19(6):679–682. Number: 6 Publisher: Nature Publishing Group.

[Mistry et al., 2021] Mistry, J., Chuguransky, S., Williams, L., et al. (2021). Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill. Google-Books-ID: EoYBngEACAAJ.

[Mizuguchi et al., 1998] Mizuguchi, K., Deane, C. M., Blundell, T. L., et al. (1998). JOY: protein sequence-structure representation and analysis. *Bioinformatics*, 14(7):617–623.

[Morot-Gaudry et al., 2001] Morot-Gaudry, J.-F., Job, D., and Lea, P. J. (2001). Amino Acid Metabolism. In Lea, P. J. and Morot-Gaudry, J.-F., editors, *Plant Nitrogen*, pages 167–211. Springer, Berlin, Heidelberg.

[Nagy and Oostenbrink, 2014] Nagy, G. and Oostenbrink, C. (2014). Dihedral-Based Segment Identification and Classification of Biopolymers I: Proteins. *Journal of Chemical Information and Modeling*, 54(1):266–277. Publisher: American Chemical Society.

[Navada et al., 2011] Navada, A., Ansari, A. N., Patil, S., and Sonkamble, B. A. (2011). Overview of use of decision tree algorithms in machine learning. In *2011 IEEE Control and System Graduate Research Colloquium*, pages 37–42.

[Newman, 1980] Newman, D. J. (1980). Simple Analytic Proof of the Prime Number Theorem. *The American Mathematical Monthly*, 87(9):693–696. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00029890.1980.11995126.

[Nishikawa, 1983] Nishikawa, K. (1983). Assessment of secondary-structure prediction of proteins comparison of computerized Chou-Fasman method with others. *Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular Enzymology*, 748(2):285–299. Publisher: Elsevier.

[Novotny and Kleywegt, 2005] Novotny, M. and Kleywegt, G. J. (2005). A Survey of Left-handed Helices in Protein Structures. *Journal of Molecular Biology*, 347(2):231–241.

[Orengo et al., 1994] Orengo, C. A., Jones, D. T., and Thornton, J. M. (1994). Protein superfamilles and domain superfolds. *Nature*, 372(6507):631–634. Number: 6507 Publisher: Nature Publishing Group.

[Padian, 1999] Padian, K. (1999). Charles Darwin's Views of Classification in Theory and Practice. *Systematic Biology*, 48(2):352–364.

[Parisien and Major, 2005] Parisien, M. and Major, F. (2005). A new catalog of protein β-sheets. *Proteins: Structure, Function, and Bioinformatics*, 61(3):545–558. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.20677.

[Park et al., 2011] Park, S.-Y., Yoo, M.-J., Shin, J.-M., and Cho, K.-H. (2011). SABA (secondary structure assignment program based on only alpha carbons): a novel pseudo center geometrical criterion for accurate assignment of protein secondary structures. *BMB Reports*, 44(2):118–122. Publisher: Korean Society for Biochemistry and Molecular Biology.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Pauling et al., 1951] Pauling, L., Corey, R. B., and Branson, H. R. (1951). The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4):205–211. Publisher: Proceedings of the National Academy of Sciences.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12(null):2825–2830.

[Peng and Xu, 2011] Peng, J. and Xu, J. (2011). Raptorx: Exploiting structure information for protein alignment by statistical inference. *Proteins: Structure, Function, and Bioinformatics*, 79(S10):161–171. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.23175.

[Perez Martell et al., 2022] Perez Martell, R. I., Ziesel, A., Jabbari, H., and Stege, U. (2022). Supervised promoter recognition: a benchmark framework. *BMC Bioinformatics*, 23(1):118.

[Pham, 1981] Pham, A.-M. (1981). *Prediction program of secondary structure from sequence of proteins according to the method of Chou and Fasman*. PhD Thesis, University of British Columbia.

[Poole and Penny, 2001] Poole, A. and Penny, D. (2001). Does endo-symbiosis explain the origin of the nucleus? *Nature Cell Biology*, 3(8):E173–E173. Publisher: Nature Publishing Group.

[Privalov, 1997] Privalov, P. L. (1997). Thermodynamics of protein folding. *The Journal of Chemical Thermodynamics*, 29(4):447–474.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

[Raganathan et al., 2008] Raganathan, S., Izotov, D., Kraka, E., and Cremer, D. (2008). Automated and accurate protein structure description: Distribution of Ideal Secondary Structural Units in Natural Proteins. arXiv:0811.3587 [q-bio].

[Ramachandran et al., 1963] Ramachandran, G. N., Ramakrishnan, C., and Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7(1):95–99.

[Ramazi and Zahiri, 2021] Ramazi, S. and Zahiri, J. (2021). Post-translational modifications in proteins: resources, tools and prediction methods. *Database*, 2021:baab012.

[Rehman et al., 2022] Rehman, I., Farooq, M., and Botelho, S. (2022). *Biochemistry, Secondary Protein Structure*. StatPearls Publishing, Treasure Island (FL).

[Ren et al., 2023] Ren, F., Ding, X., Zheng, M., et al. (2023). AlphaFold accelerates artificial intelligence powered drug discovery: efficient discovery of a novel CDK20 small molecule inhibitor. *Chemical Science*, 14(6):1443–1452. Publisher: The Royal Society of Chemistry.

[Richards and Kundrot, 1988] Richards, F. M. and Kundrot, C. E. (1988). Identification of structural motifs from protein coordinate data: Secondary structure and first-level supersecondary structure. *Proteins: Structure, Function, and Bioinformatics*, 3(2):71–84. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.340030202.

[Robinow and Kellenberger, 1994] Robinow, C. and Kellenberger, E. (1994). The bacterial nucleoid revisited. *Microbiological Reviews*, 58(2):211–232. Publisher: American Society for Microbiology.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Rost et al., 1994a] Rost, B., Sander, C., and Schneider, R. (1994a). PHD-an automatic mail server for protein secondary structure prediction. *Bioinformatics*, 10(1):53–60.

[Rost et al., 1994b] Rost, B., Sander, C., and Schneider, R. (1994b). Redefining the goals of protein secondary structure prediction. *Journal of Molecular Biology*, 235(1):13–26.

[Sachs, 1993] Sachs, A. B. (1993). Messenger RNA degradation in eukaryotes. *Cell*, 74(3):413–421.

[Sadowski and Taylor, 2012] Sadowski, M. I. and Taylor, W. R. (2012). Evolutionary inaccuracy of pairwise structural alignments. *Bioinformatics*, 28(9):1209–1215.

[Salawu, 2016] Salawu, E. O. (2016). RaFoSA: Random forests secondary structure assignment for coarse-grained and all-atom protein systems. *Cogent Biology*, 2(1):1214061. Publisher: Cogent OA _eprint: https://doi.org/10.1080/23312025.2016.1214061.

[Sauna and Kimchi-Sarfaty, 2022] Sauna, Z. E. and Kimchi-Sarfaty, C., editors (2022). *Single Nucleotide Polymorphisms: Human Variation and a Coming Revolution in Biology and Medicine*. Springer International Publishing, Cham.

[Schaarschmidt et al., 2018] Schaarschmidt, J., Monastyrskyy, B., Kryshtafovych, A., and Bonvin, A. M. (2018). Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age. *Proteins: Structure, Function, and Bioinformatics*, 86(S1):51–66. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.25407.

[Schaeffer et al., 2017] Schaeffer, R. D., Liao, Y., Cheng, H., and Grishin, N. V. (2017). ECOD: new developments in the evolutionary classification of domains. *Nucleic Acids Research*, 45(D1):D296–D302.

[Schaffer, 1991] Schaffer, C. (1991). When does overfitting decrease prediction accuracy in induced decision trees and rule sets? In Kodratoff, Y., editor, *Machine Learning — EWSL-91*, pages 192–205, Berlin, Heidelberg. Springer.

[Schilder and Ubbink, 2013] Schilder, J. and Ubbink, M. (2013). Formation of transient protein complexes. *Current Opinion in Structural Biology*, 23(6):911–918.

[Senior et al., 2020] Senior, A. W., Evans, R., Jumper, J., et al. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710. Number: 7792 Publisher: Nature Publishing Group.

[Serret, 1851] Serret, J.-A. (1851). Sur quelques formules relatives à la théorie des courbes à double courbure. *Journal de Mathématiques Pures et Appliquées*, 16:193–207.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423. Conference Name: The Bell System Technical Journal.

[Shindyalov and Bourne, 1998] Shindyalov, I. N. and Bourne, P. E. (1998). Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering, Design and Selection*, 11(9):739–747.

[Sidi and Keasar, 2020] Sidi, T. and Keasar, C. (2020). Redundancy-weighting the PDB for detailed secondary structure prediction using deep-learning models. *Bioinformatics*, 36(12):3733–3738.

[Sievers and Higgins, 2014] Sievers, F. and Higgins, D. G. (2014). Clustal Omega. *Current Protocols in Bioinformatics*, 48(1):3.13.1–3.13.16. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471250953.bi0313s48.

[Sillitoe et al., 2021] Sillitoe, I., Bordin, N., Dawson, N., et al. (2021). CATH: increased structural coverage of functional space. *Nucleic Acids Research*, 49(D1):D266–D273.

[Singh et al., 2021] Singh, J., Litfin, T., Paliwal, K., et al. (2021). SPOT-1D-Single: improving the single-sequence-based prediction of protein secondary structure, backbone angles, solvent accessibility and half-sphere exposures using a large training set and ensembled deep learning. *Bioinformatics*, 37(20):3464–3472.

[Singh et al., 2022] Singh, J., Paliwal, K., Litfin, T., et al. (2022). Reaching alignment-profile-based accuracy in predicting protein secondary and tertiary structural properties without alignment. *Scientific Reports*, 12(1):7607.

[Sitbon and Pietrokovski, 2007] Sitbon, E. and Pietrokovski, S. (2007). Occurrence of protein structure elements in conserved sequence regions. *BMC Structural Biology*, 7(1):3.

[Sklenar et al., 1989] Sklenar, H., Etchebest, C., and Lavery, R. (1989). Describing protein structure: A general algorithm yielding complete helicoidal parameters and a unique overall axis. *Proteins: Structure, Function, and Bioinformatics*, 6(1):46–60. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.340060105.

[Smith, 2001] Smith, J. B. (2001). Peptide Sequencing by Edman Degradation. In John Wiley & Sons, Ltd, editor, *eLS*. Wiley, 1 edition.

[Smolarczyk et al., 2020] Smolarczyk, T., Roterman-Konieczna, I., and Stapor, K. (2020). Protein Secondary Structure Prediction: A Review of Progress and Directions. *Current Bioinformatics*, 15(2):90–107.

[Sobolev et al., 1999] Sobolev, V., Sorokine, A., Prilusky, J., et al. (1999). Automated analysis of interatomic contacts in proteins. *Bioinformatics*, 15(4):327–332.

[Song et al., 2024] Song, F. V., Su, J., Huang, S., et al. (2024). DeepSS2GO: protein function prediction from secondary structure. *Briefings in Bioinformatics*, 25(3):bbae196.

[Spassov et al., 2007] Spassov, V. Z., Yan, L., and Flook, P. K. (2007). The dominant role of side-chain backbone interactions in structural realization of amino acid code. ChiRotor: A side-chain prediction algorithm based on side-chain backbone interactions. *Protein Science : A Publication of the Protein Society*, 16(3):494–506.

[Srinivasan and Rose, 1999] Srinivasan, R. and Rose, G. D. (1999). A physical basis for protein secondary structure. *Proceedings of the National Academy of Sciences*, 96(25):14258–14263. Publisher: Proceedings of the National Academy of Sciences.

[Stein and Moore, 1961] Stein, W. H. and Moore, S. (1961). The Chemical Structure of Proteins. *Scientific American*, 204(2):81–95. Publisher: Scientific American, a division of Nature America, Inc.

[Stenson et al., 2020] Stenson, P. D., Mort, M., Ball, E. V., et al. (2020). The Human Gene Mutation Database (HGMD®): optimizing its use in a clinical diagnostic or research setting. *Human Genetics*, 139(10):1197–1207.

[Sun et al., 2019] Sun, Z., Liu, Q., Qu, G., et al. (2019). Utility of B-Factors in Protein Science: Interpreting Rigidity, Flexibility, and Internal Motion and Engineering Thermostability. *Chemical Reviews*, 119(3):1626–1665. Publisher: American Chemical Society.

[Svenningsen et al., 2017] Svenningsen, S. L., Kongstad, M., Stenum, T. S., et al. (2017). Transfer RNA is highly unstable during early amino acid starvation in Escherichia coli. *Nucleic Acids Research*, 45(2):793–804.

[Söding, 2005] Söding, J. (2005). Protein homology detection by HMM–HMM comparison. *Bioinformatics*, 21(7):951–960.

[Taylor et al., 2005] Taylor, T., Rivera, M., Wilson, G., and Vaisman, I. I. (2005). New method for protein secondary structure assignment based on a simple topological descriptor. *Proteins*, 60(3):513–524.

[Taylor, 2001] Taylor, W. R. (2001). Defining linear segments in protein structure1. *Journal of Molecular Biology*, 310(5):1135–1150.

[The UniProt Consortium, 2023] The UniProt Consortium (2023). UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531.

[Tian and Zhang, 2022] Tian, Y. and Zhang, Y. (2022). A comprehensive survey on regularization strategies in machine learning. *Information Fusion*, 80:146–166.

[Traub and Shmueli, 1963] Traub, W. and Shmueli, U. (1963). Structure of Poly-L-Proline I. *Nature*, 198(4886):1165–1166. Publisher: Nature Publishing Group.

[Unger and Moult, 1993] Unger, R. and Moult, J. (1993). Finding the lowest free energy conformation of a protein is an NP-hard problem: Proof and implications. *Bulletin of Mathematical Biology*, 55(6):1183–1198.

[Vacic and M. Iakoucheva, 2012] Vacic, V. and M. Iakoucheva, L. (2012). Disease mutations in disordered regions—exception to the rule? *Molecular BioSystems*, 8(1):27–32. Publisher: Royal Society of Chemistry.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Wang and Dunbrack, 2003] Wang, G. and Dunbrack, R. L. (2003). PISCES: a protein sequence culling server. *Bioinformatics (Oxford, England)*, 19(12):1589–1591.

[Wang et al., 2016a] Wang, S., Li, W., Liu, S., and Xu, J. (2016a). RaptorX-Property: a web server for protein structure property prediction. *Nucleic Acids Research*, 44(W1):W430–W435.

[Wang et al., 2016b] Wang, S., Peng, J., Ma, J., and Xu, J. (2016b). Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. *Scientific Reports*, 6(1):18962. Number: 1 Publisher: Nature Publishing Group.

[Wu and Carricato, 2020] Wu, Y. and Carricato, M. (2020). Persistent manifolds of the special Euclidean group SE(3): A review. *Computer Aided Geometric Design*, 79:101872.

[Wüthrich, 1989] Wüthrich, K. (1989). Protein Structure Determination in Solution by Nuclear Magnetic Resonance Spectroscopy. *Science*, 243(4887):45–50. Publisher: American Association for the Advancement of Science.

[Xu et al., 2021] Xu, J., McPartlon, M., and Li, J. (2021). Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nature Machine Intelligence*, 3(7):601–609. Number: 7 Publisher: Nature Publishing Group.

[Yan and Maier, 2009] Yan, X. and Maier, C. S. (2009). Hydrogen/Deuterium Exchange Mass Spectrometry. In Lipton, M. S. and Paša-Tolic, L., editors, *Mass Spectrometry of Proteins and Peptides: Methods and Protocols*, Methods In Molecular Biology, pages 255–271. Humana Press, Totowa, NJ.

[Yang et al., 2018] Yang, Y., Gao, J., Wang, J., et al. (2018). Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in Bioinformatics*, 19(3):482–494.

[Yaseen and Li, 2014] Yaseen, A. and Li, Y. (2014). Template-based C8-SCORPION: a protein 8-state secondary structure prediction method using structural information and context-based features. *BMC Bioinformatics*, 15(8):S3.

[Zacharias and Knapp, 2014] Zacharias, J. and Knapp, E.-W. (2014). Protein secondary structure classification revisited: processing DSSP information with PSSC. *Journal of Chemical Information and Modeling*, 54(7):2166–2179.

[Zemla, 2003] Zemla, A. (2003). LGA: a method for finding 3D similarities in protein structures. *Nucleic Acids Research*, 31(13):3370–3374.

[Zemla et al., 1999] Zemla, A., Venclovas, C., Fidelis, K., and Rost, B. (1999). A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Bioinformatics*, 34(2):220–223. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291097-0134%2819990201%2934%3A2%3C220%3A%3AAID-PROT7%3E3.0.CO%3B2-K.

[Zemla et al., 2022] Zemla, A. T., Allen, J. E., Kirshner, D., and Lightstone, F. C. (2022). PDBspheres: a method for finding 3D similarities in local regions in proteins. *NAR Genomics and Bioinformatics*, 4(4):lqac078.

[Zhang et al., 2018] Zhang, B., Li, J., and Lü, Q. (2018). Prediction of 8-state protein secondary structures by a novel deep learning architecture. *BMC Bioinformatics*, 19(1):293.

[Zhang et al., 2020] Zhang, K., Pintilie, G. D., Li, S., et al. (2020). Resolving individual atoms of protein complex by cryo-electron microscopy. *Cell Research*, 30(12):1136–1139. Number: 12 Publisher: Nature Publishing Group.

[Zhang et al., 2008] Zhang, W., Dunker, A. K., and Zhou, Y. (2008). Assessing secondary structure assignment of protein structures by using pairwise sequence-alignment benchmarks. *Proteins: Structure, Function, and Bioinformatics*, 71(1):61–67. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.21654.

[Zhang, 2008] Zhang, Y. (2008). I-TASSER server for protein 3D structure prediction. *BMC Bioinformatics*, 9(1):40.

[Zhang and Sagui, 2015] Zhang, Y. and Sagui, C. (2015). Secondary structure assignment for conformationally irregular peptides: Comparison between DSSP, STRIDE and KAKSI. *Journal of Molecular Graphics and Modelling*, 55:72–84.

[Zhang and Skolnick, 2004] Zhang, Y. and Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.20264.

[Zhang and Skolnick, 2005] Zhang, Y. and Skolnick, J. (2005). TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309.

[Zheng et al., 2023] Zheng, W., Wuyun, Q., Freddolino, L., and Zhang, Y. (2023). Integrating deep learning, threading alignments, and a multi-MSA strategy for high-quality protein monomer and complex structure prediction in CASP15. *Proteins: Structure, Function, and Bioinformatics*, 91(12):1684–1703. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26585.

[Zhong et al., 2022] Zhong, B., Su, X., Wen, M., et al. (2022). ParaFold: Paralleling AlphaFold for Large-Scale Predictions. In *International Conference on High Performance Computing in Asia-Pacific Region Workshops*, HPCAsia '22 Workshops, pages 1–9, New York, NY, USA. Association for Computing Machinery.

Appendices

Appendix A

Supplementary information

A.1. Secondary structure classification

Proteins are composed of one or more structural domains that can fold independently of each other. Different combinations and arrangements of secondary structures can result in different protein folds, which have functional and evolutionary implications [Andreeva et al., 2020]. Folds are defined by the arrangement and connectivity of secondary structures in the protein. Protein secondary structure is the local spatial conformation of the protein's backbone atoms. This conformation occurs through the pattern of hydrogen bonds between the amino hydrogen and carboxyl oxygen atoms in the backbone. Secondary structures in protein form as an intermediate before the protein folds into its three dimensional tertiary structure. The two most common secondary structures are α helices and β sheets, which were first theorized by Pauling and Corey [Pauling et al., 1951] and described as:

- α helices are right-handed spiral conformations of polypeptide chains. In α helices, every backbone amino ($N - H$) group donates a hydrogen bond to the backbone carbonyl ($C = O$) group, which is located four residues prior. This creates a bond between these groups and links them into its spiral conformation.
- β sheets are also formed by hydrogen bonding between carbonyl and amino groups that make up the protein backbone and cause the molecule to bend and fold into a pleated sheet form.

Secondary structures were first classified into three categories, including the two most common structures and referring to everything else as coil. Afterwards, the need for a more detailed classification lead to the creation of algorithmic structure categories by Kabsch and Sander [Kabsch and Sander, 1983a] which were subsequently refined into the following classes:

- 3-turn helix (3 10 helix). A helix-like structure with a minimum length of 3 residues. This class is denoted by the letter G.
- 4-turn helix (α helix). A helix-like structure with a minimum length of 4 residues. This class is denoted by the letter H.

- 5-turn helix (π helix). A helix-like structure with a minimum length 5 residues. This class is denoted by the letter I.
- hydrogen bonded turn (3, 4 or 5 residue turn). Forms a turn-like structure and is denoted by the letter T.
- extended strand in a β-sheet conformation forming a pleated sheet structure. This class is denoted by the letter E.
- residue in an isolated β-bridge (single pair β-sheet hydrogen bond formation). This class is denoted by the letter B.
- bend (the only non-hydrogen-bond based assignment). This class is denoted by the letter S.
- coil (none of the above), denoted by the letter C.
- PPII helix (polyproline helix). A helix-like structure made up of repeating proline residues. This class is denoted by the letter P.

The previous list consists of nine categories, as the polyproline helix was later added to a newer version of the DSSP software remade by the PDB team.

A.2. Secondary structure assignment

We used DSSP to assign secondary structures to the protein structure data. The macromolecular crystallographic information file (mmCIF) format was selected because the PDB has required it for all new submissions since 2019 [Adams et al., 2019]. As a result, structures submitted to the PDB after 2019 are no longer available in the older 'PDB' format, rendering older software incompatible with recent data. DSSP has been updated to process the mmCIF format and has been extensively tested by the community.

DSSP's algorithm for assigning secondary structures to mmCIF-formatted input has evolved from its original version, which processed 'PDB' files. The main differences include both the file formatting and the secondary structure classification, which varies from the traditional 8 classes, as shown in Table A.1.

When producing mmCIF output, DSSP expands the input file by rewriting it and appending the secondary structure information. However, this rewriting process can introduce formatting errors, particularly when the input file contains quotations with the character ' . To address these issues, we provide a script that properly re-formats DSSP's mmCIF output.

As noted earlier and shown in Table A.1, the mmCIF format classifies both the β-bridge and Strand under the same STRN class, making it difficult to distinguish between them. Kabash and Sander, in their original DSSP publication, refer to β-bridge as an "isolated bridge," which is formed by a single hydrogen bond similar to those found in a β-sheet (DSSP's Strand class).

DSSP Class (Q8)	PDB Class	mmCIF Class	Description
B	B	STRN	β-bridge
C	' '(space)	OTHER	Loop or Coil
E	E	STRN	Strand
G	G	HELX_RH_3T_P	3-10 helix
H	H	HELX_RH_AL_P	α-helix
I	I	HELX_RH_PI_P	π-helix
S	S	BEND	Bend
T	T	TURN_TY1_P	Turn
	P	HELX_RH_PP_P	PPII-helix

Table A.1.: DSSP class conversion by input format. Note that classes β-bridge and Strand are indistinguishable by the mmCIF DSSP algorithm. Since Q8 does not designate polyproline helices, they are regarded as coil.

According to Kabash and Sander's description of secondary structures, an isolated bridge is part of the repeating hydrogen-bonding patterns "turn" and "bridge." Repeating turns form "helices," repeating bridges form "ladders," and connected ladders form "sheets." Based on this understanding, we convert the mmCIF output back to the original classes used in the 'PDB' format, which are recognized by all secondary structure prediction tools. Specifically, when we encounter an isolated STRN, we convert it to its corresponding β-bridge (**B**) class.

It is also important to note that the polyproline (PPII) class is a newer addition in the mmCIF format. However, since prediction methods rely on the Q8 assignment, the PPII class is not used in our analysis. Additionally, the OTHER class does not appear in the output, as DSSP only generates proper secondary structural classes. The OTHER class can be inferred when a residue lacks a structural classification.

By applying these conversions, we can accurately assess mmCIF-formatted structure files in terms of the original Q8 classification scheme.

A.3. Mutational dataset

The data we utilize for this purpose is taken from the Protein Data Bank (PDB). We obtain all current sequences of all experimentally derived 3D protein structures as of April 2023. We then remove non-protein sequences (e.g. RNA) and identical duplicates of protein sequences. This is followed by clustering of the sequences via cd-hit for 99% sequence similarity to obtain clusters containing a few amino acid mutations. This clustering procedure can also be extended for use with other tools, such as mmseqs2.

Each cluster is further filtered to remove singletons, clusters containing a single protein sequence, and duplicate sequences as a precaution from any clustering issue. Next, we align all sequences within each cluster to locate the amino acid mutations. The alignments are

put through the singleton and duplicate filtering step once again in case of any alignments that create duplicates which subsequently are filtered out into singleton clusters.

Occasionally, experimentally obtained 3D protein structures exclude ambiguously located amino acids. This exclusion leads to gaps in protein sequences. Furthermore, the alignment of protein sequences can add gaps for optimal alignment. Therefore, we filter sequences containing either of these cases to obtain solely unambiguous and ungapped alignments.

Now that the clusters are obtained and favorable proteins have been chosen, we download the 3D structures of all proteins within every cluster. The 3D structure files are obtained as mmCIF files since this format is the modernized equivalent of PDB files that are slowly being phased out because of technical limitations.

Our previously obtained ungapped alignments go through another iteration of alignment. We do this because the filtering and transformation steps can create unaligned sequences within clusters. Once we have obtained our target sequences and corresponding structures, we must ensure that the protein 3D structures contain the amino acid mutations that our aligned sequences have uncovered. This is due to experimentally obtained protein structures not being capable of perfectly mapping the location of all amino acids, which would discard ambiguously located amino acids entirely from the 3D structure files but keep the protein sequence intact. Therefore, we remove any sequences and structures that do not contain the clustered amino acid mutation. We also group sequences based on the amino acid mutation that they contain and remove any sequences that are not indispensable for uniquely characterizing at least one amino acid mutation. Finally, to obtain clear preliminary results we cleaned the data further to only contain sequences without gaps or unknown amino acids and term this data as preprocessed.

With our preprocessed data, we run DSSP to assign the secondary structure of each protein sequence with their 3D structure. We also run current secondary structure prediction tools on our data to see if they can accurately predict secondary structural changes on single amino acid mutations. Every tool's output is then normalized for comparison purposes.

With the normalized outputs, we are able to do a local and global benchmark on each tools capability to predict the secondary structure on our clusters of mutated proteins.

A.4. Mutation extraction

Finding mutations from a series of aligned sequences and assigning a consensus or mutation value to a sequence is non-trivial. To find mutations in our preprocessed data, we created a similar method from Weblogo [Crooks et al., 2004] which uses information theory to provide the significance of each mutation. The height of each mutation in the logo is characterized by the frequency (p_i) of the amino acid at a specific position i and subsequently through Shannon entropy,

$$S = -\sum_i p_i \log_2 p_i \tag{A.1}$$

Equation A.1 ranges from $[0, 1]$ with a domain of frequencies of $[0, 1]$ and has a characteristic bell shaped curve with a maximum on 0.5. Therefore, values for frequencies that are equally spaced apart from the maximum value give the same result. Unfortunately, we require a method that can distinguish between such values since the consensus frequency might be equally spaced apart from the mutation frequency (e.g. 0.25 and 0.75), but clearly one is more frequent than the other. Therefore we decided to modify the equation as follows,

$$S = -\sum_i p_i \log_2 (1 - p_i) \tag{A.2}$$

This was done to obtain mutation positions easily since positions without mutations will return 0 or Infinity. Infinity values are subsequently turned to 0. Any non-zero value will be valid for a mutation or consensus amino acid in a position where a mutation has occurred. By using sparse matrices, it is then simple to systematically find these non-zero values and assign its sequence either a consensus (maximum value) or mutation (others) category for a location in the sequence.

A.5. Protein data statistics for Mutational Sufficiency

Proteins were filtered to those containing only single amino acid mutations. We show that the filtering process does not significantly alter the distribution of proteins lengths before and after the filtration procedure in the Fig. A.1 and Fig.A.2. Most changes occur in long proteins with over 600 amino acids in length. These proteins contained a higher number of mutations as expected from their length. Clusters with a high number of proteins also decrease as duplicate single amino acid mutations found in several proteins inside a cluster were filtered out by the Mutation Sufficiency process.

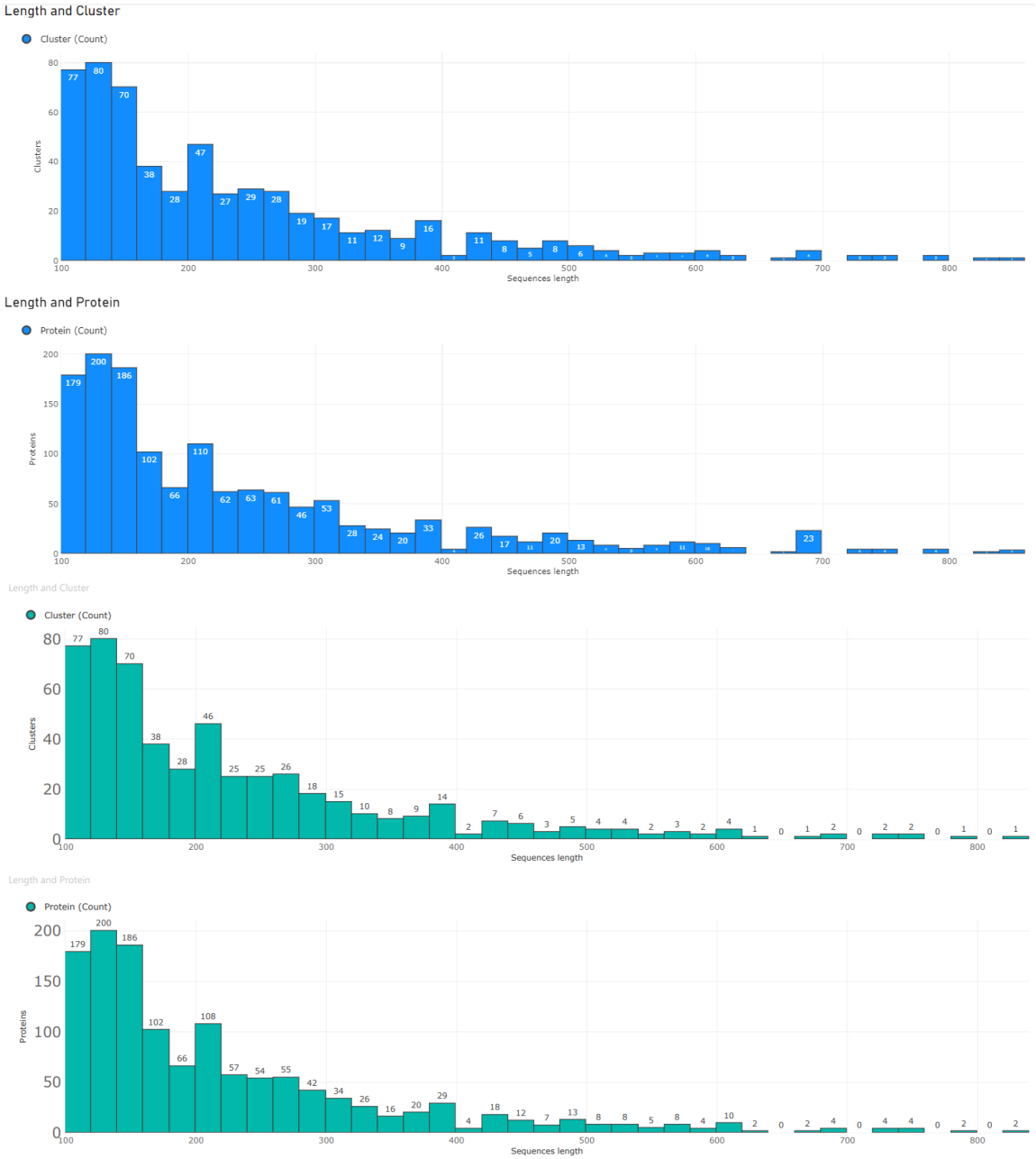


Figure A.1.|: **Sequence lengths by cluster and protein** Top (blue): Number of clusters for sequence lengths of proteins, and number of proteins with certain sequence length before filtering. Bottom (green): After $MutationSufficiency$ filtering.

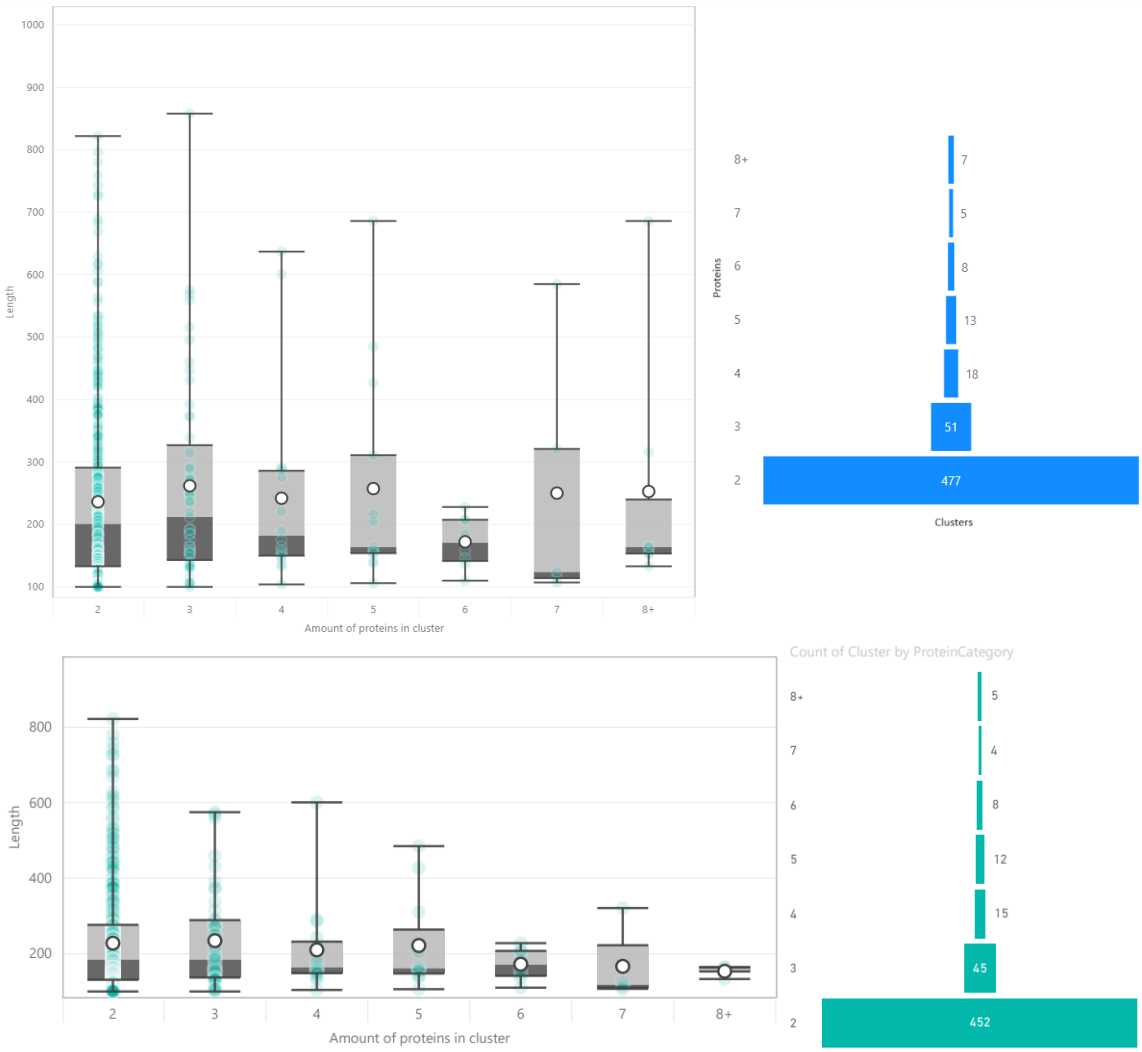


Figure A.2.|: **Length of proteins in clusters per amount of proteins.** Top (blue): Before filtering. Bottom (green): After $MutationSufficiency$ filtering.

A.6. Mutation performance from prediction methods

Section containing secondary structure metrics obtained for all prediction methods tested in the main text. 'Top' performing methods are contained in Tables A.2 and A.3. 'Average' performing methods are shown in Table A.5, and finally 'Low' performing methods are shown in Table A.4.

Table A.2.: **Single amino acid mutation benchmark on secondary structure for 'top' performing methods.**

Method	Type	Vicinity	Accuracy	SOV99	SOV_refine
af2	1d	distant	0.928	0.925	0.938
		global	0.925	0.922	0.935
		local	0.914	0.914	0.923
	2d	distant	0.927	0.924	0.937
		global	0.925	0.922	0.935
		local	0.911	0.910	0.919
	3d	distant	0.926	0.923	0.937
		global	0.925	0.922	0.935
		local	0.916	0.905	0.918
	contact	distant	0.926	0.923	0.936
		global	0.925	0.922	0.935
		local	0.918	0.910	0.922
colabfold	1d	distant	0.929	0.926	0.938
		global	0.927	0.924	0.936
		local	0.919	0.919	0.928
	2d	distant	0.929	0.925	0.938
		global	0.927	0.924	0.936
		local	0.915	0.914	0.924
	3d	distant	0.928	0.924	0.937
		global	0.927	0.924	0.936
		local	0.920	0.910	0.923
	contact	distant	0.927	0.923	0.937
		global	0.927	0.924	0.936
		local	0.922	0.915	0.927

Table A.3.: **Single amino acid mutation benchmark on secondary structure for 'top' performing methods.**

Method	Type	Vicinity	Accuracy	SOV99	SOV_refine
esmfold	1d	distant	0.896	0.891	0.908
		global	0.893	0.888	0.904
		local	0.882	0.879	0.891
	2d	distant	0.895	0.891	0.907
		global	0.893	0.888	0.904
		local	0.874	0.870	0.883
	3d	distant	0.895	0.890	0.908
		global	0.893	0.888	0.904
		local	0.877	0.863	0.881
	contact	distant	0.894	0.889	0.907
		global	0.893	0.888	0.904
		local	0.880	0.868	0.885
sspro8	1d	distant	0.943	0.940	0.951
		global	0.940	0.937	0.948
		local	0.932	0.931	0.940
	2d	distant	0.942	0.939	0.950
		global	0.940	0.937	0.948
		local	0.928	0.926	0.935
	3d	distant	0.941	0.938	0.950
		global	0.940	0.937	0.948
		local	0.934	0.926	0.940
	contact	distant	0.940	0.938	0.949
		global	0.940	0.937	0.948
		local	0.937	0.931	0.942

Table A.4.: **Single amino acid mutation benchmark on secondary structure for 'low' performing methods.**

Method	Type	Vicinity	Accuracy	SOV99	SOV_refine
raptorx	1d	distant	0.605	0.553	0.576
		global	0.600	0.547	0.571
		local	0.584	0.543	0.558
	2d	distant	0.604	0.553	0.576
		global	0.600	0.547	0.571
		local	0.563	0.526	0.539
	3d	distant	0.603	0.550	0.579
		global	0.600	0.547	0.571
		local	0.571	0.508	0.534
	contact	distant	0.600	0.547	0.578
		global	0.600	0.547	0.571
		local	0.583	0.524	0.550
rgn2	1d	distant	0.627	0.607	0.630
		global	0.622	0.602	0.625
		local	0.598	0.583	0.600
	2d	distant	0.626	0.605	0.628
		global	0.622	0.602	0.625
		local	0.582	0.568	0.585
	3d	distant	0.627	0.604	0.631
		global	0.622	0.602	0.625
		local	0.584	0.554	0.582
	contact	distant	0.626	0.602	0.631
		global	0.622	0.602	0.625
		local	0.593	0.561	0.592
spot1d_single	1d	distant	0.666	0.622	0.649
		global	0.661	0.616	0.642
		local	0.641	0.610	0.628
	2d	distant	0.664	0.621	0.648
		global	0.661	0.616	0.642
		local	0.623	0.594	0.613
	3d	distant	0.663	0.618	0.651
		global	0.661	0.616	0.642
		local	0.630	0.576	0.604
	contact	distant	0.660	0.615	0.649
		global	0.661	0.616	0.642
		local	0.643	0.590	0.619

Table A.5.: **Single amino acid mutation benchmark on secondary structure for 'average' performing methods.**

Method	Type	Vicinity	Accuracy	SOV99	SOV_refine
spot1d	1d	distant	0.793	0.783	0.810
		global	0.789	0.780	0.807
		local	0.775	0.771	0.792
	2d	distant	0.792	0.782	0.809
		global	0.789	0.780	0.807
		local	0.761	0.754	0.777
	3d	distant	0.791	0.778	0.808
		global	0.789	0.780	0.807
		local	0.768	0.744	0.776
	contact	distant	0.790	0.776	0.808
		global	0.789	0.780	0.807
		local	0.774	0.750	0.780
spot1d_lm	1d	distant	0.825	0.812	0.836
		global	0.821	0.808	0.833
		local	0.807	0.797	0.817
	2d	distant	0.824	0.810	0.835
		global	0.821	0.808	0.833
		local	0.797	0.786	0.807
	3d	distant	0.822	0.807	0.836
		global	0.821	0.808	0.833
		local	0.804	0.779	0.809
	contact	distant	0.821	0.806	0.835
		global	0.821	0.808	0.833
		local	0.807	0.783	0.811

A.7. Protein properties

In the main text, we include properties for each of the proteins of interest and to show the strengths and weaknesses of the prediction methods. These properties were aggregated from CATH, SCOP and PDB descriptor data. We generated word clouds to obtain the most common words, which resulted in the descriptors for the proteins. This section contains all the figures with their corresponding protein property word clouds.

The bar plots are shown in the main text, but we decide to include both the plots and word clouds together to more easily match the properties to their corresponding proteins. The larger the word in the word cloud, the more common the property is for the respective bar plot proteins. Figures for protein properties: A.3, A.4, A.5, A.6, A.7, A.8, A.9.

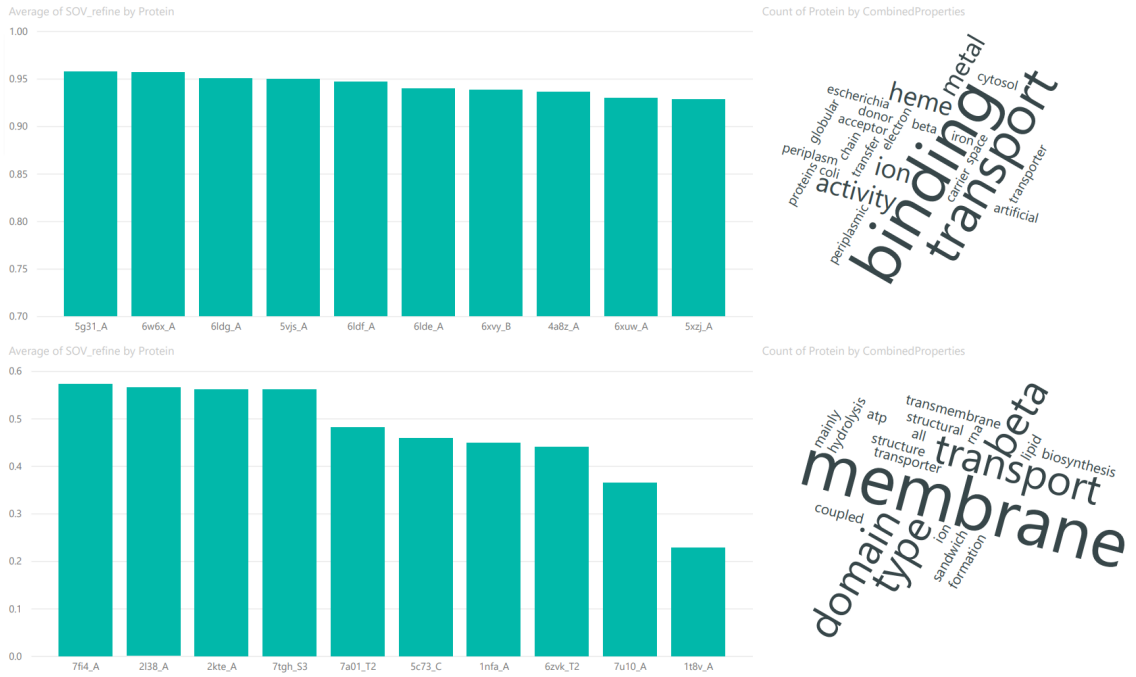


Figure A.3.|: **Overall protein structure prediction results.** Top: Best overall predicted proteins and their properties. Bottom: Worst overall predicted proteins and their properties.

Figure A.4.|: **Mutation stability results.** Disruptive and Stable mutations. Stabilizing mutations occur more often in PDB data than in prediction methods, as the latter almost always predicts destabilizing mutations. The exception is SSPro8 while still missing two thirds of stabilizing mutations.

Figure A.5.|: **Best results per method category.** Best predicted proteins along with their properties for each method category (top-performing, average performing and low performing). From left to right, Top performing methods, Average performing methods, and Low performing methods.

Figure A.6.|: **Worst-predicted proteins per method category.** The worst-predicted proteins along with their properties for the different method categories. From left to right, Top performing methods, Average performing methods, and Low performing methods.

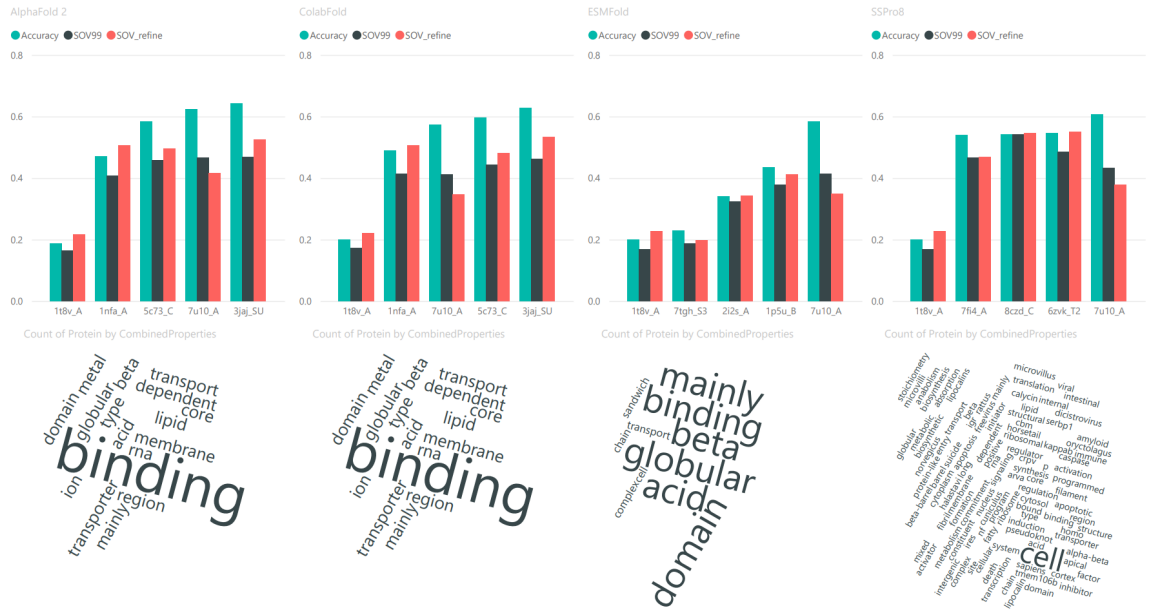


Figure A.7.|: **Top performing methods.** Worst performing proteins for each of the top performing methods. From left to right: AlphaFold2, ColabFold, ESMFold, and SSPro8.

Figure A.8.|: **Average performing methods.** Worst performing proteins for each of the average performing methods. Left: SPOT-1D, Right: SPOT-1D-LM.

Figure A.9.|: **Low performing methods.** Worst performing proteins for each of the low performing methods. From left to right: Raptor-X Property, SPOT-1D-Single, and RGN2.

A.8. Details on prediction methods

Fig. A.10 provides a detailed view of the prediction method results for SOV_REFINE, where boxes represent the 25th to 75th percentiles and whiskers indicating values within 1.5 interquartile range (IQR). The spread reveals that even top-performing methods struggle with some proteins, achieving below 50% in SOV_REFINE.

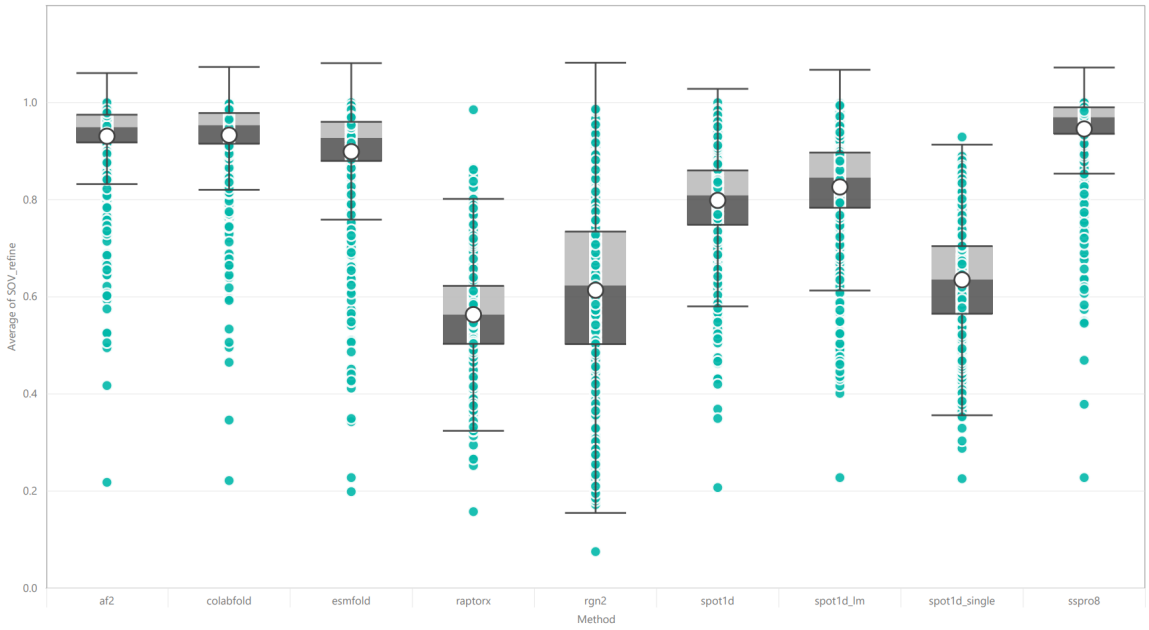


Figure A.10.|: **SOV_REFINE of each structure prediction method.** The boxes range between 25 and 75 percentiles, while the whiskers encompass 1.5 IQR. The white circles depicts each method's mean SOV_REFINE score.

A.9. Alphafold2 batch processing

The prediction of a 3D structure by Alphafold2 requires two main subprocesses. First, multiple sequence alignments (MSA) have to be done to obtain features that are utilized as an input for the following subprocess. Secondly, the neural network component takes the MSA features to produce and refine a predicted 3D structure of the target protein sequence. For long sequences, the first subprocess requires a high amount of RAM (∼128gb), an ordinary amount of CPU processing power (∼8 threads), and no GPU processor. The second subprocess requires an ordinary amount of RAM (∼32gb), a low amount of CPU processing power (∼4 threads), and a high amount of GPU processing power. The differing computing needs from the two subprocesses can be exploited by isolating them and computing them in separate clusters. This was done by ParaFold [Zhong et al., 2022] for use of high performance computing clusters and further refined with concurrent computing by members of the community. Unfortunately, the original project has not been updated with the community improvements. We have updated the project by fixing and

adding unfinished functionality for ease of use. Our improvements can be obtained at https://github.com/ivanpmartell/ParallelFold.

A.10. RGN2 local processing

RGN2 was originally made for Google Colaboratory [1]. This is a service that lets you run Python notebooks with a GPU on a server for free. We converted the Python notebook code into Python files for use in a local environment. The files include run 'run_aminobert.py' and 'run_rng2.py'. They are run successively starting with the aminobert script to obtain a tertiary structure prediction. The prediction is then run through an atomic relaxation (emrefinement [2]) software to remove any inconsistencies in the prediction. We noticed that the original relaxation script from RGN2 had an error which meant that some proteins were not relaxed and finalized to produce a prediction file. We fixed the error in the 'ter2pdb.py' file, which we also include. All files can be found under the 'extra/rgn2_local_files' folder in our main repository for this project.

A.11. Mut2dens model details

The model Mut2dens consists of extremely randomized trees trained using scikit-learn[3] with default hyper-parameter values. We utilized the default hyper-parameters for training the model. The input consists of nominal windowed data concatenated with a window length of 7, or window side length of 3. We utilized this window length as it was the best window length for top predictors. Therefore, the input data can be viewed as a vector of length $m = 7 \cdot p$, where p is the number of structure prediction methods utilized. The best model during testing utilized an input combination of SSPro8, ColabFold, and ESMFold. We also included Raptor-X Property and found a positive effect on the refinement of highly incorrect predictions. Thus, the final Mut2dens model makes use of four predictors ($p = 4$) and a window size of 28 ($m = 28$) for Mut2dens final model. This model slightly sacrified mean performance to increase the performance on inaccurate predictions.

Training the model takes less than 30 seconds using 8 cores from an Intel Xeon 2600 family CPU, while inference takes less than 3 seconds. Therefore, the majority of the time taken when utilizing this refinement procedure is spent obtaining predictions from the desired predictors. The time efficiency of the predictors is given below.

[1] https://colab.research.google.com/
[2] https://zhanggroup.org/ModRefiner/
[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

Predictor	Average time taken
Raptor-X Property	~ 1 minute
SPOT1D-Single	~ 5 minutes
ESMFold	~ 5 minutes
SPOT1D-LM	~ 5 minutes
RGN2	~ 5 minutes*
ColabFold	~ 20 minutes^{+}
AlphaFold2	~ 2 hours^{+}
SSPro8	~ 3 hours^{+}
SPOT1D	~ 4 hours^{+}

Table A.6.: * Time taken using a truncated amount of atomic relaxation. Without truncation, time could exceed 2 hours. $^{+}$ utilizes MSA procedure.

A.12. Predictors computational performance

To run the predictors, we utilize cloud resources which contain common server infrastructure CPU resources, such as the Intel Xeon 2600 family of processors. We utilized 16-core CPUs as a greater number of cores did not seem to improve performance significantly. When utilizing GPU resources, we utilized a single Nvidia V100. Per protein, with an average length of 300 amino acids, the average total time taken for each predictor is given in Table A.6

It is important to note that most of the time taken to predict proteins from these predictors come from their use of multiple sequence alignment or physics-based atomic relaxation techniques. These procedures can average around 60% to 80% of the predictor's time taken during prediction, which we include to account for all processing done after a protein sequence is inputted into the predictor.

A.13. Machine learning models

In this section we describe the architectural and hyper-parameter details of the multiple machine learning models we investigated. The models investigated include tree-type and neural-type models.

A.13.1. Tree-type model details

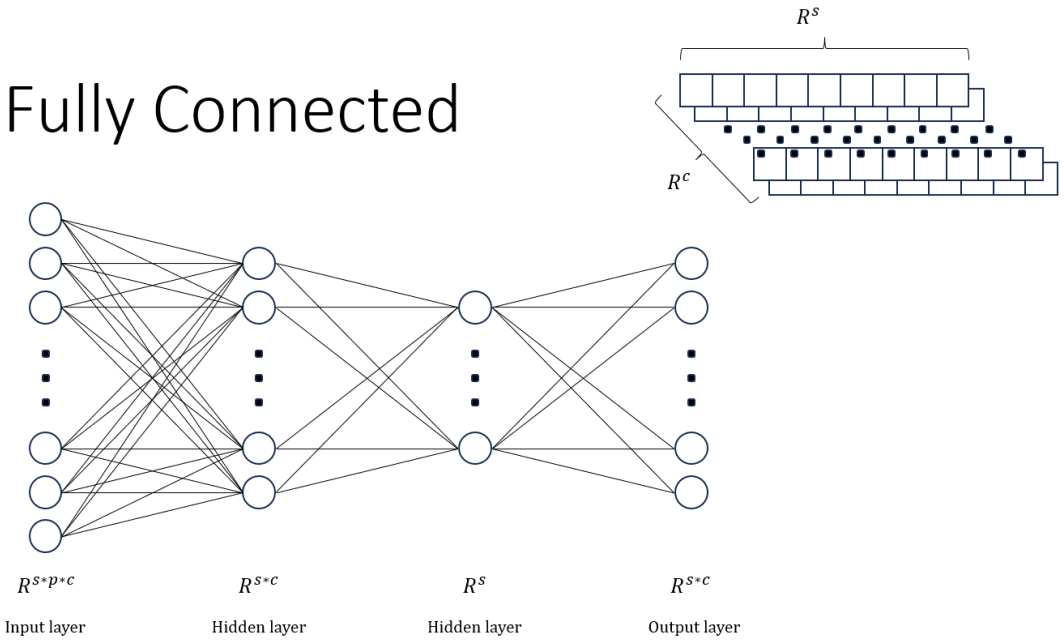
All tree-type models were created as classifiers from Scikit-learn. Their hyper-parameters follow their default values, which are shown in Table A.7.

Property	Decision Tree	Random Forest	Extra Trees
Split criterion	Gini Impurity	Gini Impurity	Gini Impurity
Trees	1	100	100
Minimum split samples	2	2	2
Minimum leaf samples	1	1	1
Features considered	All (M)	$\sqrt{M}$	$\sqrt{M}$
Bootstrap samples	No	No	No
Pruning	No	No	No
Depth	Unlimited	Unlimited	Unlimited

Table A.7.: **Tree-type model hyper-parameters.** All tree-type models were created as similar as possible. For the decision tree, all features had to be considered as only a single tree is created. Bootstrapping and pruning were not utilized to avoid removing any possible context from all predictors. Overfitting was only an issue after window lengths increased. If greater window lengths are required, utilizing these techniques could potentially alleviated overfitting.

A.13.2. Neural-type model details

All neural-type architectures were developed using PyTorch. The diagrams and details of the different architectures are given below.



s: maximum sequence length
p: amount of predictors
c: amount of secondary structure classes (Q8)

Figure A.11.|: **Fully-connected architecture.** Each layer of the network consists of a linear layer that connects all neurons to the next layer followed by batch normalization, a rectified linear unit (ReLU) activation function, and a 20% neuron dropout probability.

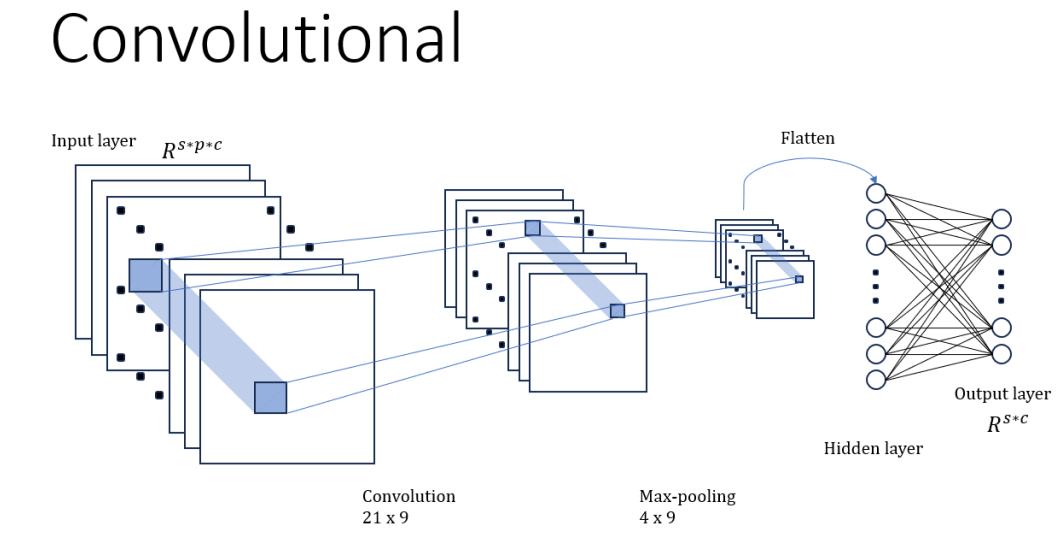


Figure A.12.|: **Convolutional architecture.** Each layer of the network consists of a convolutional layer with a halving number of channels starting from 64. Each layer also contains batch normalization, a ReLU activation function, and a 20% neuron dropout probability. The convolutions are then flattened into a vector and passed through a final linear layer as in the fully-connected architecture.

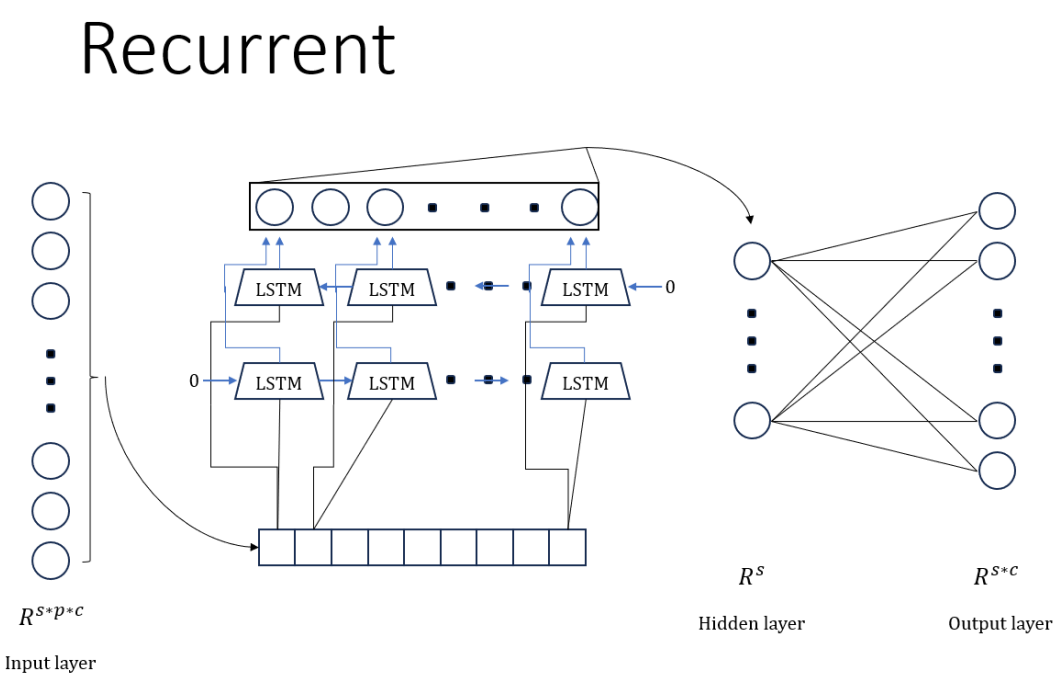


Figure A.13.|: **Recurrent architecture.** It consists of two long short-term memory (LSTM) layers that process the input in opposite directions, also known as a bidirectional LSTM. One processes the input from start to end, while the other from end to start. The LSTM layers also contain a 10% probability of neuron dropout. The LSTM layers' output is combined and passed through a final linear layer as in the previous architectures.

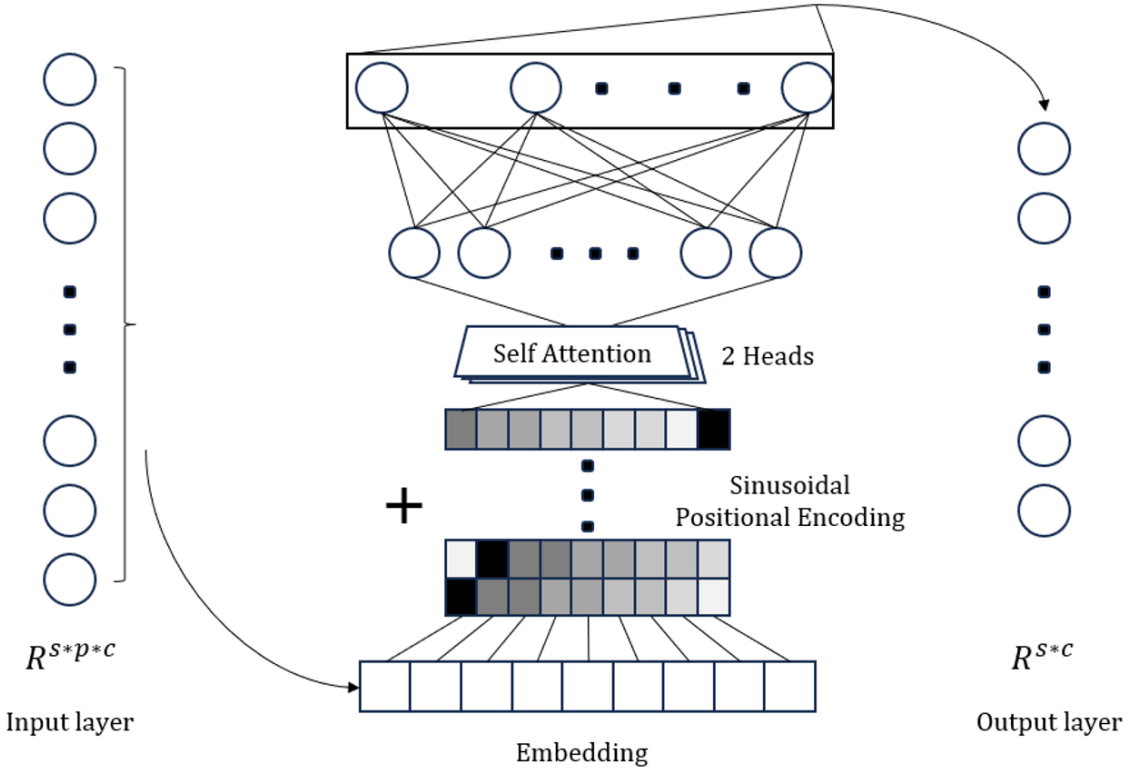


Figure A.14.|: **Transformer architecture.** It first embeds the input into vectors to be processed by positional encoding. This is then passed to two self-attention layers with 50% neuron dropout probability. The attention layer outputs are passed to intermediary fully-connected linear layers and then their outputs are combined and passed to a final linear layer as in the previous architectures.

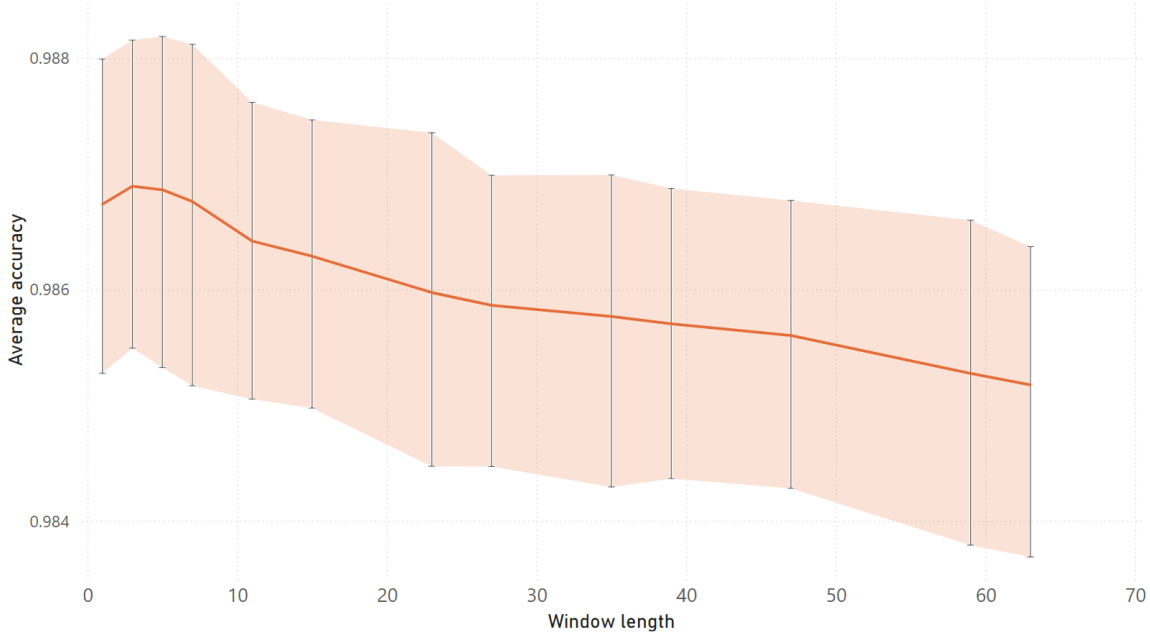


Figure A.15.|: **Training accuracy of top-performing predictors for different window lengths.** Diagram showing performance details for top-performing methods where an ExtraTree model has lower accuracy as the window length gets longer. This decrease might seem minuscule but it transfers remarkably well to test datasets, like CASP15. Within low window lengths, the performance of models increase until about a window length of 7. Afterwards, performance deteriorates quickly for unseen data. The reason for this is likely to be from overfitting the data with longer window lengths as the limited dataset provide a decreasing number of data points as the window size increases.

Appendix B

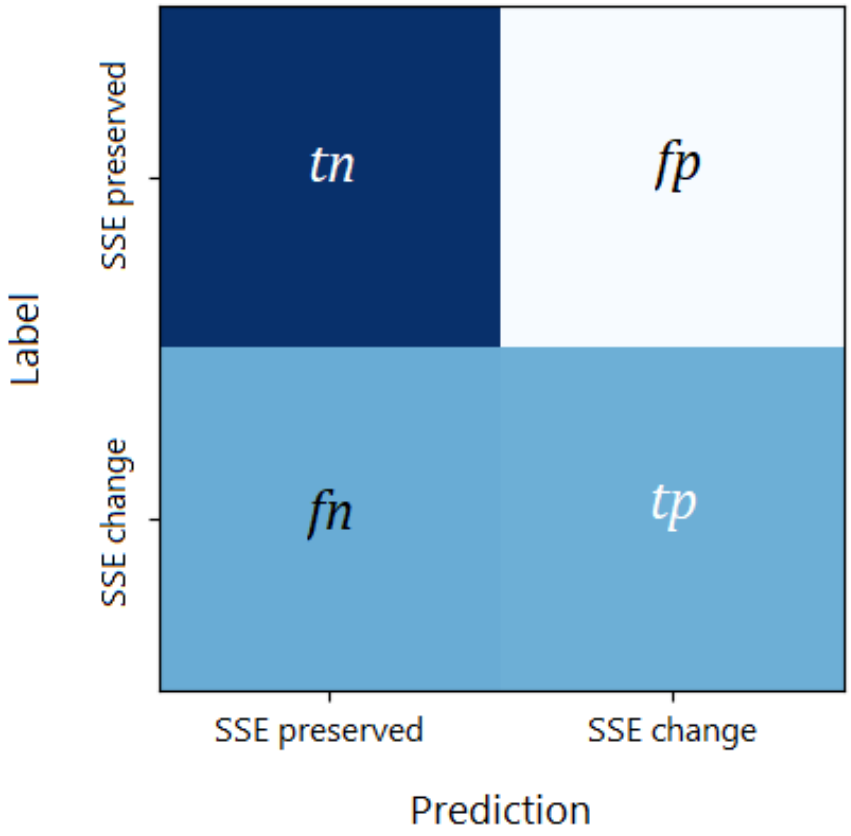
Binary classification measures

Choosing the proper measures for the evaluation of supervised machine learning models is a crucial step for testing the reliability of the models to perform the task that the model has been trained to perform. In our case, protein secondary structure can be represented analogously to the classification of SSE changes due to single amino acid mutations. For this simplified purpose, each amino acid can be classified in only two possible ways - SSE change or SSE preservation - meaning that secondary structure changes can be designed as a binary classification task.

Binary classification is a type of supervised learning in machine learning algorithms. This means that the algorithm must be supplied with a correct label for every sample of its dataset. For the following measures, we denote $y_i \in Y$ as the label for the i^{th} sample, and $\hat{y}_i \in \hat{Y}$ as the prediction coming from a machine learning model for that same sample i. Y and $\hat{Y}$ are the complete set of labels and predictions respectively for each amino acid in the proteins corresponding to all the dataset samples.

In the statistical sense, classification involves identifying the set of categories that a sample belongs in. In terms of supervised learning, a label is known and assumed to be given for each sample. When a model outputs a label for a sample, the algorithm that is training the model needs to know if the model made a mistake to correct it. The four possible scenarios that can happen in classification models are shown in Figure B.1. There are two scenarios involving correct classification and two involving incorrect classification. We refer to the indicator function to separate correct and incorrect classifications. The indicator function is defined as,

$$1(\text{condition}) := \begin{cases} 1 & if \text{condition} = true, \\ 0 & if \text{condition} = false \end{cases} \tag{B.1}$$



- The prediction is positive (SSE change) and matches the true label (SSE change), known as a true positive (tp).
- The prediction is negative (SSE preserved) and matches the false label (SSE preserved), known as a true negative (tn).
- The prediction is positive (SSE change) and does not match the true label (SSE change), known as a false positive (fp).
- The prediction is negative (SSE preserved) and does not match the false label (SSE preserved), known as a false negative (fn).

Figure B.1.|: **Binary classification scenarios.** List of outcomes that are applicable to binary classifications on secondary structure changes

B.1. Accuracy

Accuracy measures the number of correct predictions over the total number of predicted samples. The function used to compute the accuracy is the following,

$$accuracy(Y, \hat{Y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i) \tag{B.2}$$

where Y is the complete set of true labels and $\hat{Y}$ is the complete set of output labels. That is, $\hat{y}_i$ is the predicted label of the i-th sample and y_i is its corresponding true label over the total number of samples $n_{samples}$. We can also define accuracy in terms of statistical classification measures as,

$$accuracy(Y, \hat{Y}) = \frac{tp + tn}{tp + tn + fp + fn} \tag{B.3}$$

For easier visualization of these classification measures, we make use of confusion matrices. In a confusion matrix, an entry (i, j) corresponds to the number of observations in group i that are predicted to be in group j. An example can be seen in B.2.

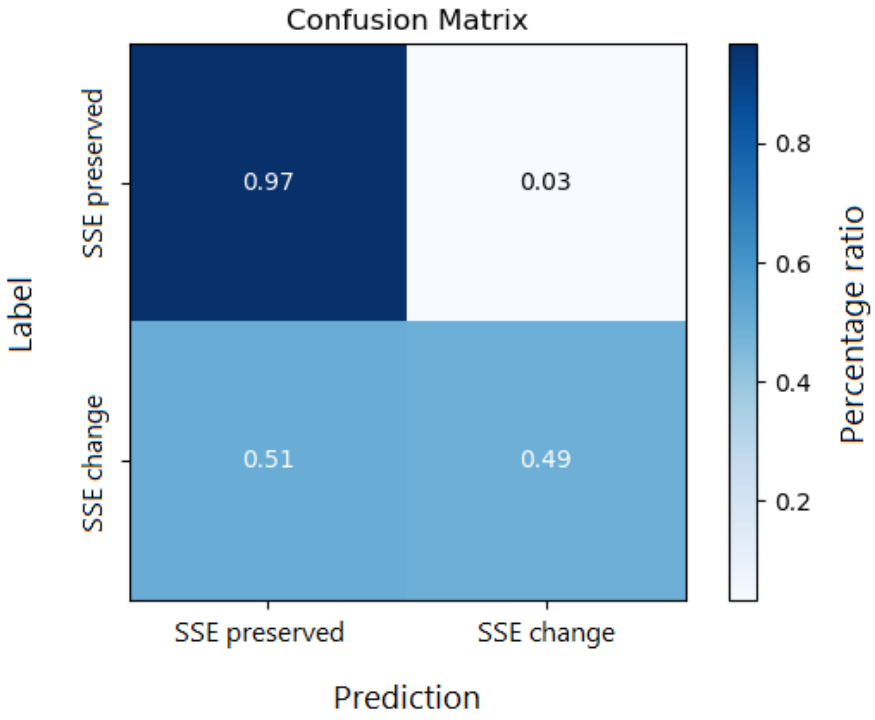


Figure B.2.|: **Confusion matrix.** Example showing potential values for true positives (0.49), true negatives (0.97), false positives (0.03), and false negatives (0.51)

B.2. Sensitivity

Sensitivity, also known as *True Positive Rate* (tpr), measures the proportion of SSE changes in the dataset that are correctly identified.

$$tpr = \frac{tp}{p} = \frac{tp}{tp + fn} \tag{B.4}$$

In Equation B.4, the positively classified samples p by the model are comprised by the tp and fn classifications.

B.3. Specificity

Specificity, also known as *True Negative Rate* (tnr), measures the proportion of preserved SSEs that are correctly identified.

$$tnr = \frac{tn}{n} = \frac{tn}{tn + fp} \tag{B.5}$$

In Equation B.5, the positively classified samples n by the model are comprised by the tn and fp classifications.

B.4. Precision

Precision, also known as *Positive Predictive Value* (ppv), is a measure of the classifier in terms of how relevant the results are. Precision decreases as the number of false positives increase.

$$ppv = precision = \frac{tp}{tp + fp} \tag{B.6}$$

B.5. False discovery rate

False discovery rate (fdr), is a measure of the proportion of false positives among all positive results. False discovery rate decreases as the precision increases.

$$fdr = \frac{fp}{tp + fp} = 1 - ppv \tag{B.7}$$

B.6. False negative rate

False negative rate (fnr), is a measure of the proportion of true positives that are incorrectly identified as negatives. False negative rate decreases sensitivity increases.

$$fnr = \frac{fn}{tp + fn} = 1 - tpr \tag{B.8}$$

B.7. Matthews correlation coefficient

Matthews correlation coefficient (mcc) is a measure of the quality of a classification even with high imbalances in class sizes. A coefficient of +1 represents a perfect prediction, 0 a random prediction and -1 an inverse prediction. It is defined as,

$$mcc = \frac{tp \cdot tn - fp \cdot fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \tag{B.9}$$