PROTESSA: A NEW METHOD FOR SECONDARY STRUCTURE ASSIGNMENT BASED ON TOPOLOGY

by

P. Ford Combs
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Bioinformatics and Computational Biology

Committee:

_____	Dr. Iosif Vaisman, Committee Chair
_____	Dr. Patrick Gillevet, Committee Member
_____	Dr. Dmitri Klimov, Committee Member
_____	Dr. Iosif Vaisman Director, School of Systems Biology
_____	Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science
_____	Dr. Fernando R. Miralles-Wilhelm, Dean, College of Science
Date: _____	Fall Semester 2021 George Mason University Fairfax, VA

Protessa: A New Method for Secondary Structure Assignment Based on Topology

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

by

P. Ford Combs
Master of Science
George Mason University, 2018
Bachelor of Arts
University of North Carolina – Chapel Hill, 2011

Director: Iosif Vaisman, Professor and Director,
School of Systems Biology

Fall Semester 2021
George Mason University
Fairfax, VA

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

Secondary Structure ..... SS
Secondary Structure Assignment ..... SSA
Helix ..... H
Strand ..... E
Coil C
Define Secondary Structure of Proteins ..... DSSP
Secondary Structural Identification ..... STRIDE
Hydrogen Bond ..... HB
Strongest Acceptor plus Bifurcation ..... SAB
Local Environment ..... LE
Delaunay Tessellation ..... DT
Coordinates of Alpha Carbons ..... CAC
Hydrogen Bond ..... HB
Voronoï Tessellation ..... VT
Machine Learning ..... ML
Three-dimensional 3D
Artificial neural network ..... ANN
Support Vector Machine ..... SVM
Random Forest ..... RF
Cross-validation CV
Uniform Manifold Approximation and Projection ..... UMAP

ABSTRACT

PROTESSA: A NEW METHOD FOR SECONDARY STRUCTURE ASSIGNMENT BASED ON TOPOLOGY

P. Ford Combs, Ph.D.

George Mason University, 2021

Dissertation Director: Dr. Iosif Vaisman

Secondary structure assignment (SSA) is the classification of each residue in a protein structure as helix, strand, or coil and, in this work, a new method for SSA is developed. SS is vital for stabilizing the overall structure and function of a protein; therefore, it plays a significant role in protein classification schemes, homology modeling, and structure comparison. It is also used to train secondary structure prediction methods, which try to determine secondary structure based on the amino acid sequence alone. The task of SSA is difficult because helices and strands in proteins rarely conform to their theoretical ideals. Most existing SSA methods rely on parameters, such as hydrogen-bond patterns or inter-atomic distances with arbitrary cutoffs. As a result, various SSA methods generate substantially differing assignments. ProTeSSA (Protein Tessellation-based Secondary Structure Assignment) is a new method that does not require parameters. It is based on the Delaunay tessellation (DT) of a protein's C-alpha coordinates (CAC). The DT of a

protein is a simplicial complex, where each residue is a member of a set of simplices, or tetrahedra, each forming a group of four natural neighbors. This topological data is mined to generate a descriptor for each residue, in part using a novel application of persistent homology.

Residue-based models were trained and tested on a test set of proteins, that was kept separate from training. The ProTeSSA models achieved greater than 85% accuracy at the residue level, using the protein structure author(s)'s assignments as ground truth, and low misclassification between helices and strands, less than 1 per test protein. A k-means cluster model was also developed and achieved high accuracy. Since the cluster model did no require training with SSAs from other methods, it is purely objective and provides a fascinating counterpoint to current SSA methods. The success of ProTeSSA indicates the potential to shift from parameter-based methods to an objective and consistent SSA method that relies solely on protein topology rather than parameters and cutoffs that stem from preconceived SS definitions.

INTRODUCTION

Proteins are complex biomolecules whose myriad functions range from transporting molecules across membranes or down microtubules to breaking substrate bonds and forming new compounds. These vital, life-giving functions are completely determined by the structure of a protein. When DNA is transcribed and translated into a chain of amino acids, the chain folds onto itself producing a three-dimensional structure whose chemical properties bestow its functionality. The links between sequence, structure, and function are strong and minor changes in sequence can dramatically alter structure and function.

The formation of secondary structures (SS) is an important, intermediate step in the protein folding process. These local, three-dimensional conformations of the protein backbone are formed by short-range interactions between residues. SSs directly influence the global structure and function of a protein by creating regions of local stability that are highly conserved across evolutionarily related proteins.

Importance of SSA

Research into SS can be traced back to two papers from 1951[1,2]. In the first, Pauling, Corey, and Branson, predicted two possible helical conformations of the protein backbone. One of these, known as the alpha-helix, is a type of SS formed when the

backbone winds in a right-handed spiral that is stabilized by a characteristic hydrogen-bond (HB) pattern formed between residues four positions apart.

In the second paper, Pauling and Corey described a dramatically different conformation called the pleated-sheet. In these structures, the backbone form crinkled planes in which individual segments, known as strands, running parallel or anti-parallel to others are stabilized by HBs.

In contrast to the relatively stable helix and strand conformations, the class of coil is given to regions of backbone that do not have discernible HB patterns or structures. This lack of stability allows coiled regions of the backbone to be comparatively flexible and is linked to lower evolutionary conservation, when compared to helices and strands. Though several other SS classes have been defined over the years through the identification of differences in HB patterns and other characteristics, the helix (H), strand (E), and coil (C) remain the three fundamental SS classifications.

It is fascinating to note that these predictions were essentially based on an understanding of what makes up a protein, *i.e.*, amino acids. By studying the chemical properties, such as bond angles and interatomic distances, the researchers were able to imagine these potential SSs. Later, as we were able to accurately determine the locations of atoms in a protein, using techniques like X-ray crystallography, the structures of Pauling, Corey, and Branson were found to exist.

Since the prediction and discovery of these local conformations, SS has played a critical role in many areas of protein research. For example, both CATH[3] and SCOPE [4,5], the two major databases of protein classification, use SS to classify proteins based

on their structural and evolutionary relationships. In both, the highest-level classification is based on whether the protein contains mostly alpha-helix, mostly beta-strands, or some mixture of the two. The importance of SS is also seen in protein homology modeling where a 3D structure of a protein is predicted from the sequence. One step of this process is searching for template models that best match an input sequence and SS is critical in scoring these templates[6]. Furthermore, one of the most common visualizations of protein 3D structure, known as the ribbon diagram, developed by Jane. S Richardson, is principally based on showing the protein backbone trace elaborated with simplified, idealized SS elements. An example is shown in Figure **1**.

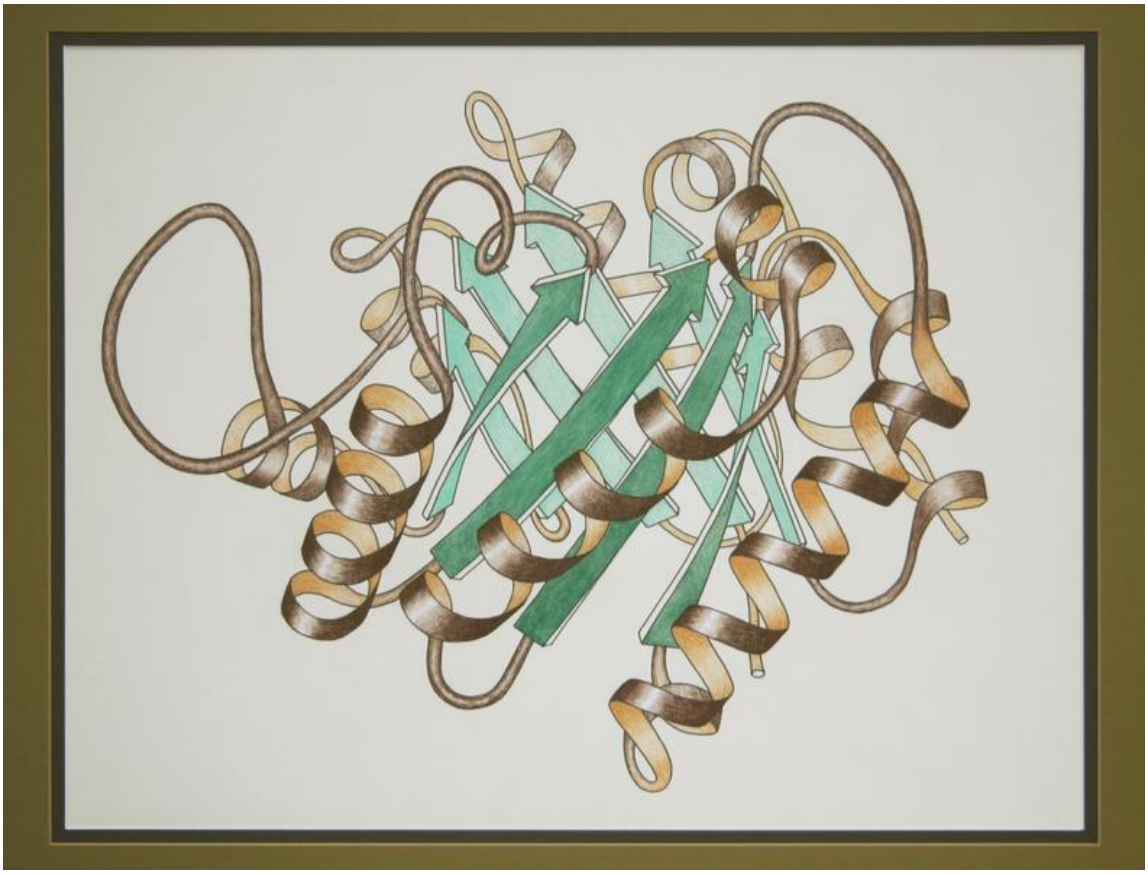

Figure 1. Ribbon drawing of Triosephophate isomerase by Jane S. Richardson. The helices are shown as wide-faced brown spirals and the beta-strands are shown as green arrows. The image is unaltered and shared freely under the Creative Commons Attribution 3.0 license: https://creativecommons.org/licenses/by/3.0

Secondary structure assignment (SSA) is the process of determining a residue's SS class by examining its position within the overall, three-dimensional (3D) protein structure. While there are different subtypes of helices and strands, in SSA it is common to use the simplified SS classification scheme of helix, strand, and coil. The simple ribbon diagram, as shown in Figure 1, demonstrates this three-class scheme with the strands and helices represented by arrows and wide, spiraling bands, respectively, and the coils are represented by a simple, tubular shape with irregular loops and twists. The study of protein structure and function, classification, homology modeling, and more rest on

our understanding of the interplay between the localized regions of stable SSs and the loose, flexible regions of coil, therefore, it is critical to have accurate methods of assigning secondary structure.

Current SSA Methods

The difficulty in performing accurate SSA stems from the fact the SS classifications are imposed by scientists onto the structures. We have decided that a helical structure stabilized by hydrogen bonds (HBs) at four residue intervals is an α-helix and that separate stretches of backbone that form parallel strands stabilized by regular HBs are β-sheets. But, nature does not stay within these bounds. Sometimes an α-helix will have an internal HB across a five-residue interval, creating a bump in the structure. Sometimes the regular pattern of HBs in a β-sheet is broken by the presence of an extra residue, causing a bulge. These irregularities lend themselves to human classification because humans are excellent at finding patterns and are not bound by the rigid rules of an algorithm. Therefore, trained scientists can identify SSs even when they vary from their theoretical ideals. However, there is a major flaw in the idea of relying solely on scientists to perform SSA; because protein structures are being solved at an increasing rate, more and more scientists would need to be trained to perform the task and, since different people have different intuitions and heuristics, there would be major disagreements in SSAs. An SSA algorithm solves the issues of scale and of variability in human decision-making because algorithms can be shared and used by the whole community and their rule-based structures are designed for consistency. The drawback of

using SSA algorithms, which is a direct result of their intrinsic consistency, is their rigidity and inability to deal with irregularities.

These challenges have led to the development of over twenty different methods, each of which show varying levels of agreement when compared with others[7]. The differences in assignments made by these methods stem from their different rule-systems, *i.e.*, the variation in their parameters and cutoffs. These disagreements are well known and have been characterized[7,8]. Though there is no definitive, best SSA method, Define Secondary Structure of Proteins (DSSP) has become the de facto method.

DSSP

The original DSSP algorithm described by Kabsch and Sander in 1983 is based on HB patterns[9]. A HB is assigned between the C=O and the N-H of two residues if the electrostatic interaction energy is less than −0.5 kcal/mol. This energy is based on the bond alignment angle and distance. Kabsch and Sander define an ideal HB as one with a distance of 2.9 Å between O and N, an alignment angle of 0º, and an energy of −3 kcal/mol. Coupling the cutoff of −0.5 kcal/mol with the energy calculation, allows for a HB to be assign at a maximum distance of 5.2 Å and a maximum alignment angle of 63º. HBs that fall outside of these limits, *e.g.*, those with alignment angles of 64º, are not identified by DSSP and are therefore not considered in the SSA process. This is an example of the inflexibility of algorithms.

After the initial step of finding all HBs in a protein structure, DSSP then looks for patterns. If a HB has been assigned between two residues, then the sequence distance between these residues is used to determine the potential SS. This potential SS is only

assigned if minimum length criteria are met. For example, if there is a HB between residue *i* and residue *i*+4, the SS is potentially a 4-turn, α-helix. If there is also a HB between residues *i*-1 and *i*+3, then the residues *i* through *i*+3 are assigned α-helix. The minimum length requirement for SSs and the interatomic distance and angle parameters for assigning HBs are key parameters of the DSSP method that differentiate it from other methods.

Because of DSSP's status as the gold standard in the field of secondary structure assignment, it is important to consider its complete SS classification system, which differs from the aforementioned helix, strand, and coil system. DSSP assigns eight types of SS: H for α-helix, B for isolated β-bridge, E for extended strand, G for 3_{10} helix, I for π-helix, T for turn, and S for bend. DSSP does not assign residues the label C for coil or random coil, instead it leaves a blank in the SSA results if a residue is not assigned any of the seven classes of SS. This blank is meant to be interpreted as loop or irregular element. Furthermore, the DSSP website (https://swift.cmbi.umcn.nl/gv/dssp/) explicitly states that it is wrong to replace these blanks with random coil. In spite of this, it is commonplace to replace these blanks with Cs. This eight-class SS assignment system, DSSP's original set of seven with C as an additional SS, is often reduced to a three-class system of H, E, and C, for helix, strand, and coil, respectively. There are several methods to map from the eight classes to three, but, unless explicitly stated otherwise, in this work, the map of the eight-class system to the three-class system is as follows: H, G, and I are assigned H; B and E are assigned E; and C, S, and T are assigned C, as shown in Equation 1.

Equation 1 Mapping from eight-class to three-class SSA scheme

$$M: SSA \begin{cases} H, & H, G, or\ I \\ E, & E\ or\ B \\ C, & C, S, or\ T \end{cases}$$

STRIDE

STRIDE, whose name comes from *secondary STRuctural IDEntification*, is arguably the second most used SSA method. In developing the STRIDE algorithm, Frishman and Argos sought to understand how crystallographers perform SSA so that they could build a better algorithm[10]. Given the previously mentioned idiosyncrasies in human SSA, it was no surprise that they found wide variation in the crystallographers' methods. Some emphasize torsion angles, while others emphasize hydrogen bonds, each group using different definitions, heuristics, and cutoffs. Some crystallographers use DSSP as an early step in their process and then perform visual evaluation to finalize their assignments.

Frishman and Argos wanted to improve DSSP by designing STRIDE to deal with the irregularities in SS. The title of the paper, *Knowledge-Based Protein Secondary Structure Assignment*, points to their use of crystallographer's techniques. They used this knowledge base to design and optimize a set of weights and thresholds built into their algorithm. In brief, the STRIDE process is as follows. In the first step, two probabilities are calculated for each residue: an α-helix probability and a β-sheet probability. These probabilities are 0 if the residue's torsion angles fall outside of the generally accepted ranges, as determined from the knowledge base. Otherwise, the probabilities are weighted based on the proportion of residues with similar angles that are assigned α-helix or β-

sheet in the PDB[11]. In other words, the α-helix probability tries to answer the question, "How likely is this residue in an α-helix given its torsion angles?" The calculation takes into account the number of residues in the PDB that have similar angles and are assigned α-helix out of the total number of residues with similar angles. These probabilities are weighted by values determined from the examination of crystallographer's assignments and then HB pattern criteria, similar to DSSP's, are applied. Finally, if the values falls in a certain range, which was also determined by examining crystallographer's SSAs, then the residue is assigned the appropriate helix or sheet classification[10].

In this way, STRIDE uses scientific knowledge in the determination of the thresholds, weights, and probabilities that guide its SSA process. Regarding accuracy, Frishman and Argos found that STRIDE outperformed DSSP in nearly 70% of the protein chains they examined[10]. These results suggest that this attempt to codify the human tendencies of crystallographers led to an improvement in SSA, but this came at the cost of the addition of several new parameters which were informed by scientists' preconceived notions of SS definitions.

SABLE

As described above, the tradeoff between consistency and generality in SSA stems from the divide between the human, adaptive decision-making method and the rigid rule systems used in algorithms. To improve the generality of DSSP, Haghighi, Higham, and Henchman removed the HB energy cutoff and experimented with different HB assignment methods, while maintaining DSSP's rules for assignment of SSs based on HB patterns[12]. They focused on strongest-acceptor methods to assign HBs, where the

strongest acceptor is defined as the acceptor with the most attractive electrostatic force. Of these methods, their preferred method was SABLE, which assigns HBs to the most favorable acceptor, but allows for bifurcation of the HB, *i.e.*, strongest acceptor plus bifurcation (SAB). They pair SAB with a local environment (LE) method to allow for unassigned donors, or amide group hydrogens whose most favorable acceptor is the carbonyl oxygen of the previous amino acid residue. In this way, SABLE identifies HBs without the use of DSSP's interatomic distance and energy cutoffs.

The authors found that SABLE agrees with DSSP 95% of the time, with the highest agreement, 97%, occurring for α-helices and β-strands and the lowest agreement, 69%, occurring for π-helices. While the authors suggest that SABLE could be improved by increasing the number of unassigned donor hydrogens, their results suggest that parameter-free HB assignment methods can be used in SSA. With the removal of the distance and energy cutoffs, SABLE is an important step in the direction of parameter-free SSA, though it still bases its assignments on DSSP's HB pattern definitions.

RaFoSA

The ProTeSSA method described in this work uses a random forest classification model [13,14]. It is trained on a set of features that includes topological information obtained from the Delaunay Tessellation (DT) of a protein's coordinates of alpha carbons (CAC). Of the more than twenty SSA methods in existence, two have important similarities to ProTeSSA: 1)RaFoSA[15] and 2)VoTap[16].

RaFoSA shares two similarities with ProTeSSA: 1) the model requires only CAC as input, and 2) the classification is performed by a random forest classifier. DSSP

requires an all-atom model of a protein structure in order to assign HBs and SSs and this is not always possible. Some structures have not been solved to that level and therefore an accurate method that requires only CAC is valuable. The RaFoSA random forest model uses a set of 30 features for each residue in a protein backbone to assign SS. These features include distances between neighboring alpha carbons, sign and angle of local torsion angles, and number of residue contacts at increasing distance cutoffs.

The model was trained and tested on DSSP labels. In testing, the full DSSP classification scheme was used as well as two 3-class mappings: 1) HBEGITS → hcsccc and 2) HBEGITS → hsshhcc, which is the same as Equation 1. The overall agreement between RaFoSA and the 7-class DSSP scheme was 93%. Using the first mapping, the agreement was 95.35% for H, 96.07% for E, and 96.72% for C. Using the second mapping, the agreement was 95.77% for H, 97.00% for E, and 92.13% for C [15]. Like SABLE, RaFoSA is another attempt to remove parameters from the SSA process.

VoTAP

The Voronoï Tessellation (VT) Assignment Procedure, known as VoTAP, is related to ProTeSSA because the VT is the dual of the DT used in ProTeSSA. A VT takes in a set of points and subdivides the space into regions around each point, where the edges of these regions are the midpoint between the central point and its neighboring points. These edges define regions called Voronoï cells. The cell around a given point includes all of the points whose Euclidean distances to the central point are less than or equal to the Euclidean distance to any other point. In VoTAP, the CAC are tessellated in a multi-step process. Because the Voronoï cells around points on the edge of the structure

extend infinitely, an extra layer of points is added around the CAC and progressively equilibrated. This generates the final VT, which includes a bounded cell for each residue.

Residues in the final VT whose cells share a face are considered to be in contact. The advantage of defining contacts by VT is that there is no need for a cutoff distance. The are some special considerations when regarding the beginning and end of a protein chain, but generally the VoTAP algorithm is as follows. A $n \times n$ matrix is built, where n is the length of the protein, in which each value represents the contact state for each pair of amino acids in the protein. This value is either 0, 1, or 2, for no contact, normal contact, or strong contact, respectively. To build the algorithm, the following procedure was performed for each residue in a set of 282 proteins. The contact values between residue *i* and residues *i*-6 through *i*-2 and those between residue *i* and residues *i*+6 through *i*+2 are used to create a 10-element string, the authors call a print. The contacts between residue *i* and residues *i*-1 and *i*+1 are not included, because they are nearly always the same. This print is then associated with the SSAs of residue *i*-2 through *i*+2, *i.e.*, each residue's print is associated with a quintuplet of SSAs, using the same 3-class system described in Equation 1. The total set of prints and associated quintuplets is then used to guide SSA. Three probabilities are calculated based on the print: 1) the probability that the residue is H, 2) the probably the residue is E, and 3) the probability that the residue is C. These probabilities are used to make a temporary SSA that is solidified through steps that remove very short SS elements and look for parallel and anti-parallel E elements.

On the test set, VoTAP SSAs agree with DSSP's 83.2%, overall. The agreements for H, E, and C elements were 93.0%, 77.3%, and 79.3%, respectively. VoTAP agreed slightly more often with STRIDE; the overall agreement was 84.4%, with agreements between H, E, and C elements of 96.7%, 79.1%, and 78.3%, respectively[16].

Delaunay Tessellation and SSA

The first mention of DT in protein structure analysis was in 1996 [17]. Since then, it has continued to be studied as a useful method for examining and understanding protein structure [18–23]. Taylor, Rivera, Wilson, and Vaisman introduced a new SSA method based on DT-derived topological data[19]. The DT will be discussed more below, but in general the DT of a set of points in 3-dimensions (3D) is a simplicial complex in which the points are connected by edges to form a set of tetrahedra, known as simplices. These tetrahedra conform to the criterion that no point other than the four vertices can fall within their circumspheres. In this way the DT of the CAC of a protein, can be thought of as the list of all groups of four natural neighbors, where each residue belongs to some number of tetrahedra and each tetrahedron defines a set of four neighbors.

To perform SSA using DT data, the authors of [19] developed a *t*-number for each residue. This number is based on the understanding that tetrahedra can be classified as one of five types with respect to the vertices' sequence distance: Type 1) no vertices are sequence neighbors, Type 2) two vertices are sequence neighbors and the other two are not, Type 3) two pairs of two vertices are sequence neighbors with a gap between the pairs, Type 4) three vertices are sequence neighbors, and Type 5) all four vertices are sequence neighbors. Since each residue is a member of multiple tetrahedra, the final *t*-

numbers of a given residue are the numbers of each type of tetrahedra of which it is a member. For example, if a residue is a member of 3 tetrahedra of Type 2, its *t*2 value is 3. Based on the 5 *t*-numbers of each residue, the authors built a 15-feature vector for each residue and trained a classification tree model that was able to perform SSA with sensitivity of 0.699, 0.849, and 0.917 for C, E, and H, respectively, and a specificity of 0.894, 0.885, and 0.948, for C, E, and H, respectively.

Significance

The descriptions of SSA methods above have illustrated that while DSSP is the de-facto standard SSA method, there is much room for improvement. STRIDE attempted to improve on DSSP's method by baking in scientific knowledge and SABLE tested parameter-free methods of assigning HBs. Furthermore, RaFoSA, VoTAP, and the *t*-number method all sought to use only CAC with varying success. These results point to the possibility of performing SSA based solely on CAC without using arbitrary cutoffs or parameters; this was the goal of developing the ProTeSSA method.

ProTeSSA is parameter-free and uses the DT of a protein's CAC to build a feature vector used in training a random forest classifier. There are several important aspects of this approach:

1. It is parameter-free, therefore it can be considered objective and can be used as an arbiter where other methods disagree.
2. The accuracy of ProTeSSA is in part of reflection of the consistency of whatever SSA method that was used to provide training classifications.

3. The feature vectors produced for ProTeSSA exist in a space that can be clustered to develop a completely objective SSA method that provides insight into the nature of SS.

The importance of developing a better SSA method lies in the numerous applications of SS. Since CATH and SCOPE rely on SS as a key feature in assigning evolutionary and structural relationships between proteins, different SSAs could reshape these databases and impact our understanding of these relationships. Furthermore, homology modelling programs, which use SS in scoring templates, and secondary structure prediction algorithms, which are trained on SSAs, could be improved by a better SSA method. These advances in our understanding of protein structure could have a broader influence as well. Because of the enormous variety of protein functions and the link between sequence, structure, and function, a better understanding of protein structure in general could lead to major advances in fields like biotechnology, biomedicine, and bioengineering.

Specific Aims

Construct DT-based SSA model

The first specific aim of this research was to build a viable SSA model. First, a data set of high resolution, X-ray protein structures was downloaded from the PDB [11]. This set was separated into a training set and a test set. A 64-feature descriptor based on the DT of the protein's CAC was built for each residue and various machine learning (ML) classification models were trained on DSSP, STRIDE, and the structure author(s)'

labels. These models were then tested on the test set of proteins, kept separate from training, to determine accuracy.

Improvement of the Model

The second specific aim of this work was to improve ProTeSSA. While the topological data explored in the first aim led to the development of an accurate classifier, this aim focused on the exploration of other potentially informative features that might improve the process. These new features included different distance functions and geometrical descriptions of the tetrahedra, such as edge ratio and aspect ratio. Importantly, these new features also included a novel application of persistent homology, applied here for the first time to protein secondary structure analysis.

Web Application Implementation

The final aim of this work was to develop a web application that applied the ProTeSSA method to a protein structure input by a user. The application needed two interfaces: 1) a simple interface designed for a one-click approach and 2) an advanced interface that allows the user more control. In addition to the application, a database of precomputed secondary structures was also provided.

PROTESSA: INITIAL MODEL VALIDATION

Introduction

The initial ProTeSSA model was built as an expansion of previous research into the DT of proteins [19]. In this work, the DT of the CAC of a protein's structure is mined to create a 64-attribute topological descriptor for each residue comprising sequence distance data for each simplex of which that residue is a vertex. The accuracy of this model shows that the DT can be used to build a parameter-free SSA method, which has important implications for the field of SSA research.

Materials and Methods

Delaunay Tessellation and *minifolds*

The DT of a set of points in any dimension is the simplicial complex in which no point is inside of the circum-hypersphere of any of the simplices [24]. The DT of a set of points in two dimensions, is formed by connecting points by edges to form a collection of triangles such that the circle that contains all three vertices of each triangle, known as a circumcircle, contains no other point. An example of a two-dimensional DT is shown in Figure 2. In the 3D world of proteins, the simplices that form the DT simplicial complex are tetrahedra, the sphere that contains the four vertices, known as the circumsphere, does not contain any point, and each simplex defines a set of four neighbors.

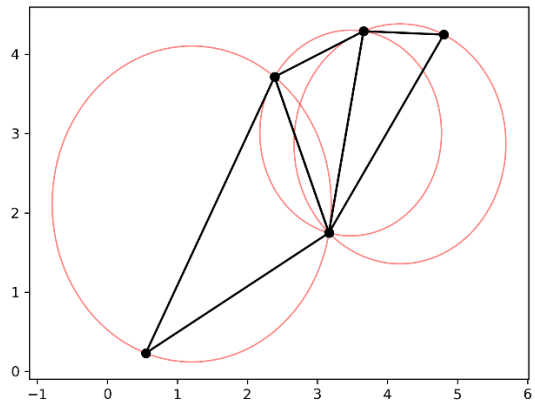

Figure 2 Delaunay tessellation of five, randomly selected points in two dimensions. The points and simplices are shown in black and the circumcircles are shown in red. Each simplex defines a set of three neighbors.

In this work, the CAC of a protein are tessellated. This results in a list of Delaunay simplices, or tetrahedra as described above, wherein each simplex is a set of 4 neighbors. Here, a *minifold* of a residue is defined as the complete set of tetrahedra of which it is a vertex. The number of tetrahedra in a residue's minifold varies, as seen in Figure 3, but the majority of minifolds contain between 5 and 15 tetrahedra, with residues classed as H or E tending to have more and those classed as C tending to have less.

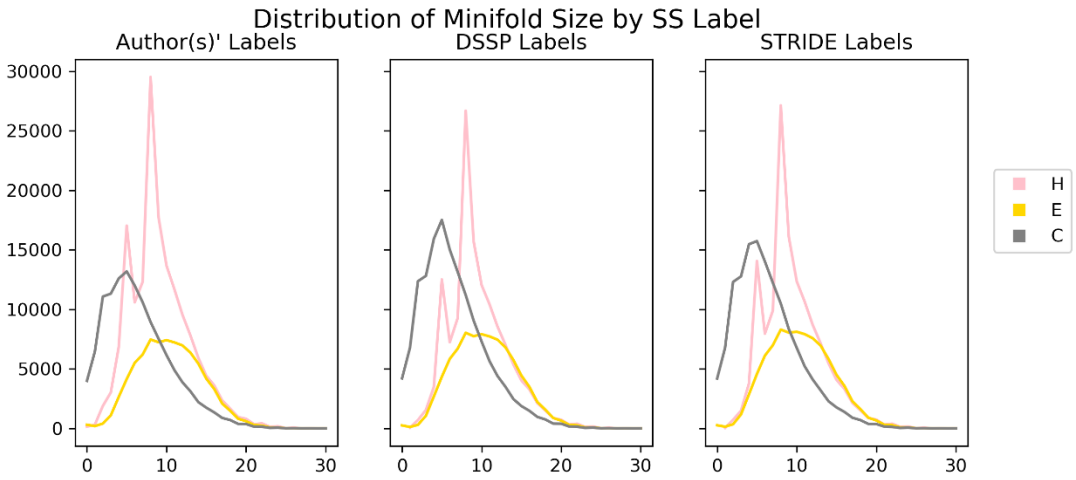

Figure 3 Distribution of minifold size, i.e., the number of tetrahedra in a minifold, for each H, E, and C SS class as defined by the structure author(s)', DSSP, and STRIDE.

Dataset and Topological Descriptor

Training and Testing Set

As of August 2021, the PDB contains over 180,000 protein structures and the PISCES web server regularly culls the database and generates lists according to different sets of parameters[11,25]. The dataset for this work was generated from a list of 4,705 proteins downloaded from PISCES. This set had two parameters: 1) a maximum 25% sequence identity was allowed between any pair of proteins in the set and 2) the resolution quality had to be equal or better than 1.6Å. Next the pdb file for each of these proteins was downloaded from the PDB and the chain identifier, as labeled in the PISCES list, was used to extract the data for the relevant chain. This data included the CAC, amino acid sequence, and the author(s)' SSA. Any protein chains that contained internal missing residue coordinates or whose length was less than 50 were removed,

resulting in a final dataset of 2,732 protein chains, 1,640 of which were used for training, and 1,092 of which were kept separate and used as the test set.

Topological Descriptor

To build a topological descriptor for each residue in a protein, first the DT of the CAC was determined. As described above, this results in a list of tetrahedral simplices, which are groups of four neighbors. Each residue's minifold is then determined and evaluated for sequence distance information, which is the basis for the topological descriptor of ProTeSSA.

For each simplex in a residue's minifold, the vertices are arranged into sequential order and three distances are calculated: 1) the sequence distance between the first and second residue, 2) the sequence distance between the second and third residue, and 3) the sequence distance between the third and fourth residue. The sequence distances can be quite large, *e.g.*, a simplex could have vertices corresponding to residues 1, 2, 3, and *n*, where *n* is the length of the protein, results in a sequence distance of *n*-3 between the third and fourth residues. To reduce the sequence distance space and simplify the data to aid in the ML training process, a transformation function, Equation 2, was applied.

Equation 2 Sequence distance transformation function.

$$T: d \begin{cases} 0, & if\ d = 1 \\ 1, & if\ d = [2-4] \\ 2, & if\ d = [5-9] \\ 3, & if\ d > 9 \end{cases}$$

The above transformation function maps the three sequence distances in each tetrahedra to a value in the range [0-4], thus limiting the number of types of simplices. Given that there are three sequence distances per tetrahedron and each distance can take any of 4 values, there are 4^3 or 64 different types of tetrahedra described by this function. One way to think about this is that the three sequence distances of a simplex can be thought of as x, y, and z coordinates in a 4×4×4 matrix, where each sequence distance value represents a position from 0-4 along one axis, and the value of a given cell in the matrix is equal to the number of simplices in the minifold corresponding to its coordinates. Another way to think about this is the number of each type of tetrahedron is stored in a 64-feature vector, where each index in the vector is associated with one type of tetrahedron and the value at the index is the number of that type of tetrahedron present in the minifold.

The transformation function was selected from a large set of transformation functions that were examined. The classification results for some of the other transformation functions can be seen in Table 1, notably the accuracy decreases as large distances get mapped to 0. This occurs because SSs are local regions of stability and as larger distances are mapped to 0, local topological information is lost. The above function was selected because of its emphasis on distinguishing local sequence distances.

Table 1 Random Forest and Support Vector Machine results for classification models built with ten-fold cross-validation. Abbreviations: CCR – Percent Correctly Classified Residues, TPH – True Positive Helix, TPE – True Positive Strand, TPC – True Positive Coil

Transformation Function	Random Forest Results	Support Vector Machine Results
$T: d \begin{Bmatrix} 0, & if\ d = 1 \\ 1, & if\ d = 2 \\ 2, & if\ d = 3 \\ 3, & if\ d > 3 \end{Bmatrix}$	CCR: 83.5% TPH: 0.916 TPE: 0.854 TPC: 0.735	CCR: 83.2% TPH: 0.906 TPE: 0.836 TPC: 0.755
$T: d \begin{Bmatrix} 0, & if\ d = 1 \\ 1, & if\ d = [2,3] \\ 2, & if\ d = [4,5] \\ 3, & if\ d > 5 \end{Bmatrix}$	CCR: 83.1% TPH: 0.907 TPE: 0.844 TPC: 0.743	CCR: 82.2% TPH: 0.885 TPE: 0.827 TPC: 0.754
$T: d \begin{Bmatrix} 0, & if\ d = [1,2] \\ 1, & if\ d = [3,4] \\ 2, & if\ d = [5,6] \\ 3, & if\ d > 6 \end{Bmatrix}$	CCR: 82.0% TPH: 0.905 TPE: 0.849 TPC: 0.704	CCR: 81.3% TPH: 0.901 TPE: 0.834 TPC: 0.705
$T: d \begin{Bmatrix} 0, & if\ d = [1-3] \\ 1, & if\ d = [4-6] \\ 2, & if\ d = [7-9] \\ 3, & if\ d > 9 \end{Bmatrix}$	CCR: 79.0% TPH: 0.886 TPE: 0.842 TPC: 0.641	CCR: 79.1% TPH: 0.883 TPE: 0.814 TPC: 0.677
$T: d \begin{Bmatrix} 0, & if\ d = [1-4] \\ 1, & if\ d = [5-8] \\ 2, & if\ d = [9-12] \\ 3, & if\ d > 12 \end{Bmatrix}$	CCR: 76.0% TPH: 0.876 TPE: 0.828 TPC: 0.574	CCR: 75.7% TPH: 0.796 TPE: 0.803 TPC: 0.796
$T: d \begin{Bmatrix} 0, & if\ d = [1-5] \\ 1, & if\ d = [6-10] \\ 2, & if\ d = [11-15] \\ 3, & if\ d > 15 \end{Bmatrix}$	CCR: 74.5% TPH: 0.851 TPE: 0.814 TPC: 0.570	CCR: 73.4% TPH: 0.774 TPE: 0.777 TPC: 0.652
$T: d \begin{Bmatrix} 0, & if\ d = [1-7] \\ 1, & if\ d = [8-16] \\ 2, & if\ d = [17-26] \\ 3, & if\ d > 26 \end{Bmatrix}$	CCR: 71.2% TPH: 0.820 TPE: 0.760 TPC: 0.555	CCR: 70.0% TPH: 0.768 TPE: 0.711 TPC: 0.622

While the transformation function converts the data into a manageable space, there is another aspect of the DT that warranted consideration. In this work, the edges in

the tessellation correspond to neighbors, but points on the outside of the structure can be connected by very long edges, well beyond the length of a meaningful biological connection. The long edges were thought to be a potential source of noise in the data, since they define neighbor relationships that are unlikely to be involved in local phenomenon of SS. Therefore, two distance cutoffs were used to remove these long edges from the DT: 1) 10 Å and 2) 8 Å. Any simplices that contained an edge length greater than the cutoff value, was removed from the tessellation. These two pruned tessellations combined with the complete DT were used to build three datasets of topological descriptors for each residue.

SS Classification Labels

The parameter free nature of ProTeSSA sets it apart from many other SSA methods. The transformation function and edge-length cutoff described above are not parameters in the since of other methods because they simply filter the data, rather than define allowable angles, HB distances, or other metrics associated with different SSs. This means that ProTeSSA does not attempt to define SS element *a priori*. Therefore, it is necessary to provide classification labels in training so that the model can learn and this has interesting implications. The accuracy of the ProTeSSA model is determined by the model's ability to capture SS information and the consistency of the method on which it was trained. This means that ProTeSSA is uniquely qualified for comparing other SSA methods.

In this work, classification training labels for each protein came from the author(s)' assignments, DSSP, and STRIDE. This resulted in a total of nine datasets built

using the three sets of classification labels for each protein paired with the three sets of topological descriptors for each residue in each protein, where the three sets of descriptors come from the three tessellations produced by either applying no cutoff, the 10 Å cutoff, or the 8 Å cutoff.

Classification Model Background

In order to determine the type of ML model to train and test, five different types of classification models were examined: naïve Bayes, neural network, support vector machine, decision tree, and random forest. These were chosen because of their marked differences in approach. A naïve Bayes classifier is a probabilistic model that is based on Bayes' theorem and assumes independence among the features. It is one of the simplest ML models which states the probability that a given set of features belonging to a given class equals the likelihood of each feature belonging to that class multiplied by the prior probability of that class divided by the prior probabilities of each of the features. Since, the product of the prior probabilities of each feature is constant, Equation 3, is often used to calculate the proportional probability for each class, of which the highest is chosen to be the prediction.

Equation 3 Naïve Bayes classifier equation to compute probability that a set of features X belongs to class y.

$$P(y \mid x_1 \ldots x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

Artificial neural networks (ANN) are extremely versatile models that have been used in many areas of computer science. There are many forms of ANNs, but they all consist of a network of artificial neurons, which take in some number of input values, converts it via a function, and produces an output. In its simplest form, an ANN has three layers: an input layer, a hidden layer, and an output layer, where the input layer contains the vector of features, the hidden layer contains some number of weighted functions, and the output layer is a vector of the length of the number of classes. The weights in the hidden layer are randomized upon instantiation, but as the model is trained on a set of features and classes, these weights are changed incrementally until they produce the optimal classification output. This kind of ANN is show in Figure 4, but more complex architectures can contain many hidden layers of different sizes in which the artificial neurons have varying functions.

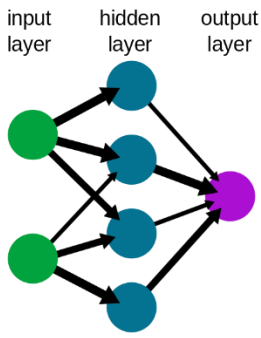

Figure 4 Simple ANN with two inputs, one hidden layer, and a single output value.

Binary support vector machines (SVMs) seek to find the hyperplane with the widest margin that passes through the feature space and separates the samples into their

respective classes. Mathematically, this is equivalent to finding the vector of weights whose dot product with the feature vector is greater than 1 or less than -1, depending on the class.

Equation 4 Support vector machine hyperplane equation.

$$w^T x - b = 0$$

The hyperplane equation, Equation 4, often includes an intercept b. In ML models, this intercept is often included by extending the weight vector by 1 and adding an additional feature of value 1 for each sample. Thus, in training the model, this extra weight becomes the intercept. While the above explanation works for linearly separable data, it does not hold for data that is not linearly separable. In those cases, the hinge loss function can be used to find the hyperplane that maximizes the number of correctly classified samples. Extending binary SVMs into multiclass situations is accomplished by using methods like a one vs. one or a one vs. rest approach, where binary SVMs are built to separate each pair of classes or to separate each class from all other classes, respectively.

A decision tree model is a tree like model in which a sample is classified by moving from the root node down through a series of binary decisions until a leaf node is reached. Given a sample with a set of features, at each internal node the decision tree makes its classification by examining the value of one of the features and moving to its left or right node based on the value of that feature. This process repeats until a leaf is

reached and the sample is classified. At each level in training a decision tree model, the feature that provides the highest information gain, *i.e.*, the feature that splits the data into the appropriate classes with the greatest accuracy, is chosen. Samples that have a value below the splitting value for that feature are sent to one node, while sample that have a value above the splitting value for that feature are sent to the other node. In this way, a decision tree can be trained to classify data.

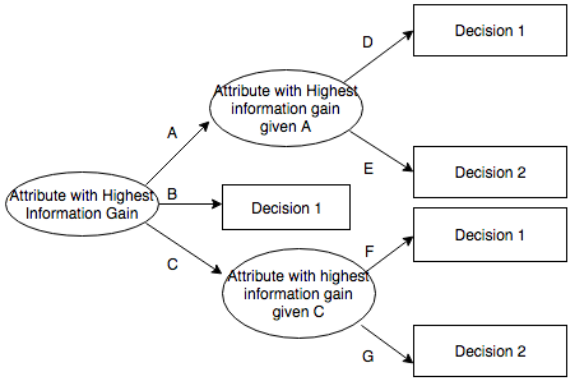

Figure 5 Decision Tree Framework by Acoggins38. It is unchanged and shared under the following license: https://creativecommons.org/licenses/by-sa/4.0/deed.en

A random forest (RF) model is an ensemble ML method; it is a combination of several ML models, in this case many decision trees, that are combined to produce a single output. In the previously described decision tree model, the number of nodes from root to leaf is known as the depth. As decision trees becomes deeper, their bias tends to shrink, *i.e.*, they make fewer and fewer errors on the training data, which often leads to inflexibility when classifying new data. This is known as overfitting. RFs circumvent

overfitting by combining many decision trees, often around 100, with some randomness. The randomness can come from taking random subsets of the training data during training or from taking random subsets of features when building the trees, but the result is always a set of decision trees with markedly different characteristics. By averaging across the predictions or taking the majority vote, RFs can make a classification that leverages the bias of deep decision trees while maintaining generalizability.

Model Comparison Results

To examine which of the above ML models would be best suited to SSA, cross-validation (CV) of each type of model was performed. In CV, the training set is broken up in *k* smaller sets, called folds, and the model is trained on *k*-1 of the sets and tested on the remaining set. This process is repeated *k* times and the average accuracy represents the performance of the model. The advantage of CV is that the model can be examined and the hyperparameters can be tuned without using the test set. This helps prevent overfitting the model because, without CV, continually testing and adjusting the hyperparameters would result in tuning the model to the test set. With CV, the model can be tuned to achieve better accuracy on the training data without using the test data. The test data is therefore a true test of the model's performance because it did not influence the training procedure.

To select the best type of ML model, 10-fold CV was performed for each type of ML model on the full DT data using the author(s)' labels with training sets of size 300; 600; 1,200; 3,000; 6,000; 12,000; 21,000; and 30,000, where each set contained equal amounts of residues for each class of H, E, and C structure. Figure 6 shows the results of

10-fold CV of each type of model on varying training set size. At the smallest training set sizes there is variability in the average CV accuracy, but as the set size increases a curve develops that flattens for each model between the training set sizes of 6,000 and 12,000. The RF model was the best performing model, slightly, but consistently outperforming the other models.

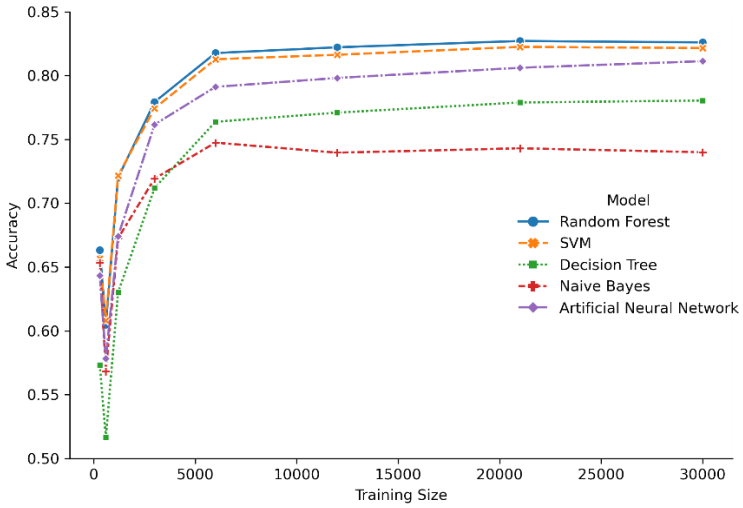

Figure 6 ML model type comparison using 10-fold CV. The x-axis is the number of residues in the training set and the y-axis is the average accuracy.

To further test whether the topological descriptor was capturing SS information, the same procedure as above was performed except the training labels were randomly switched so that only 33% were correctly labeled and 66% were misclassified. With only one-third of the labels correct, ML models are expected to achieve a maximum accuracy of about 33%. Figure 7 shows the results of 10-fold CV on the different sizes of training sets with randomized labels. As expected, the models achieved very low accuracy

ranging from about 15% to 35% depending on the model type and training set size. This indicates that the topological descriptor is capturing SS information and the models trained on correct labels are learning how to perform SSA.

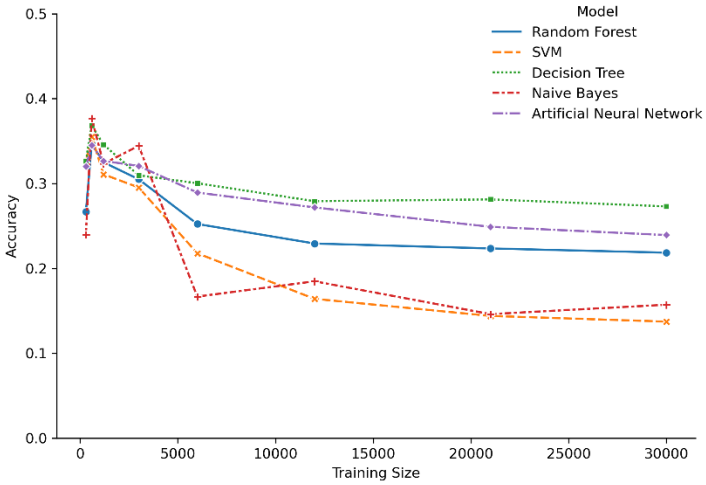

Figure 7 ML model type comparison using 10-fold CV with randomized labels. The x-axis is the number of residues in the training set and the y-axis is the average accuracy.

Given the results shown in Figure 6, a RF model and residue-based training set size of 12,000 with equal parts H, E, and C were selected for further analysis and 9 RF models were trained: 1) Author(s)' labels used in training on DT topological data with no simplices removed, 2) Author(s)' labels used in training on DT topological data with simplices with an edge longer than 10 Å removed, and 3) Author(s)' labels used in training on DT topological data with simplices with an edge longer than 8 Å removed, 4) DSSP labels used in training on DT topological data with no simplices removed, 5) DSSP labels used in training on DT topological data with simplices with an edge longer than 10 Å removed, 6) DSSP labels used in training on DT topological data with simplices with

an edge longer than 8 Å removed, 7) STRIDE labels used in training on DT topological data with no simplices removed, 8) STRIDE labels used in training on DT topological data with simplices with an edge longer than 10 Å removed, and 9) STRIDE labels used in training on DT topological data with simplices with an edge longer than 8 Å removed. In testing, the SS labels from the same source as was used in training were used as ground truth, *e.g.*, the 3 models trained on the author(s)' SSAs were tested using the author(s)' SSAs for the proteins in the test set.

Results

Figure 8 shows boxplots of the distributions of classification accuracy per protein in the test set for each of the 9 models with each median shown in white. Statistical significance, as determined by the non-parametric Mann-Whitney *U* test, is also displayed in this figure with horizontal brackets indicating the pairs of distributions that were tested and asterisks indicating significance, with increasing numbers of asterisks indicating lower p-values.

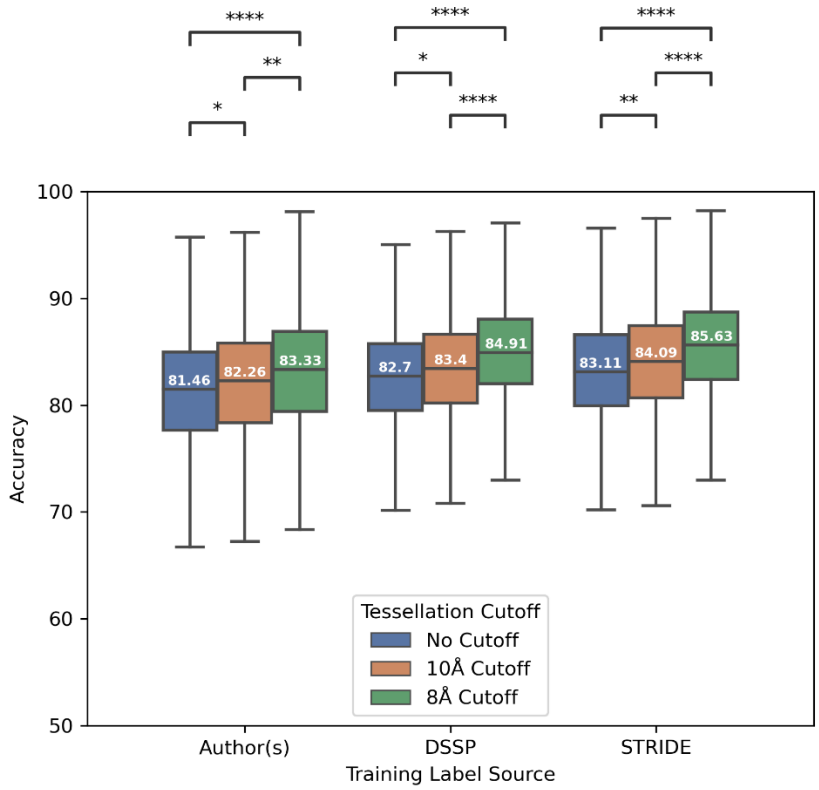

Figure 8 Boxplots, with medians shown in white, of the distributions of classification accuracy per protein in the test set for each of the nine RF models trained on the ProTeSSA data: 1) Author(s)' labels used in training on DT topological data with no simplices removed, 2) Author(s)' labels used in training on DT topological data with simplices with an edge longer than 10 Å removed, and 3) Author(s)' labels used in training on DT topological data with simplices with an edge longer than 8 Å removed, 4) DSSP labels used in training on DT topological data with no simplices removed, 5) DSSP labels used in training on DT topological data with simplices with an edge longer than 10 Å removed, 6) DSSP labels used in training on DT topological data with simplices with an edge longer than 8 Å removed, 7) STRIDE labels used in training on DT topological data with no simplices removed, 8) STRIDE labels used in training on DT topological data with simplices with an edge longer than 10 Å removed, and 9) STRIDE labels used in training on DT topological data with simplices with an edge longer than 8 Å removed. Statistical significance is shown above the chart. * indicates a p-value <= 0.05, ** indicates a p-value <= 0.01, * indicates a p-value <= 0.001, and **** indicates a p-value <= 0.0001. Outliers were not shown in this figure.**

The confusion matrices shown in Figure 9 show how the predicted labels and true labels compare for all of the residues in the test set. The model trained on the author(s)' SSA had the high true positive helix value and the lowest true positive strand and coil values. The model trained on DSSP labels had the highest true positive coil value and the STRIDE model had the highest true positive strand value.

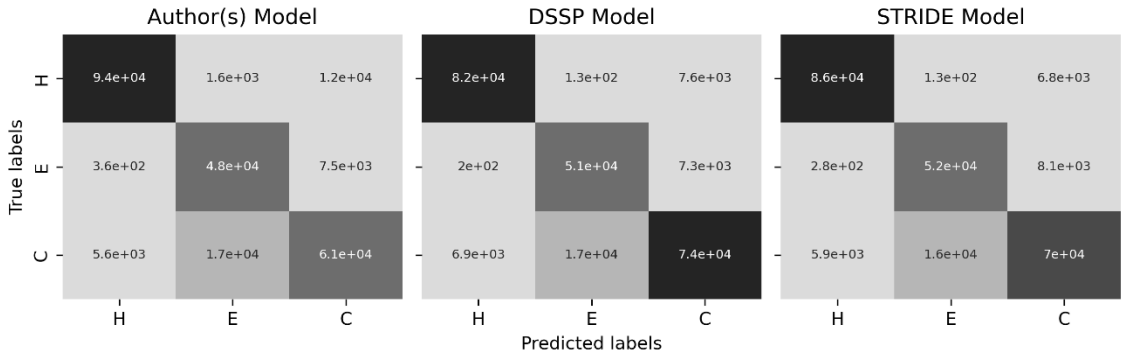

Figure 9 Confusion Matrices for the three 8 Å edge-length cutoff models. The left matrix is for the model trained on the structure author(s)' SSA where ground truth is the structure author(s)' labels for the test set. The middle is for the model trained on DSSP where ground truth is the DSSP labels for the test set. The right is for the model trained on STRIDE where ground truth is the STRIDE labels for the test set. The darker the cell's color, the higher the value.

Figure 10 shows the average true positive helix classification and the average true positive strand classification per protein for each of the models. The true positive strand accuracy increased as the edge-length cutoffs were applied with the greatest value of 85.1% achieved by the model trained and tested on DSSP SSAs with an edge-length cutoff of 8 Å applied to the DTs. The true positive helix accuracy also increased with the edge-length cutoff, though not as dramatically as the true positive strand accuracy, with the highest value achieved by the model trained and tested on STRIDE SSAs with an edge-length cutoff of 8 Å applied to the DTs. The lowest values in both cases came from the models trained on the author(s)' SSAs with no edge-length cutoff applied to the DTs.

Figure 10 True positive helix and true positive strand average accuracy per protein in the test set for each of the nine models.

Figure 11 shows the misclassification of helices as strands and strands as helices for each of the models. The lowest value for misclassification of helix as strand was 0.2% per protein and was achieved by model trained and tested on both DSSP and STRIDE. The lowest value for misclassification of strand as helix was 0.5% per protein and was achieved by two of the models trained and tested on DSSP. In both cases, the highest value for misclassifications came from models trained and tested on the author(s)' SSAs, though the edge-length cutoff increased helix misclassification while decreasing strand classifications.

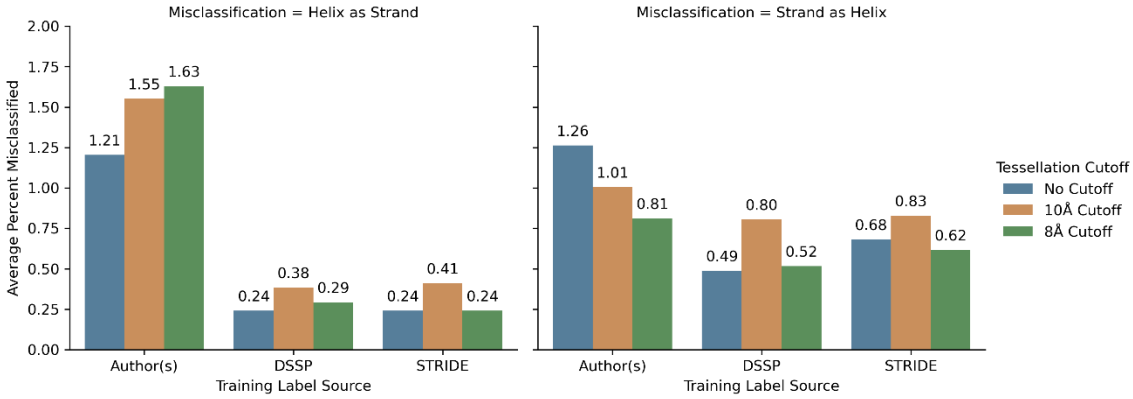

Figure 11 Average misclassifications of helix as strand and strand as helix per protein in the test set for each of the nine models.

Several key insights can be gleaned from Figure 8. Regardless of the training label source, there is a strong, statistically significant (p-value <= 0.0001, shown by '****') increase in accuracy between using the full DT of a protein to build the topological descriptor and using an 8 Å edge-length cutoff to prune the DT beforehand. This is not surprising as the DT is used to define groups of four neighbors in the protein structure and long edges that link remote residues are unlikely to represent meaningful biological connections. The whiskers in the plots also show a trend. The whiskers span the range from first quartile minus 1.5 times the interquartile range to the third quartile plus 1.5 times the interquartile range. This range represents the minimum and maximum of the distributions adjusted to remove outliers. (Note that outliers were not displayed in this figure and are discussed later.) Figure 8 shows that the models trained on the author(s)' labels have a consistently wider range than the models trained on DSSP or STRIDE, which have similar ranges. For the 8 Å edge-length cutoff models, the lower and upper whisker tips for the models trained on the author(s)' SSA, DSSP, and STRIDE

were 68.24% and 98.01%, 72.88% and 97.17%, and 72.96% and 98.17%, respectively. The 8 Å DSSP model had the narrowest range, 24.29%. This is an indication that the DSSP and STRIDE SSA algorithms are more consistent that the author(s)' SSA method. This is unsurprising because of the previously described tendencies of humans to use flexible and idiosyncratic methods to perform SSA. For all models, the median of the average per-protein accuracy was greater than 80%, but the best performing models were those trained on 8 Å edge-length cutoff data. The models trained on the author(s)', DSSP, and STRIDE SSAs achieved median per-protein average accuracy of 83.33%, 84.91%, and 85.63%, respectively, as shown in Figure 8.

In addition to accuracy, two other important indicators of SSA performance relate specifically to helices and strands. Because helices and strands are more ordered regions of a protein structure and are less variable than coils, the ability of a method to distinguish between helices and strands is a good measure of accuracy. One metric, is the true positive classification. Per-protein, all of the models correctly labeled helices over 80% of the time, with the 8 Å edge-length cutoff STRIDE model achieving the highest accuracy of 88%. The true positive strand accuracy was greater than 78% for all of the models, with the 8 Å edge-length cutoff DSSP model achieving the highest level of 85.1%. Interestingly, the true positive strand classification rate increased more dramatically as the edge-length cutoff increased for all the models, while this increase was less pronounced in the true positive helix classification rate.

It is also important to look at misclassification of helices as strands and strands as helices. These two SSs have very different ideal structures and therefore an accurate SSA

method should not confuse them often. While misclassifications of this type occurred on average for less than 2% of the residues in a protein, there are some notable trends shown in Figure 11. Unlike the true positive rates and overall accuracy, the edge-length cutoff did not produce a consistent effect on misclassifications. In the models trained on author(s)' SSAs, misclassifications of helices as strands actually increased as the cutoff was applied. For the DSSP and STRIDE models, the 10 Å edge-length cutoff led to an increase in misclassifications and, while the 8 Å cutoff did reduce misclassification, it only outperformed the no cutoff models in the STRIDE model.

Outliers

To examine where the SSA models were least accurate, the set of low-valued outliers from the results for the three 8 Å cutoff were examined. There were 57 protein chains present in the complete set and 17 of them were shared by the three models: 5LW3A, 4TKCA, 1ISUA, 6MYID, 4DT5A, 6CNWA, 6QPSA, 3KWEA, 5L2LA, 4CP6A, 6ITGA, 5OLRA, 5YXMA, 3D9XA, 3PMOA, 1EZGA, and 6BXDA, where the first four letters and numbers represent the protein ID and the final letter indicates the chain. Many of these protein chains contain many strands and often these strands form a long barrel. For example, protein 3KWE chain A, shown in Figure 12, displays this characteristic. Proteins 1ISU chain A and 5L2LA show a different characteristic of containing very small amounts of helix or strands structures, rather being mostly coil. Only one outlier protein chain, 3PMOA, contained mainly helices.

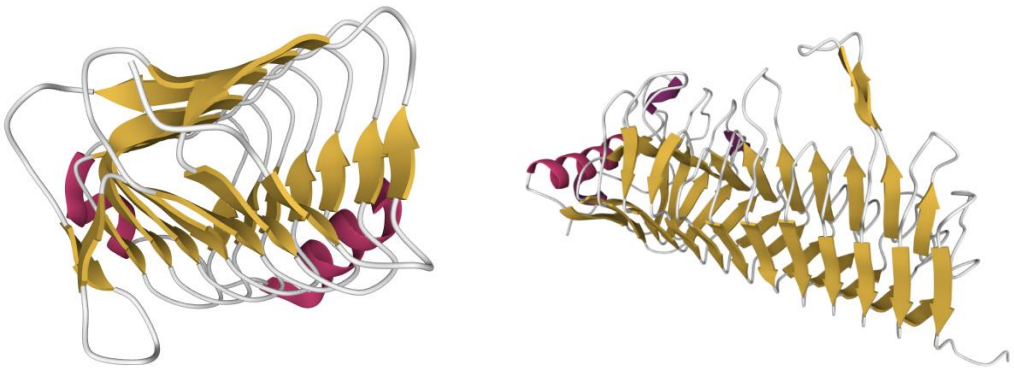

Figure 12 The structures of protein 3KWE chain A, left, and protein 5LW3 (solved by U. Rothweiler, literature *to be published*) chain A, right, show the many strand and barrel trait[11,26,27]**.**

ProTeSSA as an arbiter

The results show that the RF models agree with the SSA methods on which they were trained over 80% of the time on average. It is interesting to examine cases where the models disagree with the training label source, because the model's performance is based on the training labels source's information. ProTeSSA has no definition of SS *a priori* and there its accuracy is a reflection of the consistency of the training label source method, *i.e.*, the author(s)', DSSP's, and STRIDE's SSAs.

One example of a disagreement occurs in the 8 Å cutoff models' assignments for protein 3A99 chain A, shown in Table 2. The author(s)', DSSP, and STRIDE all labels the first 5 residues as C, but each of the models label residues 2-5 as H and only the first residue as C. All SSAs agree for residues 5-10. Figure 13 shows the structure of the cartoon representation of first 7 residues of this protein and residues 2-5 do seem to form a helical structure.

Table 2 SSA comparison for the first ten residues of protein 3A99 chain A.

SSA source	SSAs (positions 1-10)
Author(s)	CCCCC EEEEE
8 Å cutoff model trained on author(s)' SSAs	CHHHH EEEEE
DSSP	CCCCC EEEEE
8 Å cutoff model trained on DSSP SSAs	CHHHH EEEEE
STRIDE	CCCCC EEEEE
8 Å cutoff model trained on STRIDE SSAs	CHHHH EEEEE

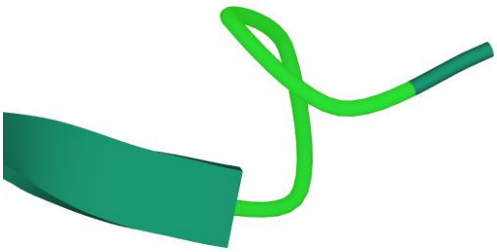

Figure 13 Residues 1-7 of protein 3A99 chain A. The light green residues, positions 2-5, were all classified as helix by the 8 Å cutoff models, but were classified as coil by the author(s), DSSP, and STRIDE.[11,27,28]

Another example of a disagreement occurs for residue 71-74 in protein 4LA2 chain A. All of the 8 Å cutoff models labels these four residues as strand, DSSP and STRIDE label residues 72 and 74 as strand and 71 and 73 as coil, and the author(s) labeled residue 71 as helix and the rest as coil, as shown in Table 3. Figure 14 shows the cartoon representation of this region with residues 71-74 shown in light green. This region does have a clear planar structure. Importantly, for the 8 Å cutoff model trained on the author(s)' SSA, residue 71 represents misclassification of helix as strand, because the author(s) labeled that residue helix, while the model labeled it strand. This is a critical

metric of the SSA method's accuracy, but here there is a clear argument that this residue could in fact be classified as strand and that the author(s)' assignment may not be correct.

Table 3 SSA comparison for residues 71 through 74 for protein 4LA2 chain A.

SSA source	**SSAs (positions 71-74)**
Author(s)	HCCC
8 Å cutoff model trained on author(s)' SSAs	EEEE
DSSP	CECE
8 Å cutoff model trained on DSSP SSAs	EEEE
STRIDE	CECE
8 Å cutoff model trained on STRIDE SSAs	EEEE

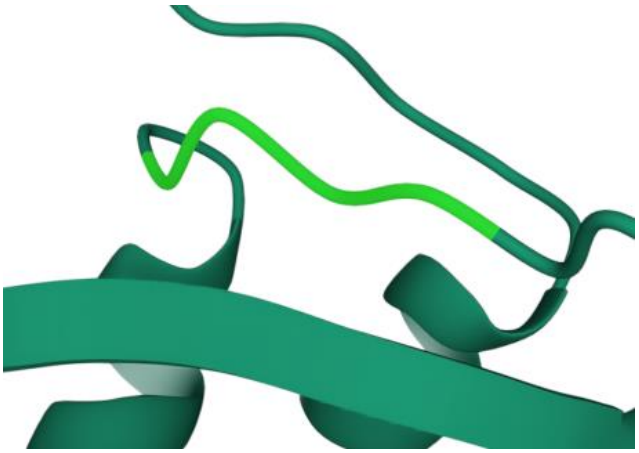

Figure 14 Part of the structure of protein 3A99 chain A. The light green residues, positions 71-74, were all classified as strand by the 8 Å cutoff models, but were classified *CECE* by DSSP and STRIDE and as *HCCC* by the author(s)[11,27,29]**.**

Discussion

It is clear that the 64-attribute topological descriptor captures secondary structure information as defined by other methods, without using a predetermined definition of SSs encoded in cutoffs and parameters. All of the models achieved an average per-protein accuracy greater than 80% with an average of less than 2% of the residue misclassification of helix as strand or strand as helix. Furthermore, the edge-length cutoff produced a statistically significant increase in overall per-protein accuracy with the 8 Å edge-length cutoff models achieving average accuracy values of 82.60%, 84.85%, and 85.55% for the author(s)', DSSP, and STRIDE SSA training label models, respectively. Because the DT of a protein structure is used to generate groups of four neighbors, the removal of long edges through the application of the edge-length cutoff appears to remove biologically and structurally meaningless connections.

Furthermore, the average per-protein true positive helix classification was greater than 80% for all nine models, with the 8 Å edge-length cutoff models achieving 85%, 87%, and 88% accuracy for the author(s), DSSP, and STRIDE training label models, respectively. While the true positive strand accuracy was lower for the no cutoff models, all of the 8 Å cutoff models achieved a value of ~84%. All of the models achieved a less than 2% helix as strand or strand as helix misclassification value per protein, with the DSSP and STRIDE models achieving values less than 1%. Interestingly, in the case of misclassification, the edge-length cutoff does not have a consistent effect. While the cause of this is unknown, the fact that the no cutoff and 8 Å cutoff models achieved

similar values, supports the idea of using the 8 Å cutoff models because of their improved accuracy across other metrics.

The topological descriptor models do not use parameters in the sense of SSA methods like DSSP and STRIDE, which use biologically- or biophysically-derived or knowledge-based parameters, though the edge-length cutoff and distance transformation function could be modulated to improve accuracy. Therefore, these models can be used as excellent arbiters for disagreements between other methods. As shown in the examples in Table 2, Table 3, Figure 13, and Figure 14, disagreements between the model assignments and the methods on which they were trained do occur, but often times further inspection of the structures reveal evidence supporting the models' assignments.

Furthermore, these models can only perform SSA as well as the method on which they were trained. This is evidenced by the fact that the models trained on the author(s)' assignment were slightly less accurate and produce a distribution of accuracy value on the test set that had a wider range than the DSSP and STRIDE models. Structure author(s) are humans that have personal idiosyncrasies and flexible rule systems that they employ when performing SSA. This leads to greater variability in their assignments as compared to the rigid rules of DSSP and STRIDE and this is reflected in the variability of the accuracy of the models.

PROTESSA: IMPROVING THE MODEL

Introduction

As described in the previous chapter, SS information was captured by the 64-attribute topological descriptor which was based purely on sequence distances between neighbors in minifold simplices. This is a great starting point, but Delaunay simplices are tetrahedra in 3D and tetrahedra have many qualities that could also be informative for SSA. Furthermore, there might be a better transformation function than the one used in the initial model exploration. Improving the ProTeSSA models trained on other methods' SSAs is important, but another aim of this work was to explore clustering. Since, the models do not rely on preconceived definitions of SSs, clusters in the topological data space have the potential to generate new definitions and understanding of SSs. In this work, many potential metrics derived from the DT were examined for their potential in SSA. In particular, a novel application of persistent homology proved particularly informative. When clustering was performed on this new feature space, the model provided interesting results that showed much overlap with other methods' SSA without any of their influence in training. This opens the door to a completely objective SSA method based purely on clusters in the SS information space.

Materials and Methods

The same dataset of high-resolution, low-percent-identity proteins that was previously described was used for this work. It contained 2,732 protein chains, 1,640 of which were used for training, and 1,092 of which were kept separate and used as the test

set. The DT of each structure was used as the basis of the topological descriptor, though the 8 Å edge-length cutoff was applied consistently because of its previous results showing statistically significant increase in accuracy.

Sequence Distance Transformation Function

In the first part of this work, several distance transformation functions were compared and the one shown in Equation 2 was selected. Upon the successful training and testing of the RF models, in particular those trained on DTs with the 8 Å edge-length cutoff applied, further analysis of the transformation function was performed in search of a more optimal function. Five distance transformation functions were examined, and the results are shown in Figure 15. For functions one through three, progressively smaller ranges are mapped to values 0, 1, 2, and 3. For function four, sequence distances of 1, 2, and ≥3 are mapped to values 0, 1, and 2, respectively. For function five, sequence distances of 1 and ≥2 are mapped to values 0 and 1, respectively. Interestingly, functions 1, 2, 3, and 4 all have a similar accuracy values and each mean accuracy is within all the others' confidence intervals. Function 5 differs as its lower accuracy falls outside of the others' confidence intervals. This suggests that mapping smaller ranges to values between 0 and 3 or 0 and 2 doesn't impede accuracy, *i.e.*, the inclusion of longer sequence distances data in the topological descriptor does not greatly improve accuracy. This makes sense because secondary structure is a local phenomenon. However, it is surprising that the fourth function, which maps distances to values 0, 1, or 2, is arguably just as accurate as the first three functions. This means that a 27-attribute topological descriptor will work just as well as the 64-attribute descriptor. The fifth function, which

results in an 8-attribute descriptor, is less accurate due to the loss of some of the local topological information. Given these results the fourth function that generates a 27-attribute topological descriptor according to function shown in Equation 5 was selected for further exploration.

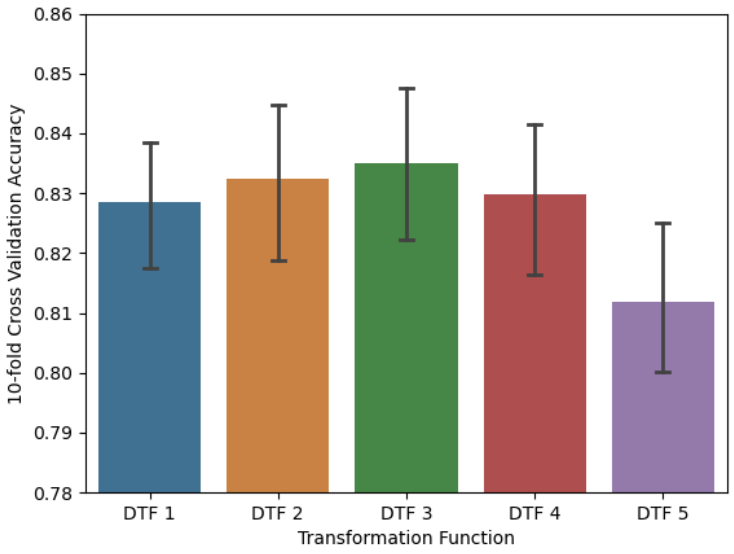

Figure 15 Comparison of RF model 10-fold CV accuracy on the 8 Å edge-length cutoff training data. Tops of the bars represent the mean accuracy and black error bars represent the 95% confidence interval. DTF 1 maps sequence distances 1, 2-4, 5-9, and ≥10 to values 0, 1, 2, and 3, respectively. DTF 2 maps sequence distances 1, 2-3, 4-5, and ≥6 to values 0, 1, 2, and 3, respectively. DTF 3 maps sequence distances 1, 2, 3, and ≥4 to values 0, 1, 2, and 3, respectively. DTF 4 maps sequence distances 1, 2, and ≥3 to values 0, 1, and 2, respectively. DTF 4 maps sequence distances 1 and ≥2 to values 0 and 1, respectively.

Equation 5 Distance Transformation Function 4

$$T: d \begin{cases} 0, & if\ d = 1 \\ 1, & if\ d = 2 \\ 2, & if\ d \geq 3 \end{cases}$$

Tetrahedral and Minifold Characteristics

The nine models that were previously tested all relied on a feature set based purely on sequence distances; each simplex in a residue's minifold was characterized by the sequence distances between the four vertices. While sequence distance data proved highly effective in training SSA models, the simplices in a DT have many other characteristics. Recalling that the simplices in the DT of a protein's CAC are tetrahedra, the geometric properties of the tetrahedra within a residue's minifold can also hold relevant information to SSA. Many of these properties were examined and are described below.

Volume

Volume was explored as a potentially informative metric because it seemed plausible that the minifolds of different SS classes could have different volumes. For example, helical residue's minifold might be larger than that of a strand. The equation for the volume of a tetrahedron is shown in Equation 6. Two volume metrics were examined: the average volume of each simplex in a minifold and the total volume of each simplex in a minifold.

Equation 6 Volume of a tetrahedron, where a, b, c, and d are each the x, y, and z coordinates of one of the four vertices of a tetrahedron.

$$V = \frac{\left|(\boldsymbol{a} - \boldsymbol{d}) \cdot \left((\boldsymbol{b} - \boldsymbol{d}) \times (\boldsymbol{c} - \boldsymbol{d})\right)\right|}{6}$$

F-vector

The f-vector of a polytope is the vector [f_0, f_1,… f_n] where f_i is the number of *i*-dimensional faces of the polytope and *n* is the number of dimensions – 1. In the 3D of protein DTs, the f-vector of a minifold contains three values: the number of points, the number of edges, and the number of faces. In a minifold, there are variable numbers of simplices and any collection of simplices can share or not share edges and faces in many ways. Thus, the f-vector of a minifold captures this information in a simple format and could potential differ between SS classes.

Edge ratio, Aspect ratio, and Radius ratio

The simplices in a minifold can take on many forms. They can be squished nearly flat or they can approximate an equilateral tetrahedron. Several metrics that capture the shape of the tetrahedra were explored [30]. These metrics involve edge lengths and the radii of the circumspheres and the insphere, where the circumsphere is defined as the sphere that contains all four vertices of the tetrahedron and the insphere is defined as the sphere that touches each of the four faces of the tetrahedron at a single point.

The edge ratio is defined as the longest edge divided by the shortest edge. The lower bound of the edge ratio is 1 for an equilateral tetrahedron and is otherwise greater than 1. The aspect ratio is defined as the circumsphere radius divided by the longest edge. When the circumsphere radius is much larger than the longest edge, this indicates a very flat tetrahedron. The radius ratio is defined as 3 times the insphere radius divided by the circumspheres radius. This value is in the range 0 to 1, where 1 indicates an equilateral

tetrahedron. Figure 16 shows two different tetrahedra and their associated circumspheres and inspheres.

Figure 16 Comparison of two tetrahedra (green) and their circum- and in-spheres, shown in grey and gold, respectively. The equilateral tetrahedron on the left has an aspect ratio, edge ratio, and radius ratio of 0.61, 1, and 1, respectively. The irregular tetrahedron on the right had an aspect ratio, edge ratio, and radius ratio of 0.56, 2.57, and 0.47, respectively.

Skew lines

Opposite edges of a tetrahedron form non-intersecting, non-parallel lines, known as skew lines and the angle between skew lines is a useful metric for measuring the shape of a tetrahedron. The equation for the angle between two skew lines is shown in Equation 7. An example for both equilateral and irregular tetrahedra is shown in Figure 17. For each tetrahedron in a minifold, there are three pairs of skew lines. In an equilateral tetrahedron, the three skew line angles are all 90º. As a tetrahedron becomes flatter, *i.e.*, as the four points approach coplanarity, two of the skew line angles will approach 0º or 180º. In radians, the values of the cosine of 0º or 180º are 1 and -1, respectively. So, by removing the arccos() from Equation 7 and taking the absolute value of the right side of the equation, we can map the skew line angle range of 0º to 360º to a value between 0 and

1, where 0 represents 90º and 1 represents a value of either 0º or 180º. In this way, each tetrahedron's skew line angles can be described by three values between 0 and 1.

Equation 7 Angle between two skew lines, where a and b are the direction vectors.

$$\theta = arccos\left(\frac{\boldsymbol{a} \cdot \mathbf{b}}{|\boldsymbol{a}| \cdot |\boldsymbol{b}|}\right)$$

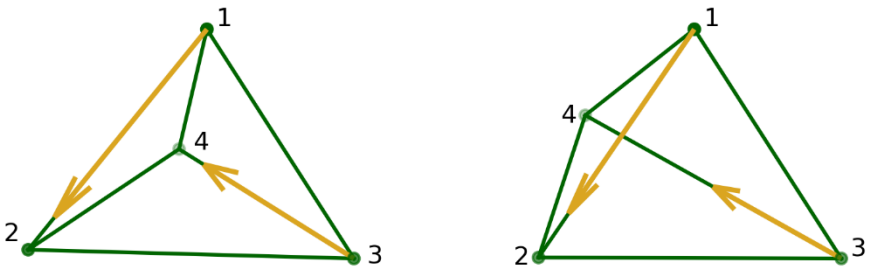

Figure 17 Example skew lines for equilateral tetrahedron (left) and irregular tetrahedron (right). The angle between the equilateral tetrahedron skew lines is 90º and the angle between the irregular tetrahedron skew lines is 106º.

Minifold Window

Many SSA methods use a window when considering certain properties of a residue. For example, for a given residue *i*, RaFoSA[15] calculates interatomic distances between pairs of residues in the range *i*-2 to *i*+2. The distances in this window can be fully calculated for residues in the range 2 to *n*-2, where *n* is the number of residues, but they cannot be calculated at the termini of a protein. One of the advantages of applying the concept of a minifold, is that the window can be used without predefined parameters.

In this work, the range of consecutive residues that are present in the minifold of residue *i*, is considered the relevant window for residue *i*. Figure 18 shows an application of the minifold window for three residues in protein 3VUB.

Figure 18 Visualization of protein 3VUB chain A with minifolds for residues 40, 47, and 101, show in red, gold, and green, respectively, shown on the left and the associated minifold windows of length 5, 8, and 3, respectively, shown in the same colors on the right.

Persistent Homology

Persistent homology (PH) is a growing area of study in topological data analysis and is applied for the first time to SSA here. In PH, topological features, or homologies, of a space are tracked as a filtration is applied. This can be visualized as a set of small spheres whose radii are increasing. The set of individual spheres are considered to be the set of topological features born at the beginning of the analysis. As the radii grow and the spheres grow in size, they will begin to intersect with each other. An intersection between two spheres is treated as the merging of topological features and marks the *death* of one feature while the other survives. Eventually all the spheres will overlap and there will only be one feature that survives infinitely. Importantly, as the radii increase it is possible

for intersecting spheres to enclose areas or volumes. These enclosed spaces are also topological features which will have a *birth*, when they are created, and a *death*, when the radii increase to the point that the involved spheres all overlap.

Ripser is a python package that can be used to compute persistent homologies and plot persistent diagrams for a set of points[31]. To apply PH here, the points included in a given residue's minifold window are normalized. This is performed by subtracting the coordinates of point *i* from the coordinates of point *i*+1. This can be thought of as moving the vector from residue *i* to residue *i*+1 such that the position of residue *i* is located at the origin. Figure 19 shows how the minifold window of residue 40 from protein 3VUB is normalized. The persistence diagram for this set of points is shown in Figure 20. From the persistence diagram, we can see that four H_0 features, or points, are born at the start. As the space is filtered, three of these H_0 features die and one lives infinitely. It can also be seen that one H_1 feature, or enclosed area, is born and dies. In this work, the number of H_0 and H_1 features, the average lifespan of those features, and the maximum lifespan of those features are calculated for the set of points in a residue's minifold window.

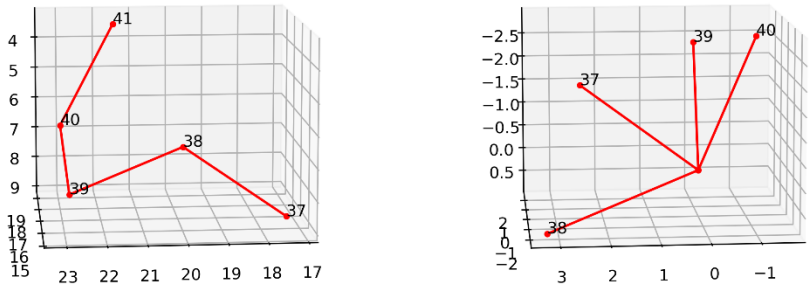

Figure 19 The minifold window of residue 40 from protein 3VUB chain A is shown on the left. On the right, the normalized vectors for each pair of consecutive residues in the window are shown.

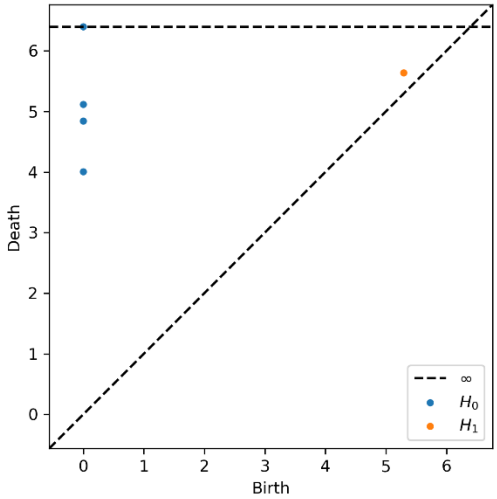

Figure 20 The persistence diagram for the normalized points from the minifold window of residue 40 from protein 3VUB chain A.

Dataset and Topological Descriptor

In addition to the previously described sequence distance based topological descriptor, it was important to test volume, aspect ratio, edge ratio, radius ratio, skew line angles, and persistent homology to determine how much SS information each captured.

One of the advantages of using an ensemble ML model is the reduced tendency for the model to overfit the training data, but the associated downside is that the model can be more difficult to interpret. For a RF model, feature importance is one way to gain insight into the model. Feature importance is a measure of the node impurity weighted by the probability of reaching that node; the magnitude of the value shows how well a particular feature splits the data into the correct classes. In other words, feature importance is a measure of how critical a feature is to the classification model.

To determine how informative each of the new features were, a RF model was trained on the training set. Figure 21 shows the importance of 7 sets of features: 1) the first 27 bars, shown in red, are the DTF 4 sequence-distance features, or simplex types, generated by Equation 5, 2) the 27 blue bars correspond to the average aspect ratio of each type of simplex in the minifold as defined by DTF 4, 3) the grey bars correspond to the average edge ratio of each type of simplex in the minifold as defined by DTF 4, 4) the purple bars correspond to the average volume of each type of simplex in the minifold as defined by DTF 4, 5) the 81 gold features are made of 3 sets of 27 features that correspond to the average value of each of the three skew angles for each simplex in the minifold as defined by DTF 4, 6) the 3 salmon colored bars correspond to the F-vector, or the number of edges, faces, and simplices in the minifold, and 7) the 6 green bars correspond to persistent homology features, including the number of H_0 and H_1 features, the average lifespan of each class of features, and the maximum lifespan of each class of feature.

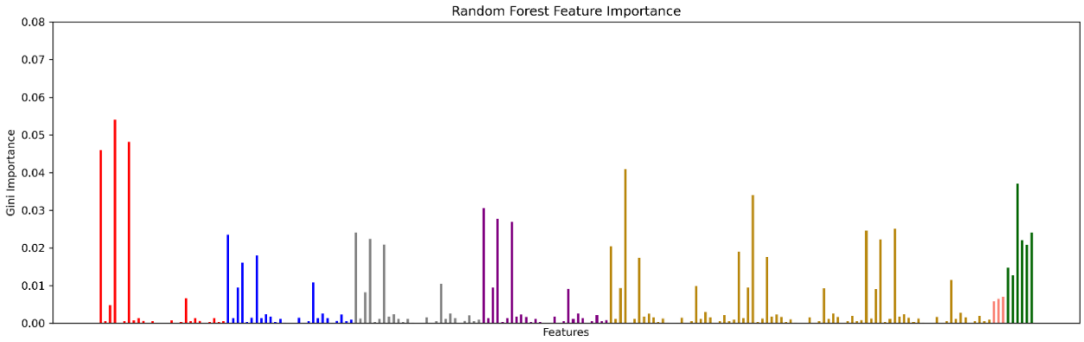

Figure 21 Feature importance for RF model.

To further determine which features would be most informative, 6 RF models were evaluated with 10-fold CV. DTF 4 (a), used 27 features as defined by DTF 4 in Equation 5. DTF 4 (b), use the same data as DTF 4 (a), except that each value was scaled by the total number of simplices included in the minifold, *e.g.*, if a minifold contained 3 of one type of simplex and 6 simplices in total, then DTF 4 (a) would have a value for 3 for that feature and DTF 4 (b) would have a value of 0.5 for that feature. AR-ER-V contained 81 features, 27 for each aspect ratio, edge ratio, and volume, where each value in each set of 27 feature represented the average of that metric for each simplex of that type, *e.g.*, if a residue's minifold contained 3 residues of one type that had edge ratios of 1.5, 2, and 3, then the feature value would be (1.5+2+3)/3. The skew angles model was similar to AR-ER-V, in that it also had 81 features where each feature contained the average metric value for a specific type of simplex, where each of the three sets of 27 features corresponded to one of the three skew angles. The F vector model contained only 3 features, corresponding to the number of points, edges, and faces, in each minifold. The Persistent Homology model had 6 features, corresponding to the number of H_0 and H_1

features, the average lifespan of each class of features, and the maximum lifespan of each class of feature.

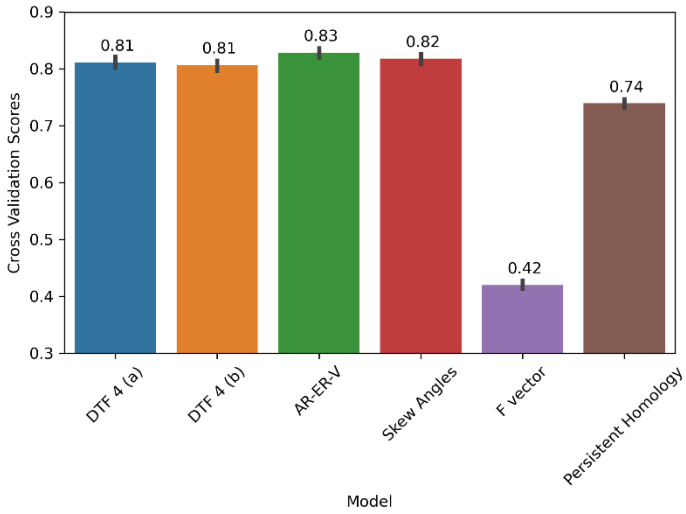

Figure 22 CV scores for separate RF models trained on each set of parameters.

Figure 22 shows to 10-fold CV results for each of the 6 models described above. Interestingly, the first four models' average scores fall within each other's confidence intervals, despite the fact that the AR-ER-V and Skew Angles models had 81 features. This suggests that the sequence distance data contained in the first 2 models is informative and adding tetrahedral metrics at the simplex level does not increase accuracy. The F vector model performed the worst by far. This is not surprising as it had both the fewest number of features and the least descriptive structural information, containing only the number of points, edges, and faces. The Persistent Homology model

performed very well despite having only 6 features, suggesting that it does in fact capture SS information.

Based on the above data, a set of 35 features was chosen for further exploration. The first 27 features of this set were similar to the previously described topological descriptor except with the new distance transformation function described in Equation 5. The next 6 features included persistence homology information, specifically the number of H_0 and H_1 features, the average life of each, and the maximum life of each. In the absence of any of these features, the value of "0" was used. The final two features used the edge ratio metric. The first was the average edge ratio of all the simplices in the minifold. The second was the average edge ratio of the simplices that had at least two residues from the backbone minifold window. Similar to the previously described nine models, in this study three models were trained on the SSA labels of the structure author(s), DSSP, and STRIDE.

Results

Boxplots of the distributions of classification accuracy per protein in the test set for the 3 models are shown in Figure 23. The Mann-Whitney *U* test was used to test for statistical significance and is displayed in this figure with horizontal brackets indicating the pairs of distributions that were tested. One asterisk indicates significance, while four asterisks indicates very high significance, as described in the figure legend.

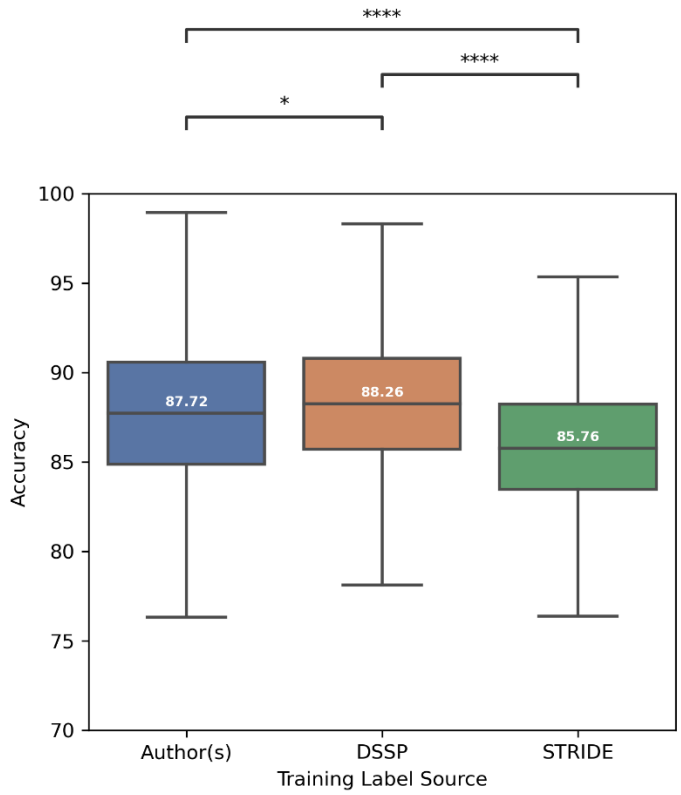

Figure 23 Boxplots, with medians shown in white, of the distributions of classification accuracy per protein in the test set for each of the three models. Statistical significance is shown above the chart. 'ns' indicates not significant, * indicates a p-value <= 0.05, ** indicates a p-value <= 0.01, * indicates a p-value <= 0.001, and **** indicates a p-value <= 0.0001. Outliers were not shown in this figure.**

Figure 24 shows the confusion matrices for the three models. While the boxplots show the average accuracy distribution, where each value is the total number of residues correctly classified divided by the total length of the protein chain, the confusion matrices show the correct and incorrect classifications at the residue level. The model trained on the author(s)' SSA had the highest true positive helix value, while the DSSP-trained model had the highest true positive strand and coil values. The STRIDE-trained model had the lowest helix as strand misclassification value, 15, but the highest strand as helix misclassification value, 480.

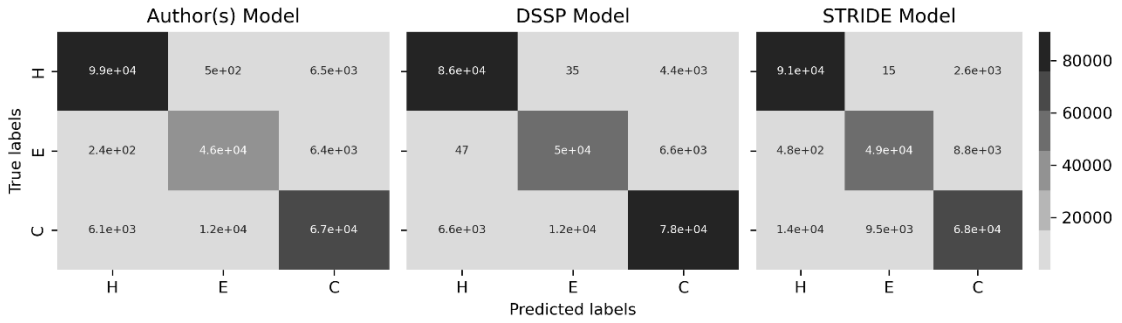

Figure 24 Confusion Matrices for the three models. The left matrix is for the model trained on the structure author(s)' SSA where ground truth is the structure author(s)' labels for the test set. The middle is for the model trained on DSSP where ground truth is the DSSP labels for the test set. The right is for the model trained on STRIDE where ground truth is the STRIDE labels for the test set. The darker the cell's color, the higher the value.

Figure 25 and Figure 26 show the true positive helix and strand accuracy and the misclassification of helix as strand and strand as helix, respectively. For the true positive helix and strand figures, the values are the average of the total number of correctly labeled helices and strands divided by the total number of helices and strands per protein, respectively. Similarly, for the misclassification figures, the values are the average of the total number of misclassifications of helices as strands and strands as helices divided by the number of helices and strands, respectively, per protein in the test set. The STRIDE-trained model correctly classified the most helical residues but missed the most stands, conversely, the author-trained model correctly classified the most strands but missed the most helices. For misclassifications, the STRIDE-model classified the least helices as strands, at an average of 0.02 per protein, but the most strands as helices, at an average of 1.56 per protein. The DSSP-trained model misclassified helices as strands and strands as helices the least often, at 0.07 per protein and 0.11 per protein on average, respectively.

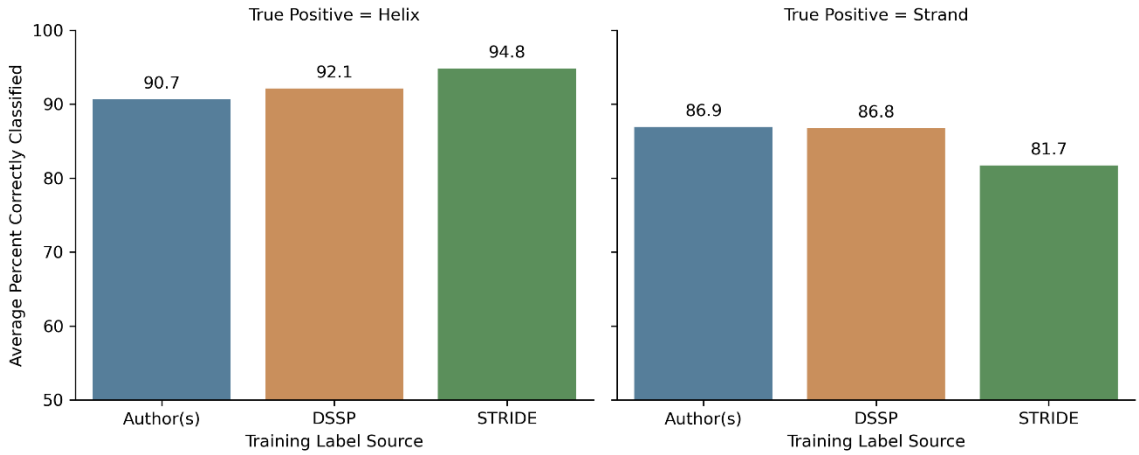

Figure 25 True positive helix and true positive strand average accuracy per protein in the test set

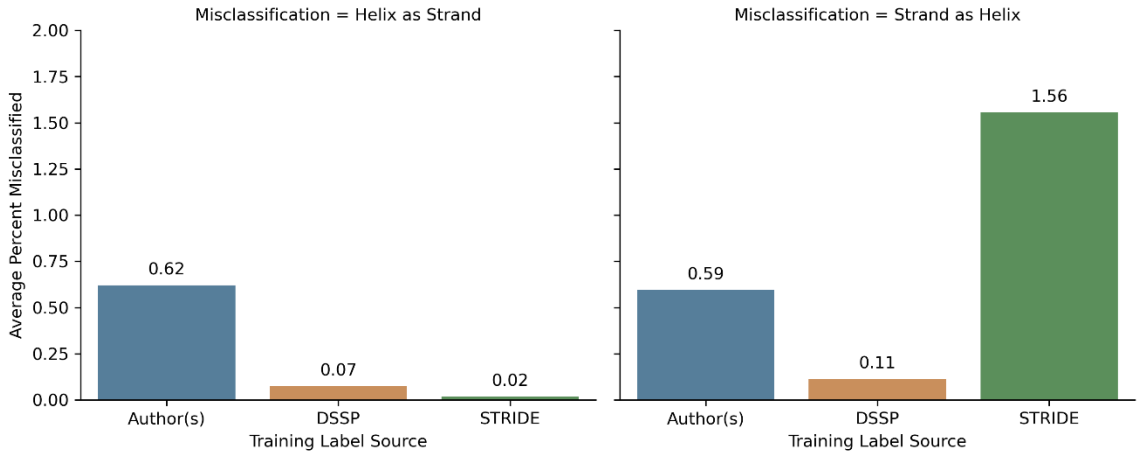

Figure 26 Average misclassifications of helix as strand and strand as helix per protein in the test set for each of the nine models.

There are many similarities and important differences between the results of the improved ProTeSSA models and the initial versions. For the initial models, the medians

of per-protein accuracy of 8 Å models trained on the author(s)' SSA, DSSP, and STRIDE were 83.33%, 84.91%, and 85.63%, respectively. The improved author-trained, DSSP-trained, and STRIDE-trained models achieved values of 87.72%, 88.26%, and 85.76%. The author-trained model showed the greatest improvement of 4.39%, while the STRIDE-trained model showed the least improvement of 0.13%. The whiskers in Figure 23 show the same trend they displayed in Figure 8; the minimum and maximum whiskers span a wider range for the author-trained model than they do for the DSSP-trained and STRIDE-trained models. The ranges spanned from 76.28% to 99.17%, 78.11% to 98.38%, and 76.32% to 95.38% for the author-trained, DSSP-trained, and STRIDE-trained model, respectively, with the author-trained range spanning 22.88% as compared to 20.26% and 19.04% by the DSSP-trained and STRIDE-trained models, respectively. Importantly, all of these ranges are smaller than the initial smallest range of 24.29% from the DSSP-trained 8 Å model. In other words, the author-trained model on the new feature set was more consistently accurate than the most consistent initial model.

The confusion matrices, shown in Figure 24, also show improvements over the initial models. For all three models, the number of correctly classified helices improved, with the author-trained and STRIDE-trained models showing the most improvement. While the number of correctly classified strands was marginally better for the previous models, the number of correctly classified coils improved for the author-trained and the DSSP-trained models, but got worse for the STRIDE-trained model. All of the new models brought down the number of helical residues incorrectly classified as strand an

order of magnitude, while only the DSSP-trained model achieved this for the misclassification of strand as helix as well.

These results are echoed in Figure 25 and Figure 26 which show the true positive helix and strand values and the misclassifications of helix as strand and strand as helix, respectively. The true positive classifications for the initial models were all lower and the misclassifications were all higher, except for one notable example: the initial STRIDE-trained model had fewer strand as helix misclassifications.

Outliers

To get a deeper understanding of the weakness of the models, the low accuracy outliers were examined. The total set of the low accuracy outliers for the three models included 34 protein chains, three of which were outliers in all three cases: 4KU0 chain D, 3D9X chain A, and 3ULJ chain A. Unlike the outliers from the first set of models, there were no common characteristics between these structures. 4KU0 is a hetero-4-mer and, while chains A, B, and C are form a triangular prism comprised mainly of strands, chain D is predominately coil. 3D9X is a homo-3-mer, and is comprised of mostly strands and 3ULJ is a monomer comprised of mostly strands. In this case, it doesn't appear that the models are having difficulty identifying a particular type of global structure, rather strands are more difficult to classify, as seen in the true positive strand value and the strand as helix misclassification value.

Clustering

The RF models described so far throughout this work have relied on other SSA methods in training. There are advantages to this: 1) the models can learn the traits of

other methods, *e.g.*, the idiosyncrasies of crystallographer(s)' SSA, 2) the accuracy of the RF models are a reflection of the internal consistency of other methods, and 3) the RF framework is flexible, so multiple methods can be used in training and their resulting predictions can be compared to determine a consensus SSA. Despite these advantages, it was important to explore the possibility of using the topological data to perform SSA without using other methods' SSAs in training. Clustering provides a means build this type of model.

K-means is a method to divide some number of samples into *k* clusters. There are several algorithms that can be used, but the end result is the same: the sample space is partitioned into *k* regions and new samples can be classified based on the region in which they fall. It is tempting to set k=3 when clustering the topological training data, because that agrees with the three-class SS scheme used by the RF models. However, the elbow method is a useful tool for objectively determining the best value for *k.* In the elbow method, increasing values of *k* are used, clustering metrics are calculated, and then these metrics are compared. Two commonly used metrics are distortion and inertia. Distortion is the average of the squared distances of each sample from its respective cluster centroid. Inertia is the sum of squared distance of each sample from its respective cluster centroid. Figure 27 shows these metrics for values of *k* from 1 to 9. In both plots it can be seen that there is a significant decrease in both values up to the value k=3, but at increasing values of *k* beyond 3 there is a smaller reduction at each step. Specifically, at k=2 to k=3, the distortion dropped 22% and 13% when compared to the previous value of *k*, respectively. From k=4 to k=9, this decrease was less than 6% at each step with the decrease shrinking

at each step. Similarly, for *k*=2 and *k*=3, the inertia dropped 48% and 33% when compared to the previous values of k, respectively. For values of *k* greater than 3, this decrease was less than 10% at each step, decreasing at each step. These results suggest *k*=3 is the best value for k-means clustering.

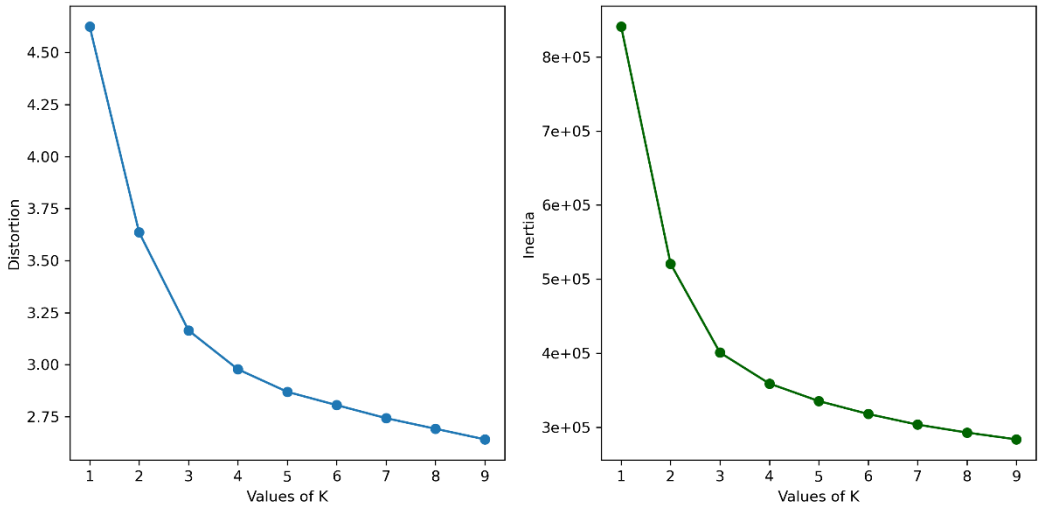

Figure 27 Elbow method for determining optimal number of clusters in topological data.

One method to inspect the potential of k-means clustering to perform SSA is dimensionality reduction. Uniform Manifold Approximation and Projection (UMAP) stands apart from other methods because it's authors claim that it better preserves global structure [32]. Here, the 35-dimenstional topological data based on sequence distance, persistent homology, and edge-ratios was reduced to two dimensions using UMAP. This projection is shown in Figure 28 with two different colorings: left) with points colored depending on their SSA as determined by the structure author(s) and right) with points colored according to their k-means (*k*=3) cluster.

There are some interesting inferences that can be drawn from these visualizations. Using the author(s)' SSAs, the helical residues and the strand residues occupy different regions of the space with little overlap. The coil residues reside in between the two other classes and frequently overlap with either the helical or strand regions. This supports the basic underlying assumptions about SS: 1) helices and strands are very different structures and 2) coils have no pre-defined structure and therefore can resemble either helix or strand. When comparing the two colorings, one can see a lot of agreement between strand and cluster 0, helices and cluster 1, and coils and cluster 2. While points occupy the entire range of the y-axis, most of the points occupy the middle of the x-axis in Figure 28. There is an exception, however; there is a small cluster of points falling in middle of the y-axis and the extreme upper limits of the x-axis. This cluster of points is comprised of coils and strands, as defined by the structure author(s), in the left plot, and clusters 0 and 2 in the right plot. This also follows the agreement between author(s)' SSAs and the clusters as described above.

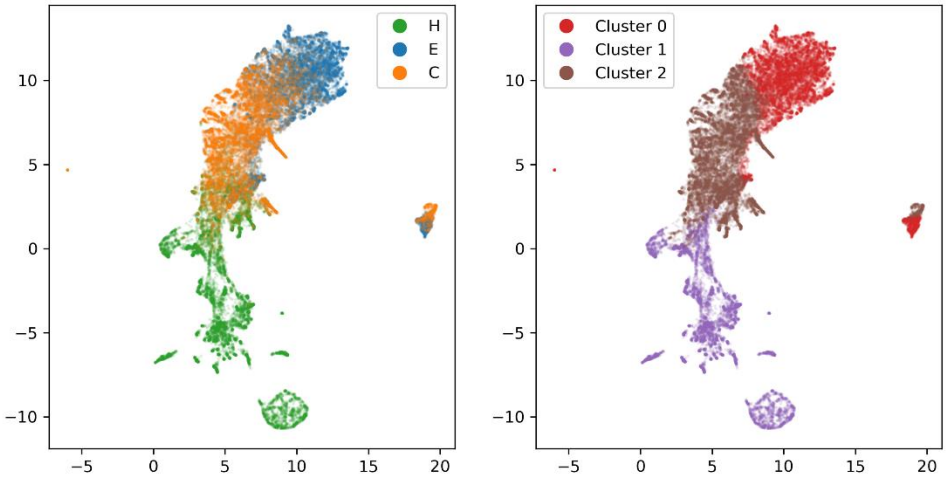

Figure 28 UMAP dimensionality reduction of the training data with marker color showing author(s)' labels on the left plot and K-means (k=3) cluster label on the right plot.

Using this overlap between the clusters and the author(s)' SSAs, it is possible to assign residues a SS classification based on the cluster assignment, specifically, for k-means cluster assignments of 0, 1, and 2, the SS classifications are strand, helix, and coil respectively. By generating SSAs using the k-means model and SS-classification conversion, it is possible to assign accuracy to the k-means predictions based on the agreement between the k-means assignment and the test set protein's structure author(s)' SSAs. This is shown in Figure 29, which includes the three boxplots from Figure 23 and the boxplot of accuracy values of the k-means model. Figure 30 shows the k-means model's helix and strand misclassification values. From these two figures, it is evident that the k-means model is highly accurate with low helix and strand confusion when it is compared to the author(s)' SSA.

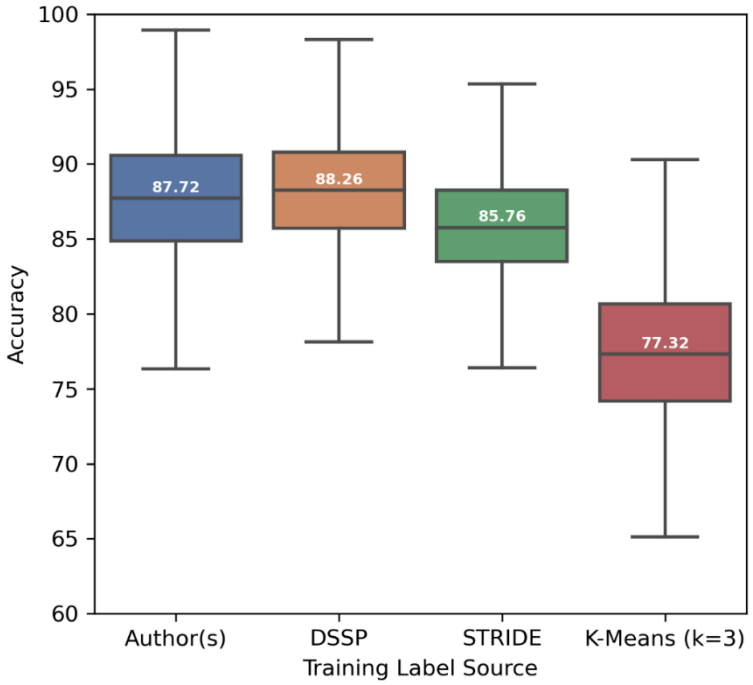

Figure 29 Boxplots, with medians shown in white, of the distributions of classification accuracy per protein in the test set for the three previously described models and the K-Means (k=3) model.

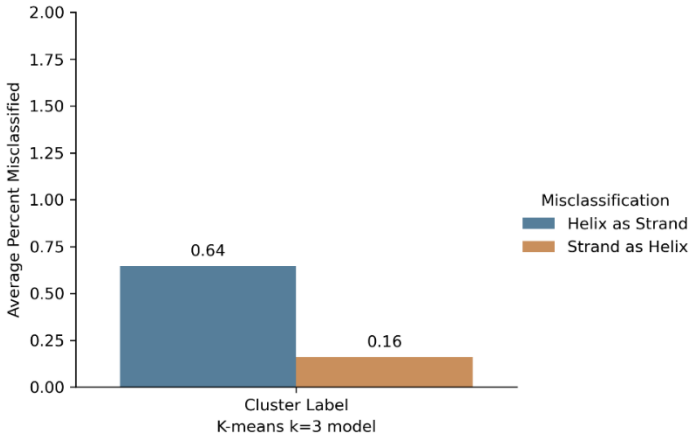

Figure 30 Cluster model misclassifications when compared to author(s)' labels.

Discussion

The initial models proved that the 64-attribute, sequence-based topological descriptor captures SS information without parameters and can be used to train accurate SSA models. Further exploration into metrics that describe the simplices revealed other parameters that capture SS information. Important insight into three topics was gained from exploration: 1) the minifold window, 2) persistent homology, and 3) edge ratio.

Minifold Window

The minifold window is a novel way to define a backbone window in protein structure analysis. RaFoSA uses a 5-residue window to generate features for a given residue, but this is not possible for residues 1, 2, *n*-2, and *n*-,1, where *n* is the number of residues in a protein chain. This type of window is common in many algorithms used in protein structure analysis and the termini always present a problem that must be accounted for. This results in some level of inconsistency in all cases because the termini must be treated differently than every other residue in the protein. ProTeSSA does not have this problem because the minifold is used to describe the window. Every residue in the DT is a member of some number of simplices and this collection of simplices has edges that run along some length of the protein backbone. This length is variable; it may be three or four residues long at the termini or six residues long for internal residues, but though the value changes the definition is consistent. This is critical for ML models that can only learn to predict as well as the input data allows because inconsistent inputs will hamper learning.

Persistent Homology

While PH is a growing area of study in a wide range of fields, this work marks its first known application to secondary structure. 6 PH training features were used in these models, all based on the residues present in the minifold window. The positions of these residues are normalized such that the vector from residue *i* to residue *i*+1 becomes the vector from the origin to a point of the same magnitude and direction as the original vector. This allows the PH technique to be applied in the same manner regardless of direction and path of the minifold-window segment of the backbone.

The PH features were the number of H_0 and H_1 features, the average life not including infinity of those features, and the maximum life below infinity of those features. In the absence of any of these features, the value of "0" was used. The number of H_0 features is simply the number of residues in the minifold window. The average life and maximum life of an H_0 feature are measures of how far apart and how clustered the residues are in the normalized space. In other words, if there are four points that are clustered into two groups, the average life and the maximum life will be quite different, while four equally distant points would have similar average life and maximum life. An H_1 feature indicates an enclosed area. This will occur when the normalized points approximate a ring and the average life and maximum life are indications of how long this hole lasts. Conceptually, we can imagine that a prototypical helix is normalized to a ring and a prototypical strand is normalized to two disparate clusters of points. The results support this understanding.

Edge ratio

Of the many tetrahedral metrics explored in this work, the edge ratio, the value of the longest edge divided by the shortest edge, appeared to capture the most SS information. While the aspect ratio and radius ratio both use the circumsphere radius in their calculation, the edge ratio requires only the edge-lengths of a simplex. This suggests that information about the circumsphere and insphere is not as useful in SSA. The reason is unclear, but it is perhaps related to the generality of the spherically-related metrics, *i.e.*, they both use all of the positions of the vertices in their calculation, while the edge ratio does not necessarily because the longest edge and the short edge could share a vertex. Furthermore, the edge ratio varies less widely than the other metrics.

Future Work

The results of the improved models show that author-trained model is much better with PH applied. While the DSSP-trained model was slightly better it some metrics, the author-trained model had the highest number of correctly classified helical residues in the test set and the highest average number of correctly labeled strands per protein. These results suggest that there may be only marginal differences between the DSSP-trained model and the author-trained model, which is significant.

DSSP is an algorithm, therefore it follows a set of clearly defined rules to perform SSA and these rules are invariable. This makes DSSP more consistent that the author(s)' SSA process. Humans use their experience and intuition, often in combination with DSSP, to perform SSA. This allows them to correctly label residues that may not conform to DSSP's strict rules, but it also inserts variability into the assignment process.

This means that the improved ProTeSSA model captures human tendencies much better than the initial models. Deeper examination into the specific cases where the author-trained and the DSSP-trained ProTeSSA models disagree could lead to important insights into when and why author(s)' and DSSP assignment disagree.

Another area of future exploration stems from the k-means cluster model. Unlike the author-trained, DSSP-trained, and STRIDE-trained models, the cluster model is completely objective. The other three models do not have built-in definitions of SSs, which sets them apart from most SSA methods, but they are trained on other SSA methods, so they do learn other methods' definitions to some extent during the training process. The cluster model does not. It is very interesting that the k-means model cluster agrees well with the author(s)' assignment in both the UMAP dimensionality reduction and when tested on the test set. The cases where the cluster model and the author(s)' disagree will be a fascinating area for future exploration. Maybe the cluster model could be an alternative way to define SSs; because the predictions are based on clusters present in the topological data, it is completely objective and therefore its definitions have grounding in nature. Perhaps there are other topological features, maybe even other applications of PH, that would lead to improvements in clustering, further improving these novel definitions of SS. The impacts of these alternative SS definitions could have far reaching consequences that have the potential to deepen our understanding of protein structure.

WEB IMPLEMENTATION

Introduction

The research described above shows that SSA is possible without the need for parameters and that a model based on topology is able to capture secondary structure information and perform SSA accurately. Therefore, it is important that the tool is made available to the public for research use and development. The ProTeSSA program is currently available for use at omics.gmu.edu/protessa. Eventually, ProTeSSA will be released as open-source code so that the bioinformatics community can develop and improve upon the tool.

Web Application

The web server version of the tool needed to have a few core components and features. Both DSSP and STRIDE have databases of precomputed SSAs and it was important for ProTeSSA to have the same. Additionally, the webserver was designed to have both a simple interface and an advanced interface. The simple interface is designed to be as close to a one-click process as possible; the user uploads a pdb file and the sever computes the structure. The advanced interface allows the user to modify some parameters such as edge-length cutoff and model selection.

Landing Page

The ProTeSSA landing page, shown in Figure 31, is designed to be easy to understand and use. There is a navigation bar at the top, which is present in all of the site's page, that allows the user to return to the home page when desired, learn more

about the tool, and contact me with any questions. Clicking “Home” brings the user back to the landing page. Clicking “About” sends the user to a PDF documentation file that gives both background on the ProTeSSA tool and information about how to use the server. Clicking “Contact” opens a messaging prompt that allows the user to send me an email with any questions or comments he or she may have. Beneath the navigation bar, there is a logo, a title that expands the abbreviation of the tools’ name, a few simple instructions and then two buttons. The buttons “one-click” and “advanced” send the user to either the simple or advanced page, respectively, where he or she can submit a file.

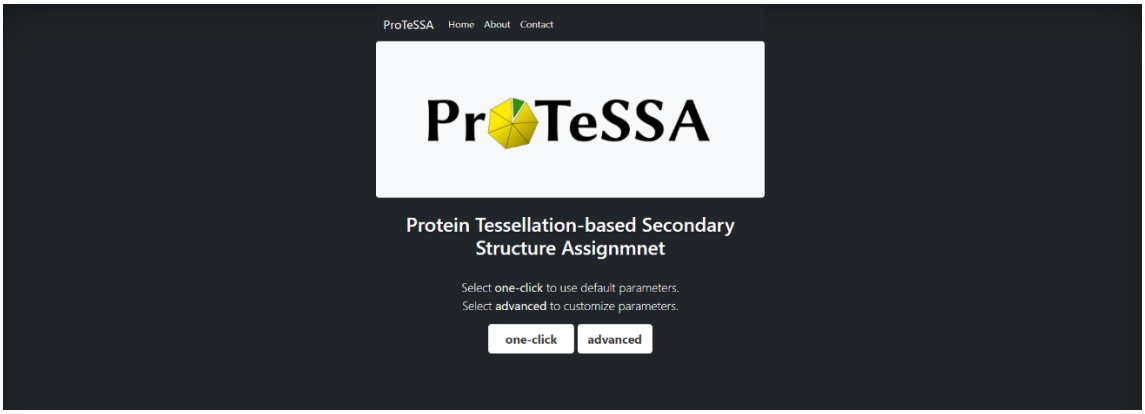

Figure 31 ProTeSSA landing page.

One-click Interface

The “one-click” button sends the user to the simple entry page shown in Figure 32. Other than a navigation bar, this page includes a file upload tool and an upload button. It was designed for ease of use and sends the user to the same results page described below.

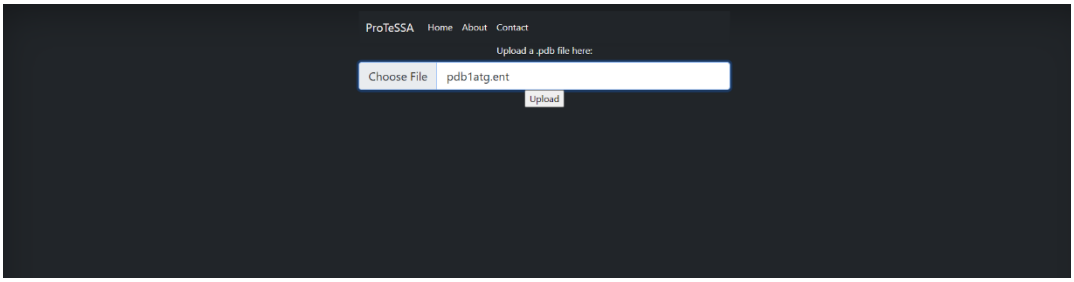

Figure 32 "One-click" PDB file entry page.

Advanced Interface

Clicking the “advanced” button on the landing page takes the user to the page shown in Figure 33. Besides the same PDB file input as the “one-click” page, there are several options for the user to choose from before uploading. First, there is a text entry box where the user can enter the chain identifiers for which SSAs are desired. The “one-click” page performs SSA on all chains, but the user may only want assignments for one or two of the chains present in a PDB file. There is also an option to choose a specific ProTeSSA model. At this point, only two models are available: one trained on structure author(s)’ SSAs and one based on K-means clustering. The final option is to choose which edge-length cutoff should be applied to the Delaunay tessellation: 8 or 10 Å.

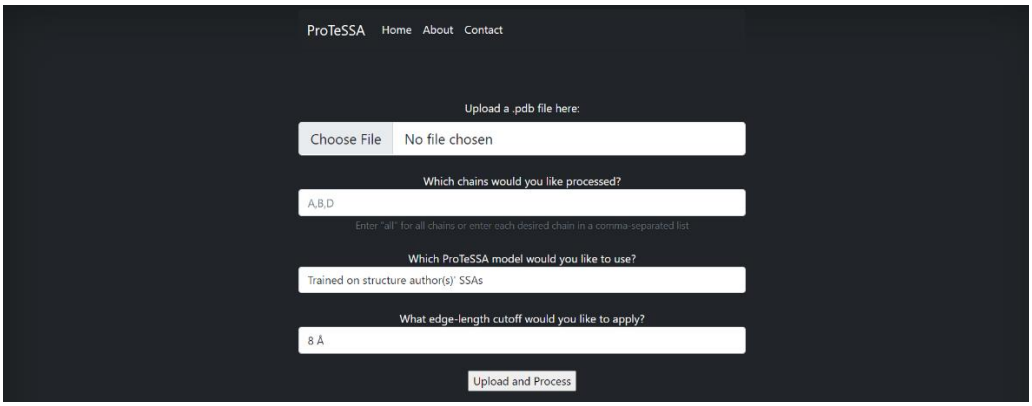

Figure 33 Advanced interface

Results page

Once the user proceeds through either the “one-click” or “advanced” interface page, he or she is sent to a results page. On this page, shown in Figure 34, the SSAs are displayed in a color-coded format where the letters for strands are gold, helices pink, and coils white. There is also a button that allows the user to download the SSAs in a .txt format. Finally, there is an interactive visualization that allows the user to see the results of ProTeSSA. By mousing over the visualization panel, the user can modulate the image. Right-clicking and holding allows the user to rotate the visualization by moving the mouse. Left-clicking and holding allows the user to pan the image up, down, left, or right by moving the mouse. The scroll wheel of the mouse allows the user to zoom in or out. There are also several buttons on the left side of the panel that can be pressed to change what is displayed: “Backbone” causes only the backbone to be displayed, “Tessellation” causes the tessellation to be displayed, “Backbone + Tessellation” shows the tessellation and backbone, and “Color by Secondary Structure” causes the backbone to be colored

using the ProTeSSA SSAs. There is also a floating toolbar that allows the user to download the visualization as a .png file.

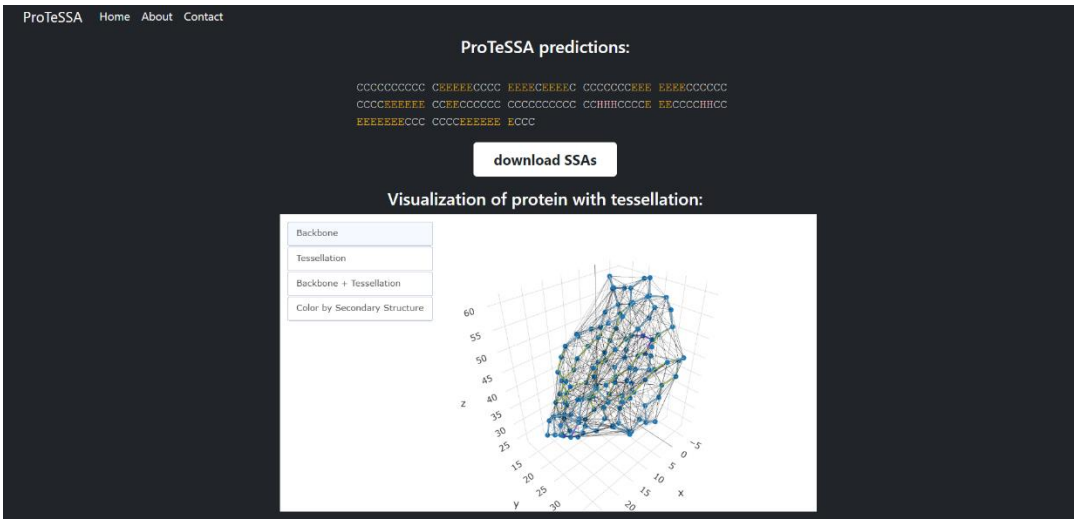

Figure 34 Results page

Future Work

The above descriptions and screenshots display the current state of the ProTeSSA web page at this time of writing. There will be continued improvement to this service as the tool is developed. Some of the future plans include adding more models to the advanced page, creating an easily searchable database, including a flat text file version of that database, creating a downloadable executable version of ProTeSSA, and implementing a GitHub page so that other may download and experiment with the tool.

REFERENCES

[1] L. Pauling, R.B. Corey, H.R. Branson, The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain, Proc. Natl. Acad. Sci. (1951). https://doi.org/10.1073/pnas.37.4.205.

[2] L. Pauling, R.B. Corey, Configurations of Polypeptide Chains With Favored Orientations Around Single Bonds: Two New Pleated Sheets, Proc. Natl. Acad. Sci. (1951). https://doi.org/10.1073/pnas.37.11.729.

[3] I. Sillitoe, N. Bordin, N. Dawson, V.P. Waman, P. Ashford, H.M. Scholes, C.S.M. Pang, L. Woodridge, C. Rauer, N. Sen, M. Abbasian, S. Le Cornu, S.D. Lam, K. Berka, I.H. Varekova, R. Svobodova, J. Lees, C.A. Orengo, CATH: Increased structural coverage of functional space, Nucleic Acids Res. 49 (2021). https://doi.org/10.1093/nar/gkaa1079.

[4] J.M. Chandonia, N.K. Fox, S.E. Brenner, SCOPe: Classification of large macromolecular structures in the structural classification of proteins - Extended database, Nucleic Acids Res. 47 (2019). https://doi.org/10.1093/nar/gky1134.

[5] N.K. Fox, S.E. Brenner, J.M. Chandonia, SCOPe: Structural Classification of Proteins - Extended, integrating SCOP and ASTRAL data and classification of new structures, Nucleic Acids Res. 42 (2014). https://doi.org/10.1093/nar/gkt1240.

[6] A. Waterhouse, M. Bertoni, S. Bienert, G. Studer, G. Tauriello, R. Gumienny, F.T. Heer, T.A.P. De Beer, C. Rempfer, L. Bordoli, R. Lepore, T. Schwede, SWISS-MODEL: Homology modelling of protein structures and complexes, Nucleic Acids Res. 46 (2018). https://doi.org/10.1093/nar/gky427.

[7] C. Cao, G. Wang, A. Liu, S. Xu, L. Wang, S. Zou, A new secondary structure assignment algorithm using Cαbackbone fragments, Int. J. Mol. Sci. (2016). https://doi.org/10.3390/ijms17030333.

[8] J. Martin, G. Letellier, A. Marin, J.F. Taly, A.G. De Brevern, J.F. Gibrat, Protein secondary structure assignment revisited: A detailed analysis of different assignment methods, BMC Struct. Biol. (2005). https://doi.org/10.1186/1472-6807-5-17.

[9] W. Kabsch, C. Sander, Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features, Biopolymers. (1983). https://doi.org/10.1002/bip.360221211.

[10] D. Frishman, P. Argos, Knowledge-based protein secondary structure assignment, Proteins Struct. Funct. Bioinforma. (1995). https://doi.org/10.1002/prot.340230412.

[11] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, The Protein Data Bank (www.rcsb.org), Nucleic Acids Res. (2000). https://doi.org/10.1093/nar/28.1.235.

[12] H. Haghighi, J. Higham, R.H. Henchman, Parameter-Free Hydrogen-Bond Definition to Classify Protein Secondary Structure, J. Phys. Chem. B. 120 (2016). https://doi.org/10.1021/acs.jpcb.6b02571.

[13] T.K. Ho, Random Decision Forests Tin Kam Ho Perceptron training, Proc. 3rd Int. Conf. Doc. Anal. Recognit. 1 (1995).
[14] L. Breiman, Random Forests, Mach. Learn. 45 (2001) 5–32. https://doi.org/10.1023/A:1010933404324.
[15] E.O. Salawu, RaFoSA: Random forests secondary structure assignment for coarse-grained and all-atom protein systems, Cogent Biol. 2 (2016). https://doi.org/10.1080/23312025.2016.1214061.
[16] F. Dupuis, J.F. Sadoc, J.P. Mornon, Protein Secondary Structure Assignment Through Voronoï Tessellation, Proteins Struct. Funct. Genet. (2004). https://doi.org/10.1002/prot.10566.
[17] R.K. Singh, A. Tropsha, I.I. Vaisman, Delaunay tessellation of proteins: four body nearest-neighbor propensities of amino acid residues., J. Comput. Biol. (1996). https://doi.org/10.1089/cmb.1996.3.213.
[18] D.L. Bostick, M. Shen, I.I. Vaisman, A simple topological representation of protein structure: Implications for new, fast, and robust structural classification, Proteins Struct. Funct. Genet. (2004). https://doi.org/10.1002/prot.20146.
[19] T. Taylor, M. Rivera, G. Wilson, I.I. Vaisman, New method for protein secondary structure assignment based on a simple topological descriptor, Proteins Struct. Funct. Genet. (2005). https://doi.org/10.1002/prot.20471.
[20] T.J. Taylor, L.I. Vaisman, Statistical geometry and topology of real and model protein structures, in: Proc. - 3rd Int. Symp. Vor. Diagrams Sci. Eng. 2006, ISVD 2006, 2006. https://doi.org/10.1109/ISVD.2006.34.
[21] T.J. Taylor, I.I. Vaisman, Graph theoretic properties of networks formed by the Delaunay tessellation of protein structures, Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys. (2006). https://doi.org/10.1103/PhysRevE.73.041925.
[22] M. Masso, Generation of atomic four-body statistical potentials derived from the delaunay tessellation of protein structures, in: Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS, 2012. https://doi.org/10.1109/EMBC.2012.6347439.
[23] M. Mirzaie, M. Sadeghi, Delaunay-based nonlocal interactions are sufficient and accurate in protein fold recognition, Proteins Struct. Funct. Bioinforma. 82 (2014). https://doi.org/10.1002/prot.24407.
[24] A. Okabe, B. Boots, K. Sugihara, S.N. Chiu, D.G. Kendall, Ch 2. Definitions and Basic Properties of Voronoi Diagrams, Spat. Tessellations. (2008). https://doi.org/10.1002/9780470317013.ch2.
[25] G. Wang, R.L. Dunbrack, PISCES: A protein sequence culling server, Bioinformatics. (2003). https://doi.org/10.1093/bioinformatics/btg224.
[26] K.L. Peña, S.E. Castel, C. De Araujo, G.S. Espie, M.S. Kimber, Structural basis of the oxidative activation of the carboxysomal γ-carbonic anhydrase, CcmM, Proc. Natl. Acad. Sci. U. S. A. 107 (2010). https://doi.org/10.1073/pnas.0910866107.
[27] D. Sehnal, S. Bittrich, M. Deshpande, R. Svobodová, K. Berka, V. Bazgier, S. Velankar, S.K. Burley, J. Koča, A.S. Rose, Mol∗Viewer: Modern web app for 3D visualization and analysis of large biomolecular structures, Nucleic Acids Res. 49 (2021). https://doi.org/10.1093/nar/gkab314.
[28] D. Morishita, M. Takami, S. Yoshikawa, R. Katayama, S. Sato, M. Kukimoto-

Niino, T. Umehara, M. Shirouzu, K. Sekimizu, S. Yokoyama, N. Fujita, Cell-permeable carboxyl-terminal p27Kip1 peptide exhibits anti-tumor activity by inhibiting Pim-1 kinase, J. Biol. Chem. 286 (2011). https://doi.org/10.1074/jbc.M109.092452.

[29] C.Y. Li, T. Di Wei, S.H. Zhang, X.L. Chen, X. Gao, P. Wang, B. Bin Xie, H.N. Su, Q.L. Qin, X.Y. Zhang, J. Yu, H.H. Zhang, B.C. Zhou, G.P. Yang, Y.Z. Zhang, Molecular insight into bacterial cleavage of oceanic dimethylsulfoniopropionate into dimethyl sulfide, Proc. Natl. Acad. Sci. U. S. A. 111 (2014). https://doi.org/10.1073/pnas.1312354111.

[30] P. Maur, Delaunay triangulation in 3D, Pdfs.Semanticscholar.Org. (2002).

[31] U. Bauer, Ripser: efficient computation of Vietoris–Rips persistence barcodes, J. Appl. Comput. Topol. 5 (2021). https://doi.org/10.1007/s41468-021-00071-5.

[32] L. McInnes, J. Healy, N. Saul, L. Großberger, UMAP: Uniform Manifold Approximation and Projection, J. Open Source Softw. (2018). https://doi.org/10.21105/joss.00861.

BIOGRAPHY

P. Ford Combs grew up in Norfolk, VA and attended Norfolk Collegiate School. He received his Bachelor of Arts from the University of North Carolina at Chapel Hill in 2011 and then traveled to Barcelona, Spain, where he lived and studied flamenco guitar at the Conservatori Superior de Musica del Liceu. He entered the George Mason University Bioinformatics and Computational Biology program in 2017 and earned his MS in 2018.