# Training of Template-Specific Weighted Energy Function for Sequence-to-Structure Alignment

by

En-Shiun Annie Lee

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

Threading is a protein structure prediction method that uses a library of template protein structures in the following steps: first the target sequence is matched against the template library and the best template structure is selected; secondly, the predicted three-dimensional structure of the target sequence is modeled by this selected template structure. The deceleration of new folds which are added to the protein data bank promises completion of the template structure library. This thesis uses a new set of template-specific weights to improve the energy function for sequence-to-structure alignment; this new weights improves the sensitivity of template selection step of the threading process. The weights are estimated using least squares methods with the quality of the modelling step in the threading process as the label. These new weights show an average 12.74% improvement in refining the modelling step. Further family analysis show a correlation between the performance of the new weights to the number of seeds in pFam.

**Acknowledgements**

I would like to thank my supervisor, Professor Ming Li; as well as my two readers, Professor Andrew K. C. Wong, and Professor Brendan McConkey for their guidance and helpful comments. I would also like to acknowledge Shuai Cheng Li and Babak Alipanahi from the Bioinformatics lab for defining and formulating the problem, as well as Professor Ali Ghodsi and WangHua Su from the statistics department for helpful discussions. I would like to thank my mentors, Professor Anne Condon, Christina Boucher, and Gary Li for their support in research. Finally, I would like to thank Charles Lee, Steve Wong, and Wilfred Kwok for proofreading this thesis.

## Dedication

This thesis is dedicated to my family and fiance for loving me and supporting my dreams. This thesis is also dedicated to the brothers and sisters at mCCF for their emotional and intellectual support.

Is there anything of which one can say, "Look! This is something new?" It was here already, long ago; it was here before our time. Ecclesiastes 1:10 (New International Version)

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The last century was coined the 'The century of the gene' [16]. The scientific community saw changes in the definition of a gene from an abstract element of heredity, to an open reading frames in sequence databank, to the proteom which is the functional products of the genome [20]. There were also significant scientific achievements in genomic bioinformatics in the past decade. Two milestones of genomic are the Basic Local Alignment Search Tool (BLAST) [26], the most cited paper of the 1990s, and the completion of the Human Genome Project using whole genome shotgun sequencing. The precedence for proteomic research is established the enormous genomic sequencing data generated from these advancements and the shifting definition of a gene.

With the turn of the century, the focus of bioinformatics also turned to proteomic. Proteins are indispensable to numerous intricate processes within the cell. They can aid complex tasks, for instance, by catalyzing biochemical reactions in

metabolism, or they can perform simple functions, for instance by scaffolding of cytoskeleton structures. Proteomics is the study of proteins, including areas such as protein structures and protein functions [2]. A protein is a linear chain wherein each link is built from one of twenty different amino acids. Through the interaction of amino acids in three-dimensional space, this linear chain bends and twists into a three-dimensional shape. Although the ordering of the amino acids is straightforward, the three-dimensional structure it forms is not. Proteomics emerges as a new and complex field with enormous research potential.

Scientists routinely decode protein sequences from gene sequences; however determining the three-dimensional structure of a protein remains labourous and time intensive. X-ray diffraction is used to find the structure of a protein, but first a protein crystal must be experimentally grown. The aggregation of protein moelcules into a crystal array devents on eviornmental conditions, and thus these experiments employ an exhaustive trial and error method which does not guarantee the formation of a crystal. The process is made more difficult by moving parts which requires a combination of nuclear magnetic resonance (NMR) and X-ray diffraction [6]. Laboratorial methods using X-ray crystallography or NMR spectroscopy for determining protein structures are time-consuming and expensive; thus, scientists turn to computers to help solve this problem.

## 1.1 Thesis Statement

In protein structure prediction, the algorithm is given a target sequence which does not have a known structure; the algorithm must predict a model of the target's structure. Threading, a protein structure prediction methodology, uses a library of template protein structures where the target protein sequence is molded into a template structure selected from the template library. The threading algorithm selects the best template for modelling by comparing the target sequence against each of the template structures. This comparison is performed by a sequence-to-structure alignment method with an energy function measuring the similarity of each template to the target.

Selecting a good template is essential to the quality of the three-dimensional structural model of the target sequence. This thesis identifies a template-specific weighted energy function to improve sequence-to-structure alignment for the template selection step of threading. The selection of an appropriate template structure is essential in the threading process and in the accurate modelling of the final predicted protein. The goal is to achieve a number of definite numerical improvements in the preditive ability of the energy function. More specificaly, the goal is to minimize the error distance between the energy function and the final predicted three-dimensional structure. Doing so will refine the template selection step of threading.

## 1.2 Motivation

The protein databank (PDB) is a database containing all proteins and it has grown rapidly in accordance with the genomic advancements. In 1960, the first three-dimensional structure of a protein was determined by x-ray crystallography; the present day PDB contains more than 500,000 entries. Structural biologists estimated that there are as few as 2000 unique protein folds with 50% of the folds estimated to have known structures already [6]. The deceleration in the number of novel folds hints that a sufficient structural database has been collected [18]. This holds the promise that a computational learning from existing protein structures can eventually predict the structure of a protein solely from its sequence.

Protein structure prediction is a difficult problem that still remains unsolved. In fact, the bi-annual CASP (Critical Assessment of Techniques for Protein Structure Prediction) competition gathers top research labs from around the world to evaluate the accuracy of their protein structure predictors. Protein structure prediction methods can be broadly divided into two types: comparative modelling and ab initio. In the comparative modelling method of protein structure prediction, such as RAPTOR, each template in a template library is evaluated with a scoring function to determine its similarity to the target sequence. In ab initio method of protein structure prediction, such as PROSPECTOR and ROSETTA, each decoy structure in a decoy set is evaluated with an energy function to determine its closeness to the native structure. All of the leading protein structure predictors, namely, PROSPECTOR, ROSETTA, and RAPTOR, use a database containing protein structures with a

metric measuring the quality of the database's elements.

The success of threading is directly influenced by the coverage of its template library. If the correct template is not in the library, then the resolution of the target structure generated will be of low quality. As the number of novel folds approaches towards completion, the possibility of a complete template library becomes possible. Accurate methodologies to assess the template structures and a metric to measure the template quality is essential. This thesis refines threading by focusing on the energy function for structure-to-sequence alignment in the template selection step of the threading process.

## 1.3 Objectives

The computational machine learning technique used in this thesis is the Least Squares method. This method estimates the terms from the energy function in template selection as the quality of the resulting model from threading . The new template-specific weights improves the energy function by allowing it to select a best fit template for the modelling step in the threading process. This improvement is quantified by comparing the new template-specific weighted energy functions against the old generally weighted energy functions. The two objectives to be achieved by this thesis are:

- template-specific weighted energy fucntions improves template selection
- relationship between family properties and the performance of the new template-

specific weights

## 1.4 Contributions

Each fold discovery movees researchers closer to the completion of all unique folds in the PDB. As the number of folds in PDB increase, the databases essential to comparative modelling method also draw closer to the completion. There is a need for a better mathematical definition to represent the database. Energy functions are an essential part of the ab initio methodologies of protein structure prediction; statistical scoring functions trained on final native structure are also important in comparative modelling. A candidate is the energy function of sequence-to-structure alignment where the alignment measures template quality.

Machine learning method is applied to the template library to improve the template selection step of threading. The method evaluates the energy function of the template selection step to assess the quality of the model generated by the modelling step. The method employs energy minimization, a method commonly found in ab initio methods, by closing the gap between the real energy value and the calculated mathematical energy value. More specifically, the Least Squares method is used to find template-specific weights for localized predictions.

The work in the thesis provides three new perspectives to refining the following energy functions and methods:

1. Local prediction function - Previous methods apply overall global weights to

energy and scoring function for ab initio method and comparative modelling. This thesis examines local weights applied to each template in the template library of threading.

2. Label in the Machine Learning method - Most scoring functions of comparative modelling are trained based on the final protein structure, the real native three-dimensional model. The m method in the thesis incorporates the steps of the modelling process as the final label.

3. Family analysis - The improvements of template-specific weights correlate with the number of seeds in pFam.

The work in this thesis examining the scoring function of the sequence-to-structure alignment in a broader perspective taken from energy landscaping.

## 1.5 Outline

First this thesis will present a survey of the energy function used in threading and the leading protein structure predictors. Next, the machine learning problem is mathematically defined and formulated. This thesis then presents results by comparing new specific weights determined by machine learning technique against the old general weights and discusses the findings obtained from analyzing the family properties of the template protein. Finally, the conclusion presents findings and significance, andproposes future work.

# Chapter 2

# Background

The background chapter is designed to include only the minimal biological background required to understand the premise of this thesis. The first section begins broadly with a principle which applies to all organisms: the central dogma of molecular biology. The protein is described in terms of its static structural organization and dynamic functional categorization. The next section breraks down the level of protein structure organization, i.e. a hierarchical abstraction building protein from primary to secondary to tertiary to quaternary structures. Then the following section delves into the biophysics topics of protein folding such as energy landscape and individual fold units like protein domains.

## 2.1 Protein Biochemistry

### 2.1.1 The Central Dogma of Molecular Biology

The three biological sequences that encode the functions of life are DNA, RNA, and protein. Today, the full human genome sequences has been decoded, but decoded remains disorganized. The challenge is to determine the protein structure that demonstrates the function behind this genomic sequence information. It is the final product of the protein sequence, a protein structure, that enables the complex functions within the cell.

The Central dogma of molecular biology demonstrate how information from four letters alphabet, or four different nucleotides, of DNA flows to RNA and turns into the twenty letter alphabet, or twenty different amino acids, of protein. The dogma is central because its processes occur in all living cells. The central dogma of molecular biology describes how DNA, RNA, and protein form from one to the other: DNA either replicates itself or is transcribed into RNA intermediary molecule; RNA groups into three letter code, or codon, and translates to protein [6]. The focus of this thesis is on protein structure prediction, thus the next section will focus on the third macromolecule, the protein.

### 2.1.2 Levels of Protein Structure Organization

To introduce the protein, this section considers the static components of protein structure while the next section consider the dynamic aspect of protein folding. Pro-

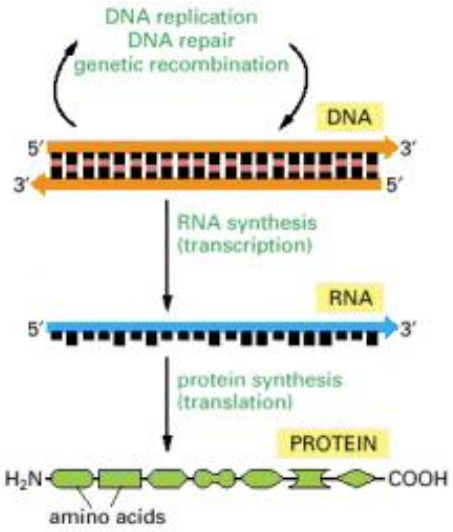


Figure 2.1: Central Dogma of Molecular Biology
Central dogma of molecular biology describe the flow of genetic information from DNA to RNA (transcription) and from RNA to protein (translation).

tein can be divided into structural subunits: from the fundamental building block of amino acid, to the primary sequence, all the way to the quaternary structure. The energy of protein folding forces it into its structure, this energy is important to the abstraction of structural classification.

### Amino Acid as the Fundamental Building Block of Protein

Amino acids are the smallest building blocks that assemble together to create proteins. It is defined by one carboxylic acid group and one amino group; thus it is called amino from the amino group and acid from the carboxylic acid group. The amino acid is anchored by a central alpha-carbon, which connects the amino group and the carboxylic acid group, in addition to a hydrogen and a variable side-chain (R). Twenty different possible side-chains can attach to the central alpha-carbon atom of the amino acids. Each possible amino acid has its own set of distinct

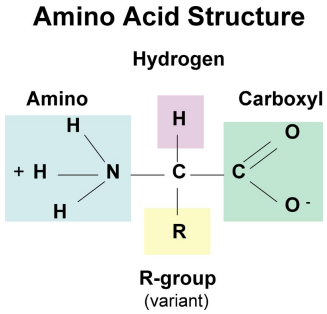


Figure 2.2: Chemical Formula of an Amino Acid
The chemical formula of an amino acid, which consists of a central alpha-carbon connected by a amino group, a carboxylic acid group, a hydrogen and a variable side-chain(R).

properties, such as hydrophobic or hydrophilic, charge or uncharged, acid or base , bulkiness, and many others.

### Polypeptide Chain as the Primary Structure

The first level of protein structure organization is the primary structure, which is a linear sequence of amino acids chained together into a polypeptide chain. Two amino acid are joined by a peptide bond and when multiple amino acids are joined head-to-tail into a long chain, a polypeptide is created. Along the core of peptide chain is the polypeptide backbone consists of repeating sequence of carbon and nitrogen atoms. A polypeptide has definite direction with endings: the amino end (NH2)of polypeptide is called the N-terminus, and the carboxyl (COOH) end of the polypeptide is called the C-terminus.

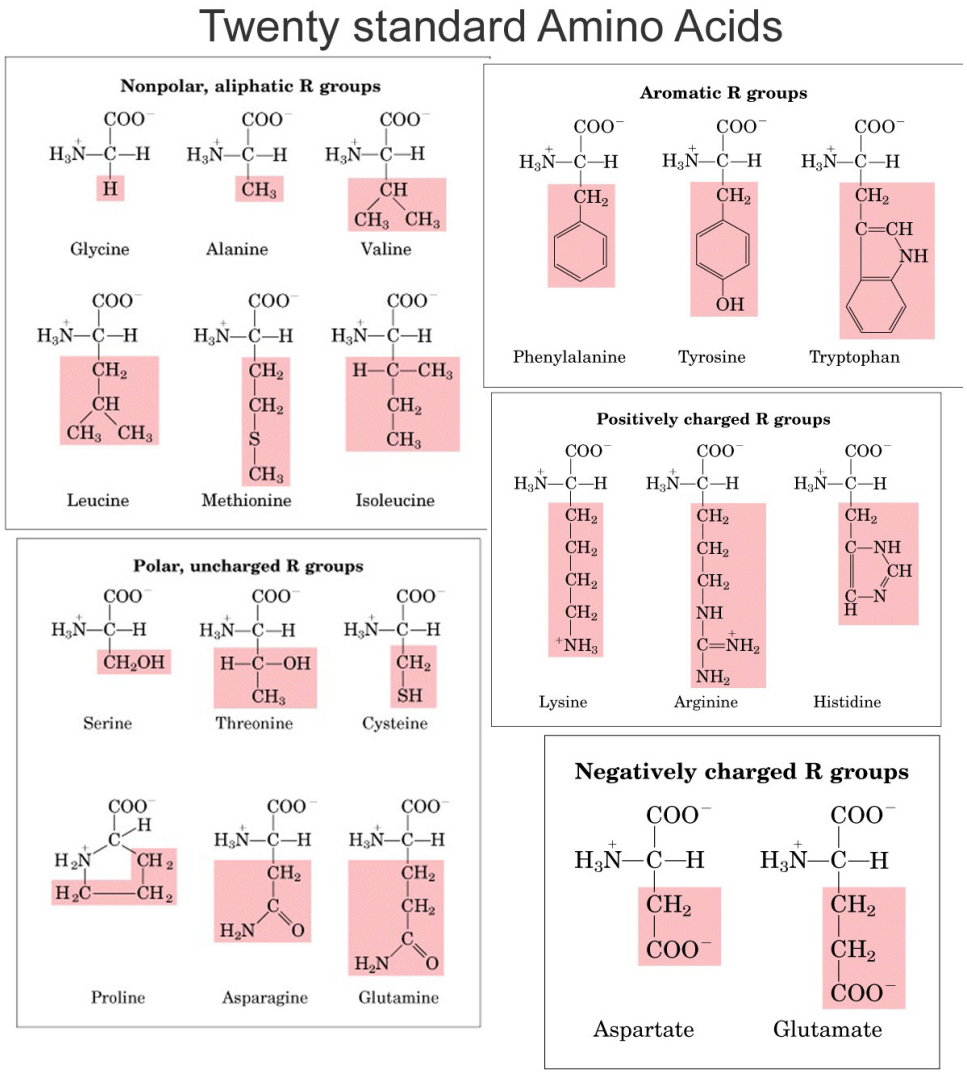


Figure 2.3: Twenty Types of Amino Acids
There are twenty unique types of amino acid, etch with its unique side-chain group, which determined by the nature of its side-chain.

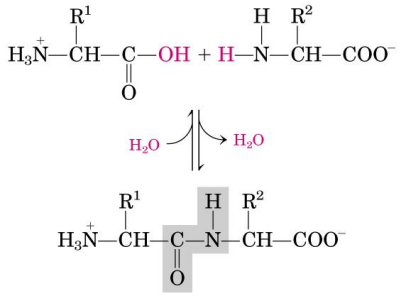

Figure 2.4: Peptide Bond
Two amino acids react with one another to give off one water and forms one peptide bond.

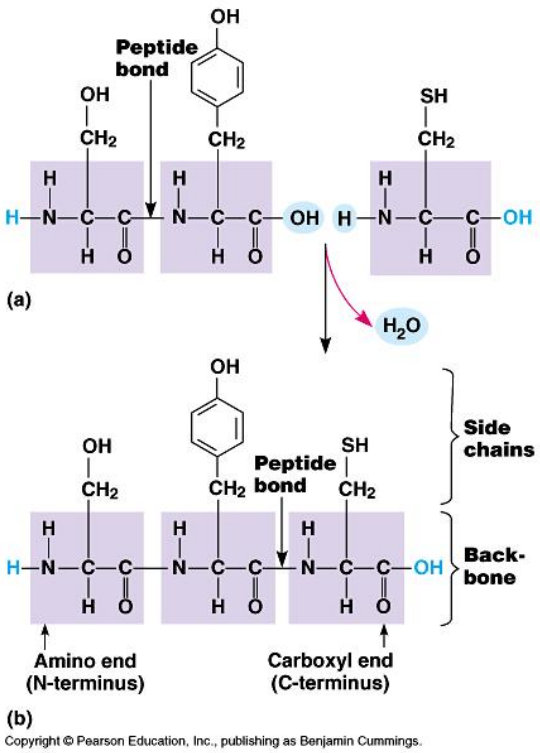



Figure 2.5: Polypeptide Chain
Multiple amino acids are chained together by multiple peptide bonds to form a polypeptide chain.

**Hydrogen Bonds Form Regular Substructures as Secondary Structure**

The polypeptide chain forms the primary structure, which interacts with its own three-dimensional space to form substructures. The hydrogen bond is an interaction between the N-H group of one amino acid and the C=O group in another amino acid, both amino acids are from polypeptide backbone of the chain. Because it does not involve the variability of side-chain characteristics, hydrogen bond is a widely common interaction without needing specificity the exact side-chain. A regular repeating conformation of these hydrogen bonds form two regular fold patterns: the alpha-helix and the beta-sheet.

The three secondary structures are: alpha helix, beta-sheet, and loop. First, the alpha helix appears like a twisted telephone cord with regular hydrogen-bond between every first and fourth residue. In this way, the $i^{th}$ amino acid forms a hydrogen bond locally with the $i + 4^{th}$ amino acid. The alpha helix is a simple regular structure which forms a complete turn every 3.6 amino acid. The beta sheet looks like a sheet where segments of the polypeptide chain line up next to one other and form hydrogen bonds. Its hydrogen bonds are between two distant strands of the polypeptide chain running side by side. If these two strands are going in the same direction, then the beta-sheet is called parallel. If one strand folds back on itself on the second strand causing them to go the opposite direction, then the beta-sheet is called anti-parallel. Two strands in a beta-sheet may be far away from each another in the sequence, making it more difficult to determine. Compared to the simple local hydrogen bonds in alpha helix, the distance between beta-strands

causes the prediction difficulty. Finally, loop has no definite structure, and usually links other structures.

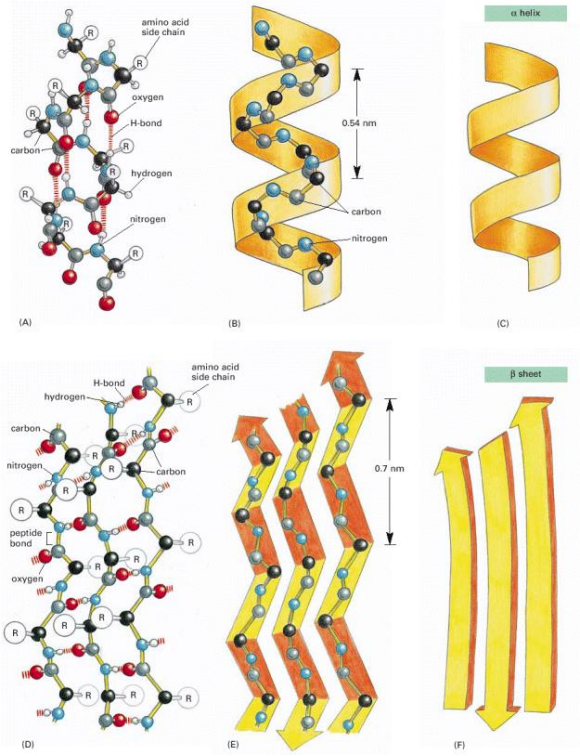


Figure 2.6: Two Secondary Structures with Hydrogen Bonds
Alpha helix regularly forms hydrogen bonds between every fourth amino acid. Beta sheet forms hydrogen bonds with two distant segments within the polypeptide sequence running beside each other in three-dimensional space.

### Tertiary and Quaternary Structures as Higher Structural Organizations

A tertiary structure is a polypeptide chain formed by secondary structures assembled into a full three-dimensional structures. Quaternary Structure is a complex protein built from subunits of multiple tertiary structures which are folded polypeptide chains.

### 2.1.3 Protein Folding

Current protein structures show the importance forces play on dictating the final structure the protein folds into. This section introduces the dynamic aspect of protein folding through its forces and energies.

The three significant forces which effect how a protein folds are spatial restrictions, non-covalent bonds, and hydrophobic forces. The restrictions on three three-dimensional arrangement of atoms includes overlapping spherical radius, van der Waals radius, and bond angle. These forces define what conformation is physically feasible for an amino acid to take. The second major force, non-covalent bonds, refers to interactions between two amino acids, such as hydrogen bond, ionic bond, and van der Waals attractions. Finally, hydrophobic forces occurs between the protein and its environment. When the protein is in an aqueous environment made up of polar water molecules, the water molecules form hydrogen bonds with hydrophilic amino acids in the protein. Thus the amino acids, variable by the side-chain characteristics, cluster by their hydrophobic character. The hydrophilic amino acids cluster on the exterior to bond with the water molecules; accordingly, the hydrophobic amino acids cluster in the interiors to escape from the water molecules.

The forces on amino acids in the protein chain described above, causes the protein chain to fold into a three-dimensional structure. Each possible three-dimensional structure has a specific energy conformation. This final folded structure is called native conformation, and is characterized by a minimized free energy. Typically, each unique protein has its own stable conformation; however, this singularity

may change as the protein interacts with other molecular. The first contribution of this thesis considers local energy minimum of template-specific weights for the energy function instead of a globally weighted energy minimum.

### 2.1.4 Protein Classification by Stable Conformation

A protein domain is any part of the polypeptide chain that folds independently into a structural subunit. A domain is a stable substructure; thus, its role in terms of free energy is central in the organization of protein structure. Physically, a domain is between 40 to 350 amino acids and is the modular unit from which many larger proteins are constructed. Small proteins may contain only a single domain, while larger proteins may consists of up to several dozen domains connected by short unstructured loops. A protein is typically 50 to 2000 amino acids long and consist of several distinct protein domains. Different domains of a protein are often associated with different function. Initially, a protein sequence folds into a stable three-dimensional conformation with functional properties. A mutation in the amino acid during evolution caneither be discarded as neutral or result in a new structure and a new functional fold. The numerous protein functions are grouped into protein families, where each family member resembles the other through its amino acid sequence and three-dimensional structure. Another functional classification of the protein is the protein fold, which is a combination of the fundamental folding element alpha-helix and beta-sheet. Structural biologists estimated that there are as few as 2000 protein folds with 1000 already known. With 50% of the

folds already discovered, biologists have concluded that there are a limited number ways protein domains can fold independently. The third contribution of thesis considers the classification of proteins by the functional groups and the effect this has on the localized weights.

## 2.2 Protein Structure Prediction

The protein structure prediction problem is stated as follows: Given a target sequence whose structure is not known, it attempts to build a predicted protein structure. This problem can be solved from scratch without knowing the existing protein structures, such as ab initio modelling, or it can solved with the help of a database of existing protein structures, such as those provided in homology sequence search or comparative modelling.

### 2.2.1 Comparative Modelling

Comparative modelling, or homology modelling, predicts three dimensional structure of an sequence by using a known template structure library. A template is a protein with known structure which can be computationally compared to the target sequence to determine if they are similar. The target protein is a known sequence whose structure is unknown [13]. The four steps of comparative modelling are listed below [12].

1. select a template – identify the possible known structure that is homologous

to the unknown sequence by searching the template structural library

2. Align target sequence to template structure – build an alignment of the unknown protein sequence from the selected template structure

3. Build backbone – model the target sequence's three-dimensional structure after the structure of the selected template

4. Construct loops and attach side-chains

The first two steps combined presented above is the threading method of protein structure prediction, which is discussed further in the next section [12]. The prediction target structure is sensitive to the following two influences: the quality of the template library and algorithm's ability to select the correct template. The deceleration in the novel folds discovered foreshadows the sufficient collection of a structural database [18]. The other limitation of this prediction method is the fold recognition problem introduced in a later section.

**Fold Recognition**

A Protein Fold is defined as similar structures created by sequences that may be either the same or different. In fact, two proteins may each have a different function and structure, yet still have similar folds. Traditionally, identifying whether a protein belongs in a fold is accomplished either by comparing sequence similarity or determining sequence-to-structure compatibility. This sequence-to-structure comparison is also known as fold recognition [12]. When given a sequence, a fold

recognition algorithm will determine the proper structural fold or family from the database of known structures [13, 5]. Fold recognition is often a key and integral step in protein threading. Threading can be considered as energy-based fold recognition; there, an energy function is used to determine the template protein structure to be used to predict the structure of the unknown protein sequence [5].

### 2.2.2 Threading

Threading can be seen as an increment of the fold-recognition problem. In addition to identifying the fold of a target protein sequence, the threading algorithm must also model a three-dimensional target structure based on the template structure. Threading received its name from fitting the target sequence to the structure of the template protein, a process that slightly resembles stitching onto an embroidery pattern. In this way, the process matches a target sequence with unknown structure to a library of known template structures.

Threading makes up the first two steps of comparative modelling [5]: template selection and sequence-to-structure alignment. The template selection step is the fold-recognition protocol for the purpose of threading. Typically it involves the following details: sorting the scores of the energy functions; ranking the target-template alignments; choosing the best scoring alignment as the template; and as the fold class, and creating a meaningful model for unknown structure from the selected template.

The majority of threading methods differ in the following three processes[5]:

1. Basic model of the protein and interaction of the amino acids

2. Energy parameterization

3. Alignment algorithm

The first template selection step is limited by the quantity of the structural template library; the template library must contain the correct template. The second alignment step is limited by the quality of the template; this quality refers to the template's similarity to the target sequence [5].

The target-template alignment step is typically known as the threading step. The energy function of the alignment is important because the quality of the alignment match is evaluated with sequence-structure similarity energy function [13, 9]. The next section below narrows the focus on the second limitation: the energy function of the target-template alignment.

**Energy Function Measures the Quality of Sequence-to-Structure Alignment**

Global sequence alignment is also called the Longest Common Subsequence problem in the field of theoretical algorithms. The sequence alignment uses a dynamic programming algorithm that scores the alignment locally by comparing residues at a single column position. Scores are based on measures such as gaps, matches, and mismatches. Unlike sequence alignment, which compares two sequences based on pairwise matching, threading compares a structure to a

sequence based on complex properties. Threading attempts to measure non-local distant interactions and thus measures the quality of a template by complex energy function [5]. Ideally, GLOBAL SEQUENCE ALIGNMENT is desired; unfortunately, only heuristics or approximations are possible to date.

It is NP-complete to obtain the optimal solution for aligning regions that are not sequential, thus only an approximate solution can be achieved. Some of these approximation algorithms include aligning non-local scoring functions, two-level dynamic programming, and frozen energy [5]. Consider two types of approximation algorithms: frozen approximation and defrosted approximation. In frozen approximation, the target side-chain are simply the template side-chains for calculations. In the defrosted method, template side chain are replaced by the target side-chain before the contact score is calculated. The inevitable trade-off between these two methods is accuracy versus speed and computational power [18].

There are many metrics for measuring the quality of a sequence-to-structure alignment. One metric is the position-specific score. Computing such score sometimes require first replacing side-chain of one amino acid with side-chain of all possible amino acids. Another metric is to use statistical contact potentials to compute the resulting substitution scores. Yet another metric to improve alignment accuracy, specifically for sequences with low similarity, is the Insertion/Deletion (Indel) frequency array [15]. Other factors that have been considered include predicted secondary structure and burial preference [18].

# Chapter 3

# Survey

This chapter explores the contemporary leaders in protein structure prediction: TM-Score, Rosetta, and Raptor. But before examining these servers, this chapter familiarizes with the context leading up to threading methodology. The first section gives an overview of protein structure prediction methods. Next, in the same section, it narrows in the threading process and its details. Finally, it examines the role of the energy function in the template selection step of threading. After the problem of interest, weighted energy function in sequence-to-structure alignment for template selection step of threading, the survey protein examines the leading predictors. Each of the top predictor is briefly described and their statistical measurement of the energy function is detailed.

## 3.1 Energy Function of Leading Predictors

Two types of energy functions are used in protein structure prediction: physics-based and statistics-based. The physics-based energy function relies on quantum mechanics and molecular mechanics; for both mechanics a conformation change is calculated on an atomic model. The statistic-based energy function depends on existing knowledge of the native protein structures, as when the frequency of propensity distribution is calculated in a set of protein structures [28].

Energy function consists of energy terms, which may be sequence-independent or sequence-specific [28]. Sequence-specific energy terms can be further broken down into local within the sequence or non-local within the sequence. Non-local sequence-specific energy terms are not next to each other in the linear sequence, rather they are local in terms of three-dimensional space. For example, secondary structures such as an alpha-helix or a single beta-strand are local; however, the hydrogen-bonds of a beta-sheet are non-local. Thus beta-sheets are more difficult to predict due to the poor modelling of hydrogen-bonds.

$$E = \sum E_i$$

Weights corresponding to energy terms are adjusted to optimization the energy function. These weights are optimized to ensure the closeness of the energy function to the structural quality of the predicted structure (or the degree of nativeness of

the predicted structure) [28].

$$E = \sum w_i E_i$$

This next section explores the energy function of the three top protein structure prediction methods: ROSETTA, TM-SCORE, and RAPTOR.

### 3.1.1 ROSETTA

ROSETTA predicts protein structure by using local sequence preference bias and the conformation of non-local interaction. Energy and scoring functions are integral components at each step of ROSETTA's prediction, such as the steps of initial model building, decoy selection, and fragment assembly. The first use of energy function in the initial model building; this is where the model of a fragment of sequence is built. Here, the energy function is a fine atomic resolution of physical energy terms [8]. These are a physical energy terms, which are not as relevant to this thesis as the statistical energy terms.

The second way an energy function is used by ROSETTA is determining the native conformation from a decoy set. A native structure is the true structure of a protein; a decoy is a predicted structure close to the native structure. ROSETTA generates many possible predicted decoy structures to form a decoy set, where then the best conformation is chosen. The best way to evaluate the energy function is to test its ability to recognize near native conformations in large sets of decoy

structures [22]. The most accurate energy function should assign the native conformation a lower energy than the decoy. The energy function of decoy selection achieves the following 4 goals: (1) variety, (2) 'close' conformations, (3) reasonably near local minimal, (4) unbias to native information [14]. The native z-score is used an alternative measure which measures the numerical standard deviation between the native structure energy and the average energy decoys. For example, if z-score is positive, then native is less in terms of energy than average decoy. If z-score is negative, then the opposite is true. The energy gap between the native structure and decoy structures is optimized in the following energy minimization formulation [34]:

$$Optimize\left(totalEnergy - \sum_{\forall decoys} similarityMeasure(native, decoy)\right)$$

The energy function is further weighted to allow for non-independent overlaps. Logistic regression is used to obtain the relative weights for the energy terms.

Lastly, consider the energy function for fragment assembly in ROSETTA. The categorization of the energy function can be thought of in these two ways [17]:

1. sequence dependent and sequence independent terms - the separation between these terms divides the problem into manageable subproblems

2. internal and external to the environment - the combination of environmental effects with pairwise effects allows the probability to be expanded independently

The implementation of environment variables into the energy function struggles with trade-off between increased in quality and reduced amount of data; both concerned are addressed with all variables defined fully in the equation as a binary cut-off. Another concern is the noise in the energy function which can convolute the fit of the ab initio conformation. This is because different sequences in a multiple sequence alignment, which is the alignment of many sequences, is not independent; an is solved by normalize the size of each family. [17]

### 3.1.2 TM-Score

Template Modelling Score, TM-Score, is a measure of template quality that is size normalized and does not use a cutoff like its predecessors do. The template assessment problem is designed to find the resulting confidence of the alignment when given a sequence-to-structure alignment and its evaluation schemes. TM-Score accomplishes two goals [35]:

1. The inherent problem of the score being correlated by protein size is solved using a p-value.

2. Instead of using a distance cutoff or threshold to partition the structures, TM-Score uses all residues and the complete range of Z-Score values.

TM-Score overcomes the drawbacks of RMSD coverage. For example, $2\mathring{A}$ RMSD with 50% alignment coverage under-performs $3\mathring{A}$ RMSD with 80% alignment coverage. The RMSD may be better; however, the coverage is not as good [35]. Following

the threading steps, the following implementation steps were taken:

1. threading using PROSPECTOR3
2. modelling using MODELLER; later, TASSER was used to show the same results

TM-SCORE claims to have a closer correlation between the final model and initial template alignment. When compared to other metrics, TM-SCORE has distinct advantages [35]. For example compared to:

- MAXSUB: TM-SCORE includes high and low accuracy-aligned regions
- RMSD: TM-SCORE is an average, resulting in equal weighting for all amino acids
- GDT-SCORE: TM-SCORE weights high and low accuracy regions differently

### 3.1.3 RAPTOR

RAPTOR is a threading algorithm which aligns local segments of a global sequence by assuming these segments are independent and identically-distributed random variables. Rearranging these segments are considered NP-hard; thus, RAPTOR uses an approximation method called integer linear programming (ILP). RAPTOR credits its performance to the ILP formulation for computing optimal solutions for the energy functions. Other programs sacrifice this accuracy for computational efficiency and speed [31].

RAPTOR uses support vector machine (SVM), another machine learning method, as fold recognition to recognize the best template for a given target. Given a sequence-to-structure threading pair and a positive instance of pairs within the same fold in SCOP; a prediction model is trained. These features include z-score, energy items, alignment length, total gap length, number of template contacts with both ends aligned, number of template contact with only one end aligned, template size, sequence size [31]. The training data is 95 structurally-aligned protein pairs from similar fold-level. The method compares RAPTOR alignment with SARF structure alignment and deemed the alignment correct if it is four amino acid shifts away. After 300 rounds of the genetic algorithm with local pattern search method with 40 random seeds to find the optimal values of other weight factors [32]. This overall accuracy is further measured as the ratio $\frac{correct}{max}$, where, $correct$ =number of correctly-aligned positions of all threading pairs and $max$ =number of maximum alignable positions. The results for the training error is 56% alignment accuracy which becomes 50% when using generated Holm's test set.

Weight factors are used for the energy function terms to optimize alignment accuracy. Weight factors only effects alignment accuracy by 10%, which makes alignment accuracy quite robust to weight factors. Also when weight factors are classified by the alignment similarity of training set to obtain 6 to 10 sets of weight factors, the accuracy improve to approximately 70%. Lastly fold recognition can be improved by approximately 2% by using the z-score to rank the templates.

## 3.2 Correlating Protein Alignment with Categorization

This section examines the connection between structure and sequence alignments through the classification of family and super-family. The structure alignment used is CE-ALIGN and FSSP; family and super-family classification are defined by SCOP. [12]

The scores are normalized by family size. Results show that typical trends of poor structure alignments are due to the following three reasons: abnormally large insertion, incorrect secondary structure alignment, SCOP classification [12]. The three-dimensional accuracy of a protein structure alignment is not absolute; different model of assessment result in variation in server rankings. There is no standard algorithm due to the lack of similar definition of two native protein structures. For example, in classification SCOP, CATH, and FSSP often disagree on weakly similar proteins. Sometimes evolutionary relations are hypothetical at best. [18]

Structural divergence is slower than sequence mutation, thus predictions make use of non-sequence-based methods such as fold-similarity and structure-based alignments. [18] This conclusion is reached due to the following two observations:

1. proteins with similar structure, yet negligible sequence similarity demonstrate convergent evolution where different proteins are in the same fold

2. the spatial arrangement of amino acids results in interaction preference and contact potential; this is manifested in the contact-based scoring matric

# Chapter 4

# Methodology

This chapter provides the necessary background knowledge to understand the mathematical formulation to the problem at hand. The first section introduces the necessary basics to machine learning and an existing proof validating the linear least squares method. The next section defines a least squares formulation to energy function of sequence-to-structure alignment. First, terminologies for describing the protein structure prediction problem are explained. The processes in threading are formally framed as algebraic formulae and machine learning variables. Finally, this chapter describes the exact procedures for the computational experiments. It details the steps at obtaining raw data, executing the algorithm, and evaluating the results.

## 4.1 Introduction to Machine Learning

Machine learning is a subfield within the discipline of artificial intelligence and it deals with heuristic algorithms that imitates learning. Supervised learning extract a PREDICTION RULE from an existing set of training data; the training dataset contains pairs containing input parameters call instances and desirable outputs called labels. This thesis uses regression, where the output is a continuous value and uses a set of local learning models rather than one global model [1].

In machine learning, the algorithm is given a set of labeled training examples where each the label is associated with an instance. the machine learning algorithm is applied to determine a PREDICTION RULE; the rule is then used on new testing examples where instances do not have corresponding labels. The PREDICTION RULE should produce predicted label consistent with the given instances of these new examples.

**Problem Definition 4.1.1** ***G****iven a data set containing $m$ data pairs $(\vec{x_i}, y_i)$, for all $i = 1, ..., m$ like $(\vec{x_1}, y_1), (\vec{x_2}, y_2), ..., (\vec{x_m}, y_m)$ where $\vec{x}$ is a $n$ dimensional vector. Let $\vec{x_i} \in X$ be the independent variable representing instances, and $y_i \in \Upsilon$ be the dependent variable, representing labels associated with $\vec{x_i}$.*

***F****ind* PREDICTION RULE*, which is a model function $f : X \longrightarrow \Upsilon$ that maps $X$ to $\Upsilon$; Let $\hat{y}_i$ be the predicted label of $y_i$ resulting from the* PREDICTION RULE*, $f(\vec{x_i}, \vec{w})$.*

The PREDICTION RULE's function also contains $n$ adjustable parameters stored by the vector $\vec{w}$. **F**ind the best parameter values for $\vec{w}$ for $f$, the PREDICTION

RULE, so that the predicted $\hat{y}$ label is as close to the real $y$ label as possible. This can be done by minimizing the LOSS FUNCTION, $\ell(y, \hat{y})$, which is the error of the predicted labels against the actual real labels.

### 4.1.1 Linear Least Squares Method with Weights

Least square method is commonly used for finding best-fitting curve to a given set of points. Two types of LOSS FUNCTION exists: perpendicular and verticle. In least squares method, verticle is used because of its independence from $x$. In linear least squares method with weights, the PREDICTION RULE is a linear function with weights and the LOSS FUNCTION is the residual squared: $\ell(y, \hat{y}) = \sum r^2$, where $r = (y - \hat{y})$. This is done by minimizing the sum of the squared residuals where the residual is the offsets from the true curve. Here, 'square' instead of 'absolute' difference is used on the residual to allow continuous differentiable quantities. However, this also causes outlying points to have a disproportional effect. The mathematical formulation for deriving the linear least squares method with weights is calculated.

**Formulation of Least Squares Method with Weights**

Let the LOSS FUNCTION be the sum of residual squared:

$$
\begin{aligned}
R &= \sum_{\forall i} (r_i)^2, \text{ where } i = \text{data points} && (4.1)\\
&= \sum_{\forall i} (y_i - \hat{y_i})^2, \text{ substitute } r_i = y_i - \hat{y_i} && (4.2)\\
&= \sum_{\forall i} (y_i - f(\vec{x_i}, \vec{w}))^2, \text{ substitute } \hat{y_i} = f(\vec{x_i}, \vec{w}) && (4.3)\\
& && (4.4)
\end{aligned}
$$

Differenciate with respect to $w$ and set to zero to find the distance minimal:

$$
\begin{aligned}
\frac{\partial R}{\partial w} &= 0 && (4.5)\\
\frac{\partial \left(\sum_{\forall i} (r_i)^2\right)}{\partial w} &= 0 && (4.6)\\
\sum_{\forall i} 2 r_i \frac{\partial (r_i)^1}{\partial w} &= 0 && (4.7)\\
2 \sum_{\forall i} r_i \frac{\partial r_i}{\partial w} &= 0 && (4.8)\\
& && (4.9)
\end{aligned}
$$

Aside: differentiate $r_i$:

$$\begin{aligned}
\frac{\partial r_i}{\partial w} &= \frac{\partial(y_i - \hat{y}_i)}{\partial w}, \text{ substitute } r_i = y_i - \hat{y}_i && (4.10)\\
&= \frac{\partial(y_i - f(\vec{x_i}, \vec{w}))}{\partial w}, \text{ substitute } \hat{y}_i = f(\vec{x_i}, \vec{w}) && (4.11)\\
&= \frac{0 - \partial f(\vec{x_i}, \vec{w})}{\partial w} && (4.12)\\
&= -\frac{\partial f(\vec{x_i}, \vec{w})}{\partial w} && (4.13)\\
&\text{Suppose that the PREDICTION RULE is a linear function} && (4.14)\\
&\text{substitute } f(\vec{x_i}, \vec{w}) = \sum_{\forall j} X_{ij} w_j \text{ where } j = 1, ..., n \text{ vector parameters of } w && (4.15)\\
&= -\frac{\partial(\sum_{\forall j} X_{ij} w_j)}{\partial w} && (4.16)\\
&= -\frac{\partial(X_{i1} w_1 + X_{i2} w_2 + ... + X_{in} w_n)}{\partial w} && (4.17)\\
&= -X_{ij} && (4.18)\\
& && (4.19)
\end{aligned}$$

Plug back into original equation:

$$
\begin{aligned}
2\sum_{\forall i} r_i \frac{\partial r_i}{\partial w} &= 0, \text{ substitute } r_i = -Xij && (4.20)\\
2\sum_{\forall i} (r_i)(-X_{ij}) &= 0, \text{ substitute } r_i = y_i - \hat{y}_i && (4.21)\\
-2\sum_{\forall i} (y_i - \hat{y}_i)(X_{ij}) &= 0, \text{ substitute } \hat{y}_i = f(\vec{x_i}, \vec{w}) and && (4.22)\\
&\text{, substitute } f(\vec{x_i}, \vec{w}) = \sum_{\forall j} X_{ij} w_j && (4.23)\\
-2\sum_{\forall i} (y_i - \sum_{\forall k} X_{ik} w_k)(X_{ij}) &= 0 && (4.24)\\
-2\sum_{\forall i} y_i X_{ij} + 2\sum_{\forall i} X_{ij} \sum_{\forall k} X_{ik} w_k &= 0 && (4.25)\\
-2\sum_{\forall i} y_i X_{ij} &= -2\sum_{\forall i} X_{ij} \sum_{\forall k} X_{ik} w_k && (4.26)\\
\sum_{\forall i} y_i X_{ij} &= \sum_{\forall i} X_{ij} \sum_{\forall k} X_{ik} w_k && (4.27)\\
YX^T &= (X^T X)^{-1} w && (4.28)\\
& && (4.29)
\end{aligned}
$$

## 4.2 Energy Function Defined in Least Squares

### 4.2.1 Defining Variables and Functions Used

Proteins are represented either as a linear sequence of strings or a thee-dimensional structure. A protein, $p$, is represented by the linear sequence, $p_{seq}$, and by the three dimensional structure, $p_{struct}$, where $p_{struct}$ is the true structure and $\hat{p}_{struct}$ is the predicted structure. The goal of protein structure prediction is to acquire the final predicted structure, $\hat{p}_{struct}$, from the initial sequence, $p_{seq}$. In machine learning,ˆrep-

resents the predicted label, as oppose to the true existing label. Two proteins align when amino acids of one protein matches to the amino acids of the other protein. A pair of matching, either by sequence character or by structural super-positioning, is referred to as pairwise or column-by-column matching and is matched. There are many different types of alignments, such as sequence alignment, structural alignment, and sequence-to-structure alignment. Let $Align_{seq}(p_{seq}, q_{seq})$ represents the sequence alignment between two protein sequences $p_{seq}$ and $q_{seq}$, where the accuracy of this type of sequence alignment is measured by the alignment score. Let $Align_{struct}(p_{struct}, q_{struct})$ be the structural alignment between two proteins with known structures, $p_{struct}$ and $q_{struct}$, where the quality of a structural alignment is determined by its Z-Score. Let $Align_{seq-struct}(p_{seq}, q_{struct})$ be the sequence-to-structure alignment between two proteins one without known structure, $p_{seq}$, and one with known structures, $q_{struct}$. The quality of a sequence-to-structural alignment varies depending on the purpose this type of alignment is used for, in the case of this thesis TM-Score is used.

| Alignment Type | Input Proteins | Output Measure |
|---|---|---|
| $Align_{seq}$ | $p_{seq}, q_{seq}$ | alignment score |
| $Align_{struct}$ | $p_{struct}, q_{struct}$ | Z-Score |
| $Align_{seq-struct}(p_{seq}, q_{struct})$ | $p_{seq}, q_{struct}$ | TM-Score |

Table 4.1: Alignment Definition
Table containing the alignment types and their input proteins as well as output measures of its alignment's quality.

### 4.2.2 Mathematical Description of Threading

This section narrows the focus down to the mathematical description of threading. First the section connects the machine learning method to the linear least squares with weights, which was introducted in the the beginning of this chapter. Next this section extracts existing work on energy minimization and localized prediction model, which will be presented in the Survey chapter. The goal of protein structure prediction is to create a predicted three dimensional structure, $\hat{s}_{struct}$, of the target sequence $s_{seq}$. In threading the target protein sequence is compared against each template protein structure in the template library $T$; the *best* template $t$ is selected as the mold protein structure to model the three dimensional structure of the target sequence after. This definition of *best* is determined by the structure-to-sequence alignment between the target protein sequence and the template protein sequence. Let $s$ be the target protein with sequence, $s_{seq}$, and let $t$ be the template protein with known structure, $t_{struct}$. In threading, a target protein with known sequence but unknown structure, $s_{seq}$, is threaded against a target protein with known sequence and structure, $t_{struct}$. The algorithm creates a predicted structural model , $\hat{s}_{struct}$, of the target sequence , $s_{seq}$, based on a template structure, $t_{struct}$, and an alignment of these two proteins $A$. $Model(s_{seq}, t_{struct}, A)$ represents the modelling of the predicted structure; this is the next step in the threading process.

**Local Energy Function Improves Template Selection**

The goal is to improve the selection of an appropriate template protein structure to model the target protein sequence. The energy function attempts to infer information about the alignment from the predicted three dimensional structural model. The weights on the energy function performs dual purposes. First, it weighs the importance of each terms in the alignment with the goal of the final model in mind. Second, the weighted scoring function it represents the distance measures for the template library space.

The ideal energy function will score two proteins from the same family higher than proteins from different families. For example, given the target sequence $s$ and the template library, $T$, containing three protein structures $t_1, t_2, t_3$. We align the target sequence against each of the template structures to produce scores $A(s, t_1), A(s, t_2), A(s, t_3)$ and select the best template $t$ base on the alignment score. If $t_1$ creates a closer resembling protein structure of $s$ than $t_2$, then we expect the score of $A(s, t_1)$ to be higher than the score of $A(s, t_2)$

### 4.2.3 Formulation of Machine Learning Variables

Threading predicts the structure $s_{\hat{struct}}$ of the target protein sequence $s_{seq}$ based on existing protein template with known structures in the template library. The template selection process is refined by adding specific weights in the energy function of the sequence-to-structure alignment. The goal is to infer information about alignment from the model created.

As an aside, note that all the calculations and manipulation will be based on the structural alignment, because it is considered the most realistic super-position of two protein structures. Let $A = Align_{struct}(s_{struct}, t_{struct})$ be the exact accurate alignment, in this case, the structural alignment. Let $x$ be the quality of the alignment and let $y$ be the quality of the prediction process. The algorithm estates $w_i$ using the least squares method.

**Instance**

Let $x$ be the energy function from summing each of the pairwise verticle column of a structural alignment, where $x = \sum_{\forall j} w_j X_j$, choose $j = \{1, 2, 3, 4, 5\}$ such that

$X_1$ = mutation score

$X_2$ = fitness score

$X_3$ = secondary structure score

$X_4$ = pairwise score

$X_5$ = gap penalty More specifically, $X_1, ..., X_5$ are pairwise alignment scores extracted from the structural alignment $A$.

**Label**

Let $y$ be the ability of a template structure to model the unknown target sequence accurately. The process of prediction uses MODELLER and TM-SCORE to obtain the predicted target structure, $\hat{s}_{struct}$. In the first step of the process, $Modeller(s_{seq}, t_{struct}, A)$ predicts the structure of an unknown target sequence $s_{seq}$

using the template structure $t_{struct}$ with alignment $A$. The modelling results in a predicted structure, $s'_{struct}(t_{struct})$, based on the template structure, $t_{struct}$. The quantitative measure of this predicted structure is evaluated by comparing $s_{struct}$ with the real native structure using TM-Score.

$$TMScore(s_{struct}, \hat{s}_{struct}) \tag{4.30}$$

$$where \hat{s}_{struct} = Modeller(s_{seq}, t_{struct}, Align_{struct}(s_{struct}, t_{struct})) \tag{4.31}$$

## 4.3 Implementation

### 4.3.1 Obtaining the Raw Values for $\vec{x}$ Instances

In order to obtain the instances of $\vec{x} = (x_1, x_2, x_3, x_4, x_5)$, first CE Align is used to obtain the true structural alignment file 'protein1.pdbprotein2.pdbal'. Then Shell script cycles through the CE Align files, parses each, and runs the Threading program by calling './thread protein1 protein2 outputfile.nrg'. The Threading program is coded in c++ and is the lab's Threading program, which runs a regular dynamic programming algorithm for one alignment. The code is modified to take in the given CE Align's structural alignment and calculate the energy terms based on the pairwise verticle columns in that structural alignment.

### 4.3.2 Obtaining the Raw Values for $y$ Labels

The labels from training data are derived from the closest truth to sequence-structure alignment. First the algorithm begins from a structure-structure alignment, which is the real truth of how two proteins relate to one another. Next, the algorithm uses MODELLER to model an unknown sequence by giving it these structure alignment and template structure. MODELLER creates a predicted structure of the unknown sequence using its given template structure. Then the method uses TM-SCORE to measure this predicted structure against the original structure as the indicator of how well the prediction performs.

To obtain the $y$ labels, MODELLER first creates a predicted target structure, then TM-SCORE measures the quality of this predicted structure against the native structure. MODELLER then take in three parameters as input: an alignment, a target sequence, and a template structure. The alignment is the true structural alignment generated by CE ALIGN. The target sequence is a FASTA sequence file format, and the template structure is a PDB structure file format containing the three-dimensional coordinates. Because PERL is particularly efficient at text manipulation, PERL script is used to read them and convert files into new formats. PYTHON script is also used to run MODELLER; thus, a PYTHON script is created for each instance of running MODELLER by taking one *.pir file and one structure to model a sequence. A SHELL script is coded to call PERL script to convert CE ALIGN's *.al files into *.pir FASTA alignment file format. At this point, the model for the $y$ label has been created, now it needs to be evaluated. This script

also acquires the required template protein structure, creates PYTHON script to run MODELLER, runs the created PYTHON script for MODELLER, and lastly, runs TM-SCORE on resulting predicted and original native structures. As a special side note, there is a special treatment of no chain '_'; called by another PERL script used to parse out TM-SCORE to be used as Y.

### 4.3.3 Normalization of the Raw Data

The least squares machine learning method is implemented with three different types of normalizations. Normalization is an important pre-processing step of the raw data to ensure that variables falls within similar ranges so that no single variable is excessively represented. Several different types of normalization were implemented; however, only the method displaying the best results is discussed in the Result and Discussion chapters. This is the method that uses statistical normalization for the overall data. The first normalization method seperates all given alignments into groupings by target sequence before normalization. The second normalization method considers all the alignments, without differencing the target sequences. The first method normalizes the data over each target protein, the second method normalizes the data over the entire dataset. Both of these normalization methods uses the statistical equation $\frac{X-M}{StandardDeviation}$. A third normalization method centered the overall data at 0 and within the range of $+1$ and $-1$.

### 4.3.4 Applying Least Squares Method with Weights

Finally, MATLAB is used to code the machine learning least squares method. The $\vec{x}$ and $y$ values were acquired by the steps described above previously. MATLAB firsts acquires these raw data by looping through the resulting flat files and matches $\vec{x}$ and $y$ values by template protein, which is a part of the file name. The internal MATLAB function  is used because it handles matrices which do not have inverses when calculating eigenvalues in the least squares method. The processed input data for the least squares method is $y, x_1, x_2, x_3, x_4, x_5$ and the output is $w_1, w_2, w_3, w_4, w_5$.

### 4.3.5 Evaluation of Results

MATLAB as well as MICROSOFT EXCEL was used to create the graphs and evaluate the results, such as calculating the training and testing errors, determining correlations between variables, plotting graphs, and other evaluation of the results. The protein categorization web tools used are SCOP, CATH, and PFAM.

| Machine Learning Variable | $X$ | $Y$ |
|---|---|---|
| Inputs | $Align_{struct}$ | $Align_{seq-struct}(s_{seq}, t_{struct})$ |
| Process | alignment score function | MODELLER |
| Output | $E = (\sum w_j * E_j)$ | TM-SCORE |

Table 4.2: Machine Learning Variables
Table containing the Values of what the Machine Learning attempting to learn

# Chapter 5

# Results

This current chapter presents the data collected from the specialized weights by least squares method. The next section presents the RMSD error for the training predictions and the testing predictions. The final section looks at the percentage of proteins from each template cluster that demonstrates an improvement in RMSD when compared to the old weights.

## 5.1 New Template-Specific Weights

The dataset contains 23 template proteins with 298 alignments after manipulation and parsing. The results used for evaluation farther eliminates protein template with insufficient input datapoints and null output weights. This is the dataset used for training and testing. Although three different normalization methods were implemented , only the best normalization method is displayed. In this normalization

method each template is standardize over the entire dataset. The newly trained template-specific weights are compared against the globally trained general weights and presented for each specific protein templates.

| ID | W1 | W2 | W3 | W4 | W5 |
|---|---|---|---|---|---|
| 1 | 0 | 0.16447 | 0 | -0.049239 | -0.11892 |
| 2 | 0.815 | 0.23948 | 0 | 0 | 0 |
| 3 | 0.41715 | 0 | 0.022737 | 0.24289 | 0 |
| 4 | -0.8192 | 0.21085 | 0.051193 | 0.66751 | 0.92151 |
| 6 | 1.0601 | 0 | -0.56003 | 0.062195 | 0.17245 |
| 8 | 0 | 0 | 0 | 0.11133 | -0.17548 |
| 9 | 0.49058 | 0 | 0.053196 | 0 | 0 |
| 10 | -0.26355 | 0.0010324 | -0.090327 | 0.42245 | 0.21192 |
| 11 | -0.056975 | 0.24329 | -0.037997 | 0.064889 | -0.071942 |
| 14 | -0.043557 | 0.89888 | 0.087409 | 0.27428 | 0.3785 |
| 17 | 0 | -4.0761 | 2.1435 | -9.2163 | -5.0151 |
| 18 | 0 | 0 | 0 | 0.22478 | 0 |
| 21 | -0.1735 | -0.48968 | -0.026472 | -0.21806 | 0.3688 |
| 22 | 0.35343 | 0.42885 | -0.079691 | 0.54267 | -0.1801 |
| 23 | -0.67744 | -0.32591 | 0.46977 | -1.1406 | -0.19094 |
| all | -0.17486 | 0.34828 | -0.22392 | 0.38325 | 0.0012677 |

Table 5.1: Template-Specific Weights
Using the second normalization method, where the data is normalized over the entire dataset. Each row in the table is the specific weights for the corresponding protein template ID, the last row is the general weight for all datasets.

## 5.2 Improvements of New Template-Specific Weights

Two types of errors are explored: training errors and testing errors. Training error is the RMSD incurred while training the weights using least squares method; the testing error is the RMSD error incurred while testing the trained weights using least squared method to get the predicted label.

First, the specialized weights are evaluated qualitatively by RMSD Error. Re-

member from machine learning the Loss Function which measures the difference between the true label and the predicted label. RMSD error measures the difference between these two labels using RMSD formula, $\sqrt{(\frac{true-predicted}{true})^2}$; The table compares newly trained weights with the old general weights by measuring the difference in the tallied difference in RMSD error for each methodology. A positive difference demonstrates an improvement; a negative value demonstrates the opposite. Overall the RMSD error improved by 12.74% for testing error and 79.96% for training error.

| | New Template-Specific | | Old General | | Difference (new-old) | |
|---|---|---|---|---|---|---|
| ID | Training | Testing | Training | Testing | Training | Testing |
| 1 | 1.67E-16 | 0.19392 | 0.88797 | 0.57303 | 0.88797 | 0.37911 |
| 2 | 1.96E-17 | 0.11724 | 0.35118 | 0.077062 | 0.35118 | -0.040178 |
| 3 | 7.17E-17 | 0.66623 | 0.50554 | 0.43505 | 0.50554 | -0.23118 |
| 4 | 0.65161 | 1.4538 | 1.4649 | 1.5457 | 0.81329 | 0.0919 |
| 6 | 1.80E-16 | 1.0213 | 0.55488 | 0.89275 | 0.55488 | -0.12855 |
| 8 | 5.55E-17 | 0.74584 | 1.1278 | 1.4596 | 1.1278 | 0.71376 |
| 9 | 7.85E-17 | 1.0715 | 0.63693 | 0.47632 | 0.63693 | -0.59518 |
| 10 | 0.27802 | 0.52055 | 0.58765 | 0.7 | 0.30963 | 0.17945 |
| 11 | 0.22227 | 0.42556 | 0.29873 | 0.53563 | 0.07646 | 0.11007 |
| 14 | 0.031323 | 0.63798 | 0.30941 | 0.21185 | 0.278087 | -0.42613 |
| 15 | 0.58274 | 1.2602 | 1.3116 | 0.62527 | 0.72886 | -0.63493 |
| 17 | 9.42E-16 | 1.0168 | 2.299 | 2.6574 | 2.299 | 1.6406 |
| 19 | 2.12E-16 | 0.43561 | 0.39986 | 0.59146 | 0.39986 | 0.15585 |
| 21 | 0.28589 | 0.53438 | 0.52993 | 0.54308 | 0.24404 | 0.0087 |
| 22 | 0.11565 | 0.4409 | 0.36824 | 0.15791 | 0.25259 | -0.28299 |
| 23 | 0.43138 | 0.81294 | 1.3341 | 1.3827 | 0.90272 | 0.56976 |
| average | 0.162430188 | 0.709671875 | 0.8104825 | 0.80405075 | 0.648052313 | 0.094378875 |

Table 5.2: RMSD results
Training and Testing errors measured in RMSD for new specific weights original general weights.

Next, consider the number of proteins that predicts the label better using new specific weights compared to the using the old general weights.

| ID | Training | Testing |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 0 | 100 |
| 3 | 0 | 100 |
| 4 | 14.28571429 | 28.57142857 |
| 6 | 0 | 33.33333333 |
| 8 | 0 | 0 |
| 9 | 0 | 100 |
| 10 | 27.27272727 | 23.80952381 |
| 11 | 53.57142857 | 57.14285714 |
| 14 | 0 | 85.71428571 |
| 15 | 33.33333333 | 91.66666667 |
| 17 | 0 | 33.33333333 |
| 19 | 0 | 66.66666667 |
| 21 | 29.41176471 | 37.5 |
| 22 | 0 | 85.71428571 |
| 23 | 21.42857143 | 61.53846154 |
| AVG | 11.20647122 | 56.56192766 |

Table 5.3: Percentage Improved
Quantitative results showing the percentage of proteins that perform better in each family.

# Chapter 6

# Discussions

This chapter discusses the improvements form the old general weights to the new template-specific weights. The first section compares the new and old weights using different evaluation critereias. The next section compares the differences between the two types of label used to measure the quality of the prediction: the final desired product of target structure and the process toward modelling the target protein. The next section consider the results in the context of protein classification. Common protein classification tools, such as SCOP, CATH, and Pfam, are used. The last section finally offers a critical review of the implementation and methodology in this thesis.

## 6.1 New Weights Compared to Old Weights

There is a positive correlation between the sum of the new weights and the testing error, measured in RMSD. This positive correlation implies that templates with larger changes in its new weights are more likely to have larger testing errors. This suggests that proteins with variable weights are more difficult to map onto the template library space. Template 17 was removed because it performed unusually well; this specific template is further explored later.


Figure 6.1: Total New Weights Versus Testing Error.
Sum the five new weights and compare it to the RMSD testing error.

Now, consider the training error and the testing error between the new weights and old weights. When the slope of the error ration line is one, the RMSD error for the old weights and new weights are the same. For both the training error and the

testing error, the slopes are less than one; this means that the old weights have a larger RMSD errors than the new weights. In other words, the new weights perform better than the old weights. There is a stronger correlation for the testing error than the training error. This is due to the dependence of the training error on the provided training dataset, while the testing error is based on the predictive power.


Figure 6.2: Ratio of Old versus New Error.
Compare the training and testing error of the new weights versus the old weights.

Next, the ratio of training error to testing error shows the predictive power of the old weights compared to the new weights. The new weights has a flatter slope and a weaker correlation, demonstrating that it has stronger predictive ability but with less certainty. While the old weights is not as strong of a predictor, its training

error heavily effects its testing error.

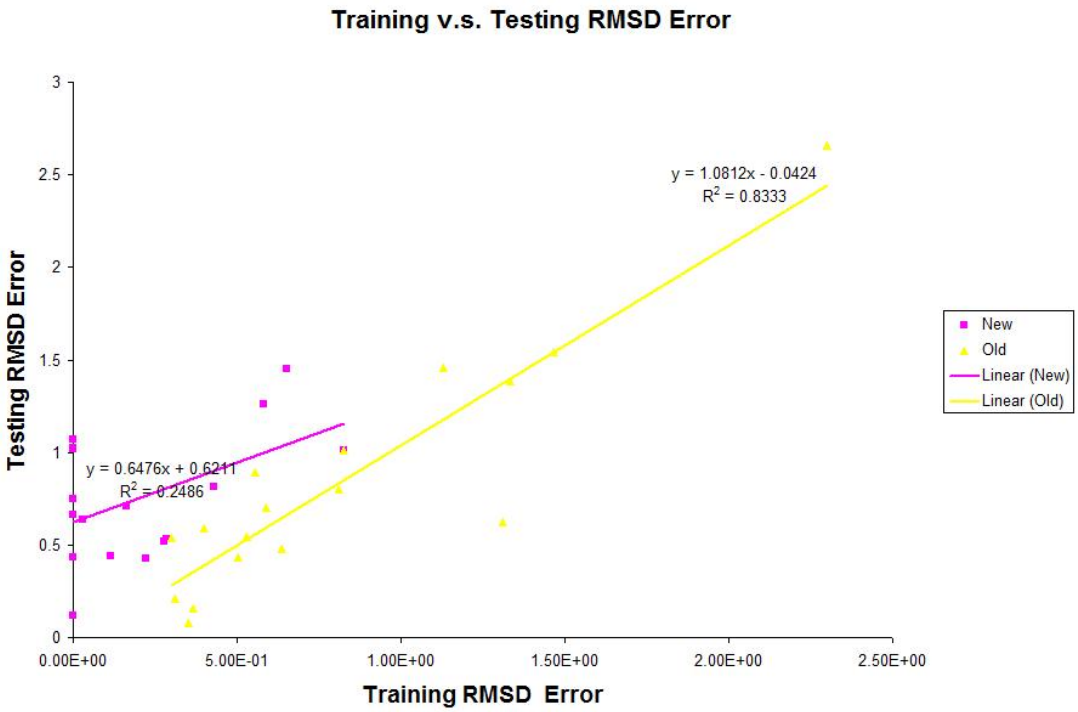


Figure 6.3: Predictive Power of Old Weights Versus the New Weights. Compare training error to the testing error measured in term of RMSD for the old weights as well as the new weights.

Finally, when the overall performance of each template is examined, the training and testing error gives counter-intuitive results. Fist, consider the training error of each of the 23 templates studied: all templates show an improvement in error but only three of the sixteen protein templates show an improvement in the majority of its proteins. These conflicting results in RMSD error and percentage improvement suggest that there are individual proteins which perform much better or much worst which causes anomalies in the final values. Looking at the testing errors by per protein template groups and nine out of sixteen perform better. However, if the percentage improvement is considered, ten instead of nine out of sixteen perform

better.

| | | Error | $> 50\%$ Improved |
|---|---|---|---|
| Training | better | 17 | 3 |
| | worst | 0 | 14 |
| Testing | better | 9 | 10 |
| | worst | 7 | 7 |

Table 6.1: The effect of template-specific weights on template performance.

## 6.2 Methodology and Measures

### 6.2.1 Normalization and Error Measurements

RMSD Error of normalization method 1 and normalization method 2 demonstrates that normalization method 2 has lower training error and testing error. This justifies why the results from the second normalization method are used for presentation and discussion.

To determine the quality of the measurements used to evaluate each template, compare the two measures: the RMSD error, and the percentage of model improvement. While there appears to be a strong one-to-one correlation between the two different measures for training errors, there is little correlation for the testing error. The overall RMSD error has little to do with the overall improvements for all the models, meaning there are individual models which heavily influence the overall RMSD error.

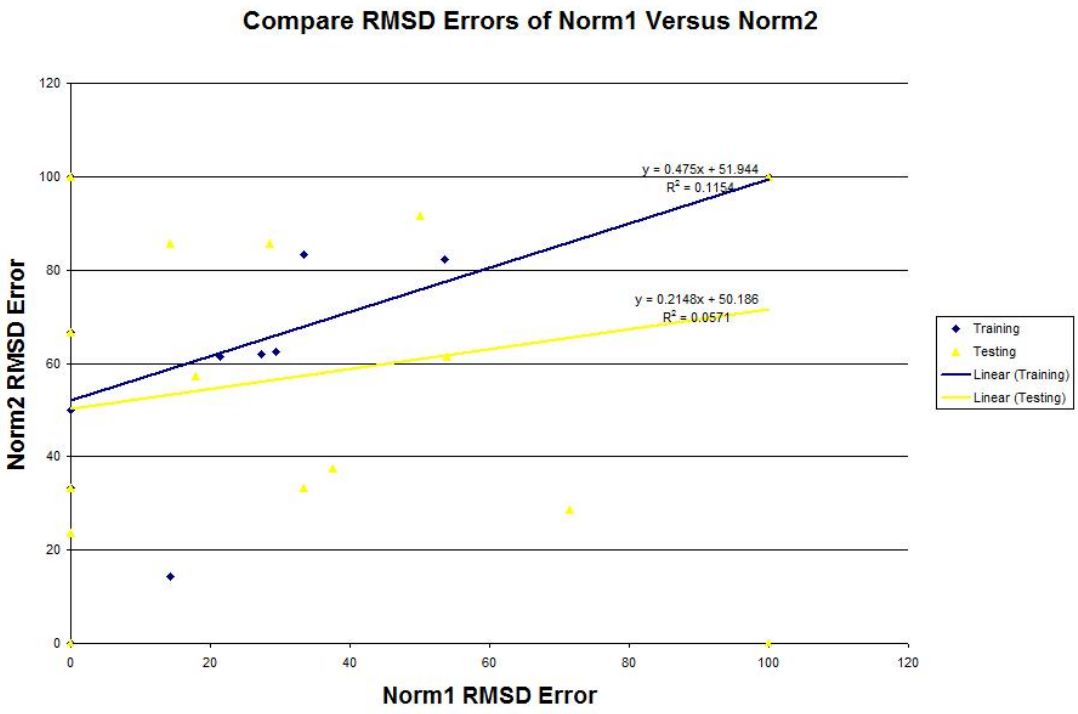


Figure 6.4: Normalization method 1 and Normalization method 2. Compare the error rates of two normalization methods implemented

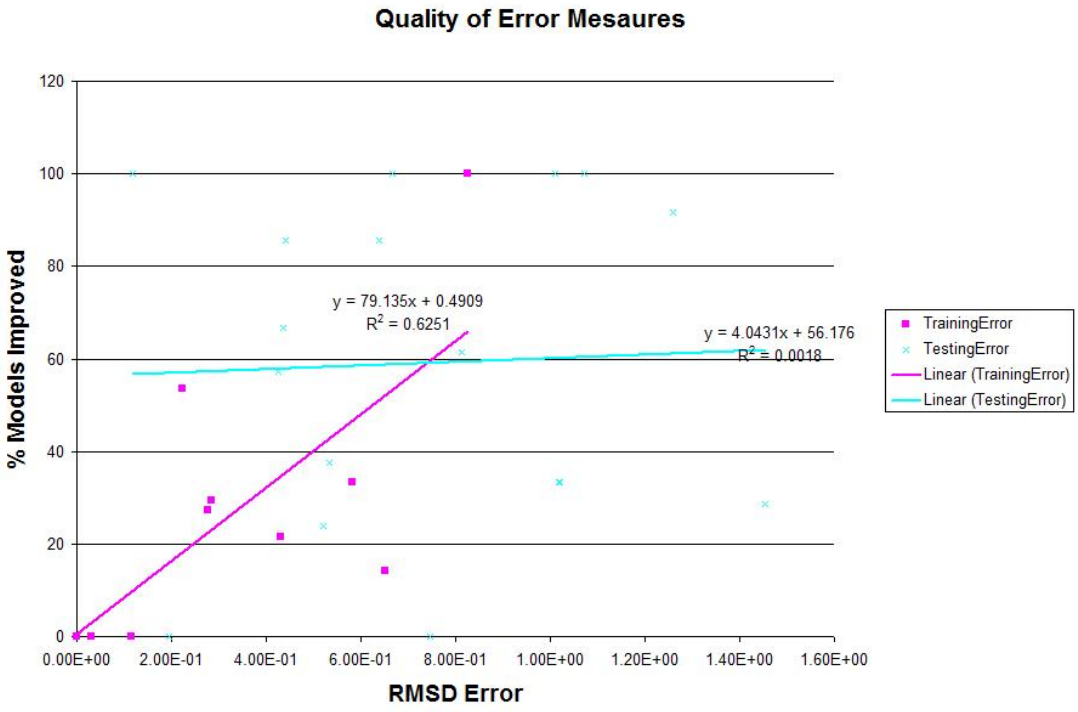


Figure 6.5: Quality of the measures.
Relationship of the two different measures presented.

### 6.2.2 Labels Represent Modelling Process rather than Resulting Structure

In machine learning, the label $\hat{y}$ represents the real desirable output predicted by the instances $\vec{x}$. This thesis considers the label with respect to the second step of threading, the modelling of the predicted target structure. Original measure using structure alignment and z-score comparing the target-to-template similarity is insufficient. The original RAPTOR uses a structure alignment to measure the quality of a template [30]. Unfortunately, the structural alignment is not a precise nor relevant description of the template's ability to model the target. In fact, template selection is followed by the modelling step to model the predicted target structure. In addition, traditional measures using Z-SCORE is insufficient in measuring the sequence-to-structure alignment quality. TM-SCORE measures template quality with size normalization with continuous, rather than discontinuous, data [35]. TM-SCORE of the predicted target structure is a better fit than Z-SCORE of the template structure because the processes of threading is used rather than using only the final product. This new label allows the prediction model to account for the process rather than only the final result.

| ScoreType | Old RMSD Error | New RMSD Error |
|---|---|---|
| Z-SCORE | 0.824805388765534 | 0.622239550607703 |
| TM-SCORE | 0.80405075 | 0.709671875 |

Table 6.2: The performance of Z-SCORE versus TM-SCORE.

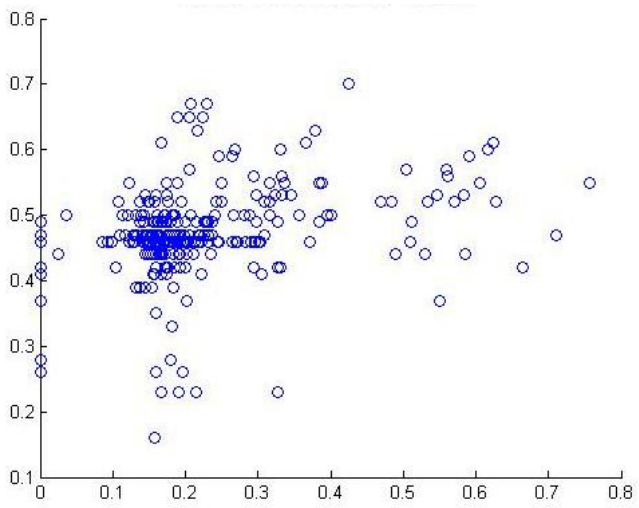


Figure 6.6: RMSD versus TM-Score as Label
Comparing RMSD with TM-Score as the label for the machine learning algorithm.

## 6.3 Protein Family Categorization

### 6.3.1 Case Study

The protein 1flm belongs to the Pfam family of pyridox oxidase. 1flm belongs to only one Pfam architecture and its Pfam family has a large number of seeds. In addition, the old weights of template 1flm has the largest testing error and its new weights have the largest summation. The large testing error for the old weights and largest new weight summation indicates that the protein is drastically different from other proteins in the template library space. However, having one unambiguous architecture and being defined by many seeds within its family allows 1flm to have a better defined space.

Figure 6.7: Case study of 1flm
Structure of 1flm protein belonging to protein family pyridox oxidase.

| | |
|---|---|
| Template ID | 17 |
| Protein Name | 1flma |
| PFAM Architecture Size | 22 |
| Number of PFAM Architectures | 1 |
| Number of PFAM Seeds | 145 |
| Number of CATH Entries | 1 |
| Size of CATH Family | 9 |
| Size of CATH Homologuous Super Family | 2 |
| Size of CATH Toppology | 32 |
| Number of SCOP Entries | 1 |
| Size of SCOP Family | 3 |
| Size of SCOP Super Familiy | 2 |
| Size of SCOP Fold | 165 |
| SCOP FoldName | all-beta |

Table 6.3: Case study of 1flma.

### 6.3.2 Family Analysis

The resulting RMSD error improvement is compared with several protein categorization characteristics extracted from SCOP, CATH, and PFAM. These includes the number of architectures in PFAM, the number of PFAM seeds, the number of CATH family and homologous superfamily, and the number of SCOP family and superfamily. When looking at these protein templates, the RMSD performance improves with respect to the number of seeds in PFAM. The number of seeds also correlates with the new template-specific weights. PFAM families with multiple seeds are more difficult to be defined by a generalized weights, this may be the reason why template-specific weights perform better for protein families with multiple PFAM seeds.

Upon examining all the protein templates which stem from multiple architectures, all except one perform better in general. PFAM uses hidden Markov model defined by seeds to form its family categorization. Protein templates which belongs to PFAM families that have multiple architectures correlates to larger CATH family and SCOP superfamily. This same relationship is not seen in protein templates with only one PFAM architecture. This implies that the architectures in PFAM correlates to the top level protein family categorization such as family and superfamily.

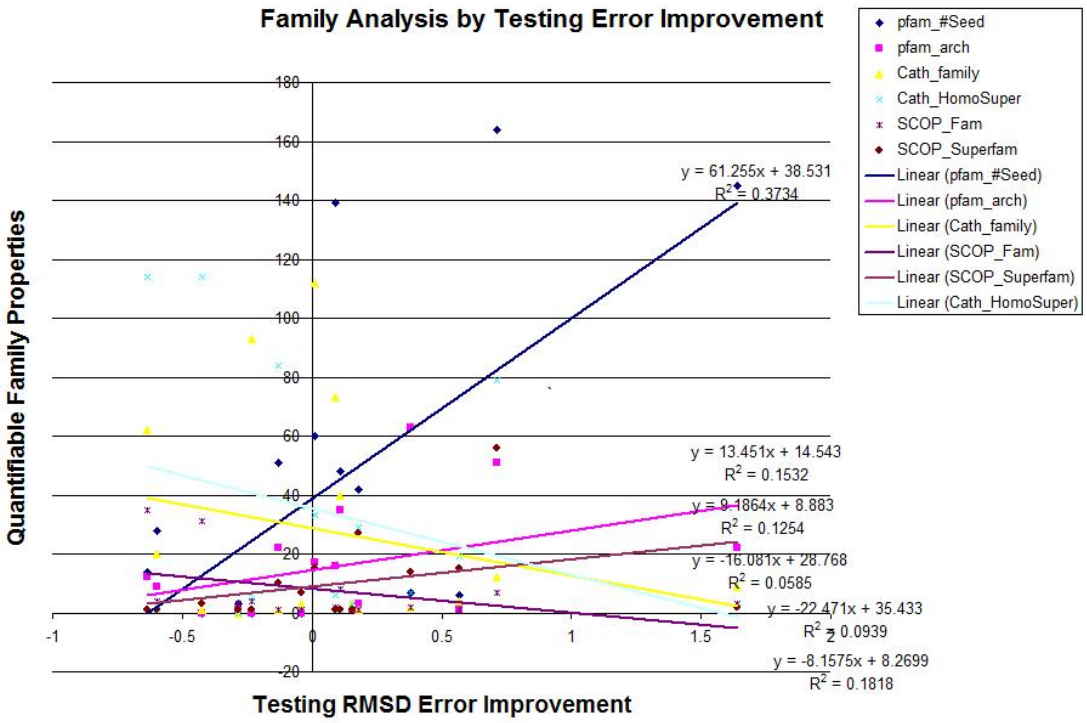


Figure 6.8: Family Categorization Analysis
Scatter plot of the RMSD error against quantifiable family properties for each protein template.

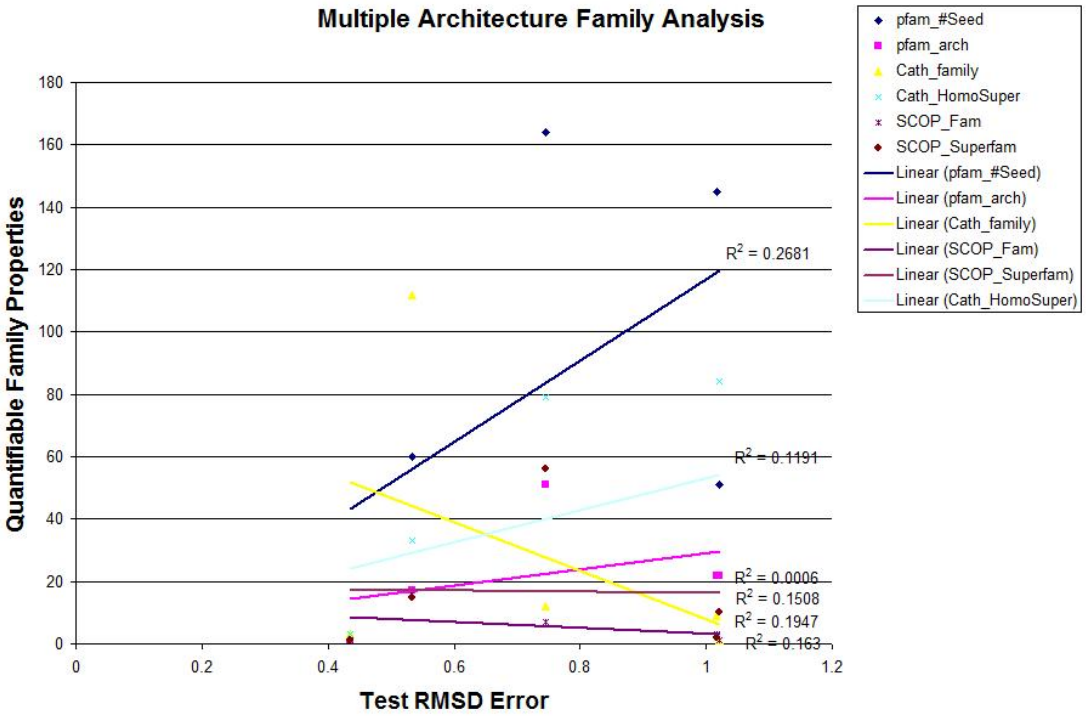


Figure 6.9: Multiple Architectures.
Considering only the protein families with multiple PFAM architectures.

## 6.4 Criticisms

The template library is much smaller of non-redundant set of proteins rather than a set containing all known proteins. Due to the large number of repeated domains and motifs there is a lot of redundancy in the protein database. The experimental set from this thesis is also smaller than the actual template library. The assumptions is that the weights from this set of proteins is accurate for overall protein set. The metric only selects one template and trains template-specific weights as its representation within the entire library of templates. Although a larger and more accurate dataset can be acquired, the datasets used in this thesis sufficiently demonstrate that template-specific weights do outperform the global general weights.

# Chapter 7

# Conclusion

## 7.1 Summary of Methodology and Findings

This section briefly recaptures the major content of the chapters from this thesis. This thesis examines template-specific weights for the energy terms of structure-to-sequence alignment used in the template selection step of the threading process in comparative modelling. First, the necessary biology background to understand this thesis is given. Next, the survey chapter narrows the problem at hand and examine the leading research in the area. Then, in the methodology chapter, after a brief problem definition, a machine learning formulation for training weighted energy functions is defined. The methodology chapter further describes the machine learning instances as energy terms from the sequence-to-template alignment, this measures the first step of threading, template selection. The machine learning provide represent a quantifiable base to the process of modelling, which represents

the second step of threading, the step where the structure of the target sequence is modelled. The implementation chapter describe acquisition of raw data, normalizing the data, as well as the implementation of least squares method. The results chapter shows how the new template-specific weights are obtained as well as presents their corresponding RMSD error and the percentage of all proteins with improvement.

Overall, the new template-specific weights show an average 12.74% improvement in RMSD error and average 56.56% of proteins perform better. Of the templates, 14 out of the of the 17 have a better RMSD and 10 out of 17 show a overall improvement. Although using Z-Score as labels show better improvements than using TM-Score as labels, a plot of the two score shows a non-linear relationship between the two. The label used in the machine learning methodes are with respect to the methodology rather than to the desirable final product. It is a refinement over the process, thus is more realistic to the prediction method. Finally, looking at various properties with consideration of family classifications, we observe a correlation between the improvement of the new specialized weights perform and the number of pFam seeds in the protein template.

## 7.2 Contribution

Central to every protein structure prediction method is a computational metric which attempts to bring out the biological information.

The first contribution of this is thesis is applying template-specific weights for an energy function for sequence-to-template alignment improves template selection in threading. Existing works in ab initio prediction uses energy minimization to optimize the energy function for selecting a decoy selection from a decoy set [34, 35, 17, 14, 8]. This technique is transferred to threading's sequence-to-structure alignment scoring function for template selection.

The second contribution of this thesis is to recognize that using RMSD with a threshold in traditional threading template libraries, such as RAPTOR is not advisable [30]. Instead it suggests that TM-Score for the modelling process should be used as the qualifying label for the machine learning formulation. In this way, specific components of template selection in machine learning illuminates the process model regardless of the method or result being used quantifying the results.

The third contribution of this thesis is the proposal of family analysis for sequence-to-structure analysis for the modelling process in threading. Existing work analyzes sequence alignment against structure alignment [12].

It has been established in the thesis that sepecific weights trained locally to each protein template performs better than general weights trained globally by all templates. Evidence from the results shows that the set of new template-specific weights improves the predictive ability of the energy function in sequence-to-structure alignment of threading. In examining the categorization of protein families, templates with more seeds in pFam shows a greater RMSD error improvement. As is evident in the number of pFam seeds and two other family properties. This finding achieves

the objective of creating template library with a metric that measures the space to reflect some family knowledge.

## 7.3 Future Work

There are many possible next steps to improve the accuracies of the specific weights. First the dataset needs to be expanded to include more initial raw data. Secondly in the methods section, alternative normalization and machine learning methods can be applied to train the function by adjusting its weights. Other types of alignments can be implemented as the true alignment for the energy function and other labels to quantify the threading process can be attempted. It will be interesting to examine family properties with respect to the quality of these weighted scores in a systematic manner. We can expand the dataset to include protein families information from PFAM and examine the connection from phylogeny or evolutionary history to three-dimensional modelling.

# Appendix A

# List of Computer Tools Used

This section lists the software programs used, as well as go into some of them in detail, and then lists the computer hardware used.

## A.1 Software List

- MATLAB Version 7.5.0.342 R2007b
- CE ALIGN [27]
- THREADING by C++ gcc version 4.1.2 (Ubuntu 4.1.2-0ubuntu4)
- CVS Concurrent Versions System (CVS) 1.12.13 (client/server)
- Bash SHELL Script: GNU bash, version 3.2.13(1)-release (i486-pc-linux-gnu)
- PERL, v5.8.8 built for i486-linux-gnu-thread-multi

- Python 2.5.1 (r251:54863, Mar 7 2008, 03:41:45) [GCC 4.1.2 (Ubuntu 4.1.2-0ubuntu4)] on linux2
- Modeller [19, 24, 3]
- TM-Score [35]
- Scop [4]
- Cath v3.2.0 [7]
- pFam [11]

### A.1.1 CE-align

CE Align is an structure alignment software program which breaks both proteins in to fragments, and then align by rules these fragments into pairs. Finally CE Align extends the alignment combinatorially by first starting the aligned fragment pairs and secondly by finding the longest/best alignment. CE program's z-score measures the quality of the alignment; a z-score value above 3.5 means there is less than 0.001 chance. RMSD is short for root mean square deviation of the position of the alpha-carbons in the protein chain. [12]

### A.1.2 Modeller

Modeller is a common modelling software program that builds full length protein structure models. It extracts those restraints from the template and then optimizes

the spatial restraints. MODELLER was first used because it is the basis of many other modelling tools and is popular amongst biologists. When using MODELLER to model structures of $3.5 - 35.7\mathring{A}$ and with a standard deviation of $4.8\mathring{A}$, there are no correlations between the maxSub score and the resulting model created by MODELLER.

### A.1.3 PFAM

PFAM is a collection of common protein families built from multiple sequence alignments and profile hidden Markov models [29]. There are two methods of building PFAM, either by PFAM-A, using human crafted multiple alignments or by PFAM-B, using automatic clustering of the rest of SWISS-PROT using the program Domainer [23]. PFAM-A uses high quality seed alignments to build the HMMs and to which additional sequences are added to generate a final alignment. The seed for the alignment is honed by iterative methods [23].

Pairwise sequence alignments are able to detect structural conservations from evolutionary function. The strength of profile HMM of PFAM is that it is able to detect weakly related proteins and multidomain proteins [23]. The weakness of PFAM is the linear HMM model, which can capture a limited order amino acids correlations [23]. Thus long distance relationship of amino acids that are far apart in the linear sequence, but close in three-dimensional proximity cannot be modelled by an HMM [23]. In addition HMM assume amino acid in the sequence is independent of the probabilities of its neighbours, which is not true such as when hydrophobicity

is taken into account [23]

### A.1.4 SCOP

SCOP classification of protein structure is used; the top level is class, followed by fold and followed by super family and then family. Classes are divided into all-alpha, all-beta, alpha or beta, alpha and beta domains. Super family represents the evolutionary distinct lineage, for example convergent evolution where there is unrelated functions but same structure. Orthologuous protein are proteins same function but in different organisms; paraloguous are different proteins of related descent from duplicate common ancestors but with different functions. Finally family is good for differentiating these paraloguous proteins. [12]

## A.2 Hardware List

- dna server - Linux dna 2.6.20-15-server 2 SMP Sun Apr 15 07:41:34 UTC 2007 i686 GNU/Linux ; i486-linux-gnu
- personal computer - windows server (Microsoft Windows XP Professional Version 2002, Intel(R) Pentium(R) 4 CPU 3.00GHz 2.99GHz, 1.00GB of RAM)

# Appendix B

# Protein Names with Family Analysis

| | | pFam | | | CATH | | | | SCOP | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | Name | ArchSize | NumArch | NumSeed | Entires | Family | HomoSupfam | Topo | Entries | Fam | SuperFam | Fold | FoldName |
| 1 | 1a12a | 63 | 1 | 7 | 1 | 2 | 7 | 1 | 1 | 2 | 14 | 165 | all-b |
| 2 | 1a34a | 0 | 0 | 0 | 1 | 3 | 36 | 34 | 1 | 1 | 7 | 165 | all-b |
| 3 | 1arb | 0 | 1 | 0 | 1 | 93 | 5 | 31 | 1 | 4 | 1 | 165 | all-b |
| 4 | 1atg | 16 | 1 | 139 | 1 | 73 | 6 | 100 | 1 | 2 | 1 | 141 | a/b |
| 6 | 1ayoa | 22 | 1 | 51 | 1 | 1 | 84 | 34 | 1 | 1 | 10 | 165 | all-b |
| 8 | 1f0xa | 51 | 2 | 164 | 4 | 12 | 79 | 644 | 2 | 7 | 56 | 668 | a+b |
| 9 | 1f1ea | 9 | 1 | 28 | 1 | 20 | 1 | 216 | 1 | 4 | 1 | 258 | all-a |
| 10 | 1f1ma | 3 | 1 | 42 | 1 | 1 | 29 | 83 | 1 | 1 | 27 | 258 | all-a |
| 11 | 1fg7a | 35 | 1 | 48 | 1 | 40 | 1 | 100 | 1 | 8 | 1 | 141 | a/b |
| 14 | 1fiua | 0 | 1 | 0 | 1 | 1 | 114 | 100 | 1 | 31 | 3 | 141 | a/b |
| 15 | 1fj2a | 12 | 1 | 14 | 1 | 62 | 114 | 100 | 1 | 35 | 1 | 141 | a/b |
| 17 | 1flma | 22 | 1 | 145 | 1 | 9 | 2 | 32 | 1 | 3 | 2 | 165 | all-b |
| 19 | 1fn9a | 1 | 1 | 2 | 2 | 3 | 3 | 290 | 1 | 1 | 1 | 334 | a+b |
| 21 | 1fp2a | 17 | 2 | 60 | 1 | 112 | 33 | 216 | 2 | 16 | 15 | 399 | all-a |
| 22 | 1fs7a | 1 | 1 | 3 | 1 | 0 | 0 | 0 | 1 | 3 | 1 | 258 | all-a |
| 23 | 1h7ca | 1 | 1 | 6 | 1 | 4 | 19 | 83 | 1 | 1 | 15 | 258 | all-a |

Table B.1: Family Analysis.
List of proteins with names and their family analysis.

# References


[1] Machine learning. Web, August 2008. 32

[2] Proteomics. Web, August 2008. 2

[3] A. Sali A. Fiser, R.K. Do. Modeling of loops in protein structures. *Protein Science*, 9:1753–1773, 2000. 69

[4] T. Hubbard C.Chothia A. G. Murzin, S. E. Brenner. Scop: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995. 69

[5] Philip E. Bourne and Helge Weissig. *Structural Bioinformatics*. Wiley-Liss, 2003. 20, 21, 22

[6] Julian Lewis Martin Raff Keith Roberts Bruce Alberts, Alexander Johnson and Peter Walter. *Molecular Biology of the Cell*. Garland, 4 edition, 2002. 2, 4, 9

[7] D. T. Jones M. B. Swindells C. A. Orengo, A. D. Michie and J. M. Thornton. Cath: A hierarchic classification of protein domain structures. *Structure*, 5:1093–1108, 1997. 69

[8] Kira M. S. Misura Carol A. Rohl, Charlie E. M. Strauss and David Baker. Protein structure prediction using rosetta. *Methods in Enzymology*, 383:66–93, 2004. 25, 66

[9] M.J. Rooman C.M. Lemer and S.J. Wodak. Protein structure prediction by threading methods: Evaluation of current techniques. *Proteins*, 23(3):337–355, 1995. 21

[10] S. R. Eddy. How do rna folding algorithms work? *Nature Biotechnology*, 22(11):1457–1458, 2004.

[11] S.R. Eddy E.L.L. Sonnhammer and R. Durbin. Pfam: A comprehensive database of protein families based on seed alignments. *Proteins*, 28:405–420, 1997. 69

[12] Jonathan W. Arthur J. Michael Sauder and Roland L. Dunbrack Jr. Large-scale comparison of protein sequence alignment algorithms with structure alignment. *Proteins: Structure, Function, and Bioinformatics*, 40(1):6 – 22, 2000. 18, 19, 30, 66, 69, 71

[13] Piotr Lukasiak Jacek Blazewicz and Maciej Milostan. Some operations research methods for analyzing protein sequences and structures. *4OR: A Quarterly Journal of Operations Research*, 4(2):91–123, 2006. 18, 20, 21

[14] Alexandre V. Morozov Brian Kuhlman Carol A. Rohl David Baker Jerry Tsai, Richard Bonneau. An improved protein decoy set for testing energy functions for protein structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 53(1):76 – 87, 2003. 26, 66

[15] V. Olman K. Ellrott, J. T. Guo and Y. Xu. Improvement in protein sequence-structure alignment using insertion/deletion frequency arrays. *Computer Systems Bioinformatics Conference*, 6:335–42, 2007. 22

[16] Evelyn Fox Keller. *The Century of the Gene*. President and Fellows of Harvard College, 2000. 1

[17] Enoch Huang Kim T. Simons, Charles Kooperberg and David Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of Molecular Biology*, 268:209–225, 1997. 26, 27, 66

[18] Adam Godzik Krzysztof Ginalski, Nick V. Grishin and Leszek Rychlewski. Practical lessons from protein structure prediction. *Nucleic Acids Research*, 33(6):18741891, 2005. 4, 19, 22, 30

[19] A. Fiser R.Sanchez F. Melo M.A. Marti-Renom, A. Stuart and A. Sali. Comparative protein structure modeling of genes and genomes. *Annual Review of Biophysics & Biomolecular Structure*, 29:291–325, 2000. 69

[20] Joel S. Rozowsky Deyou Zheng Jiang Du Jan O. Korbel Olof Emanuelsson Zhengdong D. Zhang Sherman Weissman Mark B. Gerstein, Can Bruce and

Michael Snyder. What is a gene, post-encode? history and updated definition. *Genome Research*, 17:669–681, 2007. 1

[21] Jonathan King Bonnie Berger Matthew Menke, Eben Scanlon and Lenore Cowen. Wrap-and-pack: A new paradigm for beta structural motif recognition with application to recognizing beta trefoils. *Jounal of Computational Biology*, 12:777–795, 2005.

[22] Britt Park and Michael Levitt. Energy functions that discriminate x-ray and near-native folds from well-constructed decoys. *Journal of Molecular Biology*, 258(2):367–392, 1996. 26

[23] N. Provart. Motif and profile analysis. Bio472 lecture of, 14 Mar 2007 2007. 70, 71

[24] A. Sali and T.L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, 234:779–815, 1993. 69

[25] Tamar Schlick. *Molecular Modeling and Simulation*. Springer-Verlag New York, Inc., 2002.

[26] A.A. Schaffer et al S.F. Altschul, T.L. Madden. Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997. 1

[27] I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein Engineering*, 9:739–747, 1998. 68

[28] Jeffrey Skolnick. In quest of an empirical potential for protein structure prediction. *Current Opinion in Structural Biology*, 16:166171, 2006. 24, 25

[29] Terry Speed. Motifs, profiles and hidden markov models. Presentation, Melbourne Bioinformatics Course, September 2003. 70

[30] Jinbo Xu. *Protein Structure Prediction by Linear Programming.* PhD thesis, University of Waterloo, August 2003. 57, 66

[31] Jinbo Xu and Ming Li. Assessment of raptors linear programming approach in cafasp3. *Proteins: Structure, Function, and Genetics*, 53(S6):579–584, October 2003. 28, 29

[32] Jinbo Xu, Ming Li, Guohui Lin, Dongsup Kim, and Ying Xu. Protein threading by linear programming. pages 264–275, January 3–7 2003. 29

[33] Peter Weigele Yan Liu, Jaime Carbonell and Vanathi Gopalakrishnana. Protein fold recognition using segmentation conditional random fields (scrfs). *Journal of Computational Biology*, 13:394406, November 2006.

[34] Andrzej Kolinski Yang Zhang and Jeffrey Skolnick. Touchstone ii: A new approach to ab initio protein structure prediction. *Biophysical Journal*, 85:1145–1164, 2003. 26, 66

[35] Jeffrey Skolnick Yang Zhang. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57 (4):702 – 710, 2004. 27, 28, 57, 66, 69

[36] Michael Zuker. Calculating nucleic acid secondary structure. *Current Opinion in Structural Biology*, 10(3):303310, 2000.