

ON MODELING PROTEIN SUPERFAMILIES
WITH LOW PRIMARY SEQUENCE CONSERVATION

by

Haifeng Ji

A THESIS

Presented to the Faculty of
The Graduate College at the University of Nebraska
In Partial Fulfillment of Requirements
For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Stephen D. Scott

Lincoln, Nebraska

August, 2002

ON MODELING PROTEIN SUPERFAMILIES
WITH LOW PRIMARY SEQUENCE CONSERVATION

Haifeng Ji, M.S.

University of Nebraska, 2002

Advisor: Stephen D. Scott

Understanding the mechanism of cellular redox regulation will provide huge medical benefits for human beings. Since the majority of redox proteins are thioredoxin (Trx) fold proteins, it is important to identify Trx-fold proteins. There is very low conservation in primary structure within Trx-fold protein superfamily. Thus, protein amino acid sequence alone is not enough to lead to the discovery new Trx-fold proteins. In this study, we instead model structural properties common to all Trx-fold proteins in order to discover new Trx-fold proteins. These structural properties include secondary structure patterns, as well as remapped sequences with a reduced alphabet that captures structural properties. We built hidden Markov models on these new sequences, tested on the thioredoxin-fold (Trx-fold) protein family, and efficiently identified Trx-fold proteins. In our experiments, our new model positively identified 78% of distinct Trx-fold protein families and discovered 6 new Trx-fold proteins in the *Campylobacter jejuni* database.

ACKNOWLEDGEMENTS

I am deeply appreciative to my advisor, Dr. Stephen Scott, for his advice, encouragement and support for this project.

I also sincerely appreciate Dr. Vadim Gladyshev for his help, support and great vision for this research. I would also like to thank Dr. Etsuko Moriyama and Dr. Jitender Deogun for their help in various ways.

My appreciation is also extended to Dr. Dmitri Fomenko, Peggy Wen and Gregory Kryukov for their inputs on this project. I also want to acknowledge the support from two grants: "Identity of selenocysteine and terminator UGA codons" from National Institute of General Medical Sciences (NIGMS), NIH and "Center for Bioinformatics Research" from EPSCoR Infrastructure Grant EPS-0091900, National Science Foundation.

Thanks to my family, for their love and care over the years to help me reach this point. I also want to thank my friends for their encouragement and wholehearted support.

Table of Contents

Chapter 1	Introduction.....	1
1.1	Motivation and Purpose of This Research.....	1
1.2	How This Thesis is Organized.....	3
Chapter 2	Background and Review.....	4
2.1	Background on Biology.....	4
2.2	Trx-fold Protein Superfamily.....	4
2.3	Machine Learning Approaches.....	5
2.4	Hidden Markov Models.....	7
2.5	Protein Secondary Structure Prediction.....	11
Chapter 3	Data Analysis Approaches.....	12
3.1	Using Large Data Sets to Train and Test HMMER.....	12
3.2	Hidden Markov Models on Secondary Structure Patterns.....	12
3.3	Hidden Markov Models on Remapped Sequences.....	16
3.4	Database Search.....	19
Chapter 4	Experimental Results and Discussion.....	20
4.1	Results of HMMER on Large Data Sets.....	20
4.2	Jack-knife Test Results.....	22
4.3	Results of Database Search.....	25
Chapter 5	Conclusions and Future Development.....	28
References	30
Appendix A	Examples of Protein Primary And Secondary Structure....	34
Appendix B	Positive Data Set Used for Jack-knife Test.....	35

Chapter 1 Introduction

1.1 Motivation and Purpose of This Research

The collaboration between computer scientists and biologists has created a new field called computational molecular biology. It is also called bioinformatics. Computational molecular biology means using computer science tools and techniques to solve problems in molecular biology. A major task in computational molecular biology is to “decipher” information contained in biological sequences. With the advent of new computer technologies, scientists are able to solve complex problems that cannot be solved without computers [1].

Basic cellular processes are regulated by *oxidation* and *reduction* [2-4]. Oxidation is a process where electrons are removed from a molecule or atom, while reduction is a process where electrons are added to a molecule or atom. In oxidation and reduction reactions, one chemical is oxidized, and its electrons are passed to another chemical, which is then reduced. Such coupled reactions are referred to as *redox reactions*. The regulation of redox reactions is based on oxidant and antioxidant proteins (*redox proteins*). The redox proteins are involved in many basic cellular processes, such as DNA synthesis, apoptosis, signal transduction and transcription [5,6]. Therefore, understanding the mechanism of cellular redox regulation will provide huge medical benefits for human beings, for example, in medical research in cancer, neurological and cardiovascular diseases, and reproduction problems. To understand the mechanism of cellular redox regulation, the first step is to identify redox proteins and to know the specific functions of these proteins [6,7]. Since some of the redox proteins are thiol

disulfide oxidoreductase, and most of these proteins are thioredoxin (Trx) fold proteins, it is important to identify Trx-fold proteins.

Primary structures¹ are generally conserved² within protein families. Therefore, conventional approaches such as hidden Markov model (HMM), work well. However, interfamily similarity within Trx-fold superfamily is low. Thus, sequence analysis tools such as HMMER cannot easily distinguish primary structure conservation from noise and identify new Trx-fold protein families. For example, in Figure 1.1, there are segments of five Trx-fold proteins aligned. We can see that only the two cysteines (Cs) are conserved in the alignment.



```

1A8L:  KLIVFVRKDHCQYCDQLKQLVQEL
1BED:  PVVSEFFSFYCPHCNTFEP IIAQL
1QK8:A  LVFFYFSASWCPPCRGFTPQLIEF
1F9M:A  PVVLDMFTQWCGPCKAMAPKYEKL
1MEK:  YLLVEFYAPWCGHCKALAPEYAKA

```

Figure 1.1 Alignment of segments of five Trx-fold proteins.

In a more rigorous test, we used HMMER to attempt to identify distinct Trx-fold protein families based on primary structure alone (Section 3.1). HMMER is a software that implements profile HMM for protein sequence analysis. In this test, 0% of distinct Trx-fold protein families were identified, indicating that primary structure alone is insufficient to find new Trx-fold protein families. Therefore, in this study, we use structural properties to identify new Trx-fold protein families. These structural properties

¹ A protein's linear sequence of amino acids is called its primary structure.

² Conservation means the preservation through time of some amino acids in the sequence of an evolving protein.)

include secondary structure patterns, as well as remapped sequences with a reduced alphabet that captures structural properties. We built hidden Markov models on these new sequences, tested on the Trx-fold protein family, and efficiently identified Trx-fold proteins that contain low primary structure conservation. We discovered that the profile HMM algorithm built on secondary structure patterns positively identified 78% of distinct Trx-fold protein families. We also identified 6 new Trx-fold proteins in the *Campylobacter jejuni* database.

1.2 How This Thesis is Organized

This thesis contains five chapters. Chapter 2 will introduce some background knowledge on biology, machine learning, hidden Markov model, and protein secondary structure prediction. In Chapter 3, we will describe how we do the experiments. In Chapter 4, we will describe the experimental results and give some discussion. Finally, in Chapter 5, we will conclude this thesis and present some suggestions on future work.

Chapter 2 Background and Review

In this chapter, we first introduce some background knowledge on biology. Then we talk about the thioredoxin-fold protein superfamily. After that, we will discuss why we use machine learning approaches to solve the problem. Then, we will introduce hidden Markov models. Finally, we will introduce the concept of protein secondary structure prediction.

2.1 Background on Biology

A protein is composed of a chain of amino acids. There are 20 amino acids commonly found in proteins. Each of them is represented by a one-letter code. A protein's linear sequence of amino acids is called its *primary structure*. The regular folding of a protein in repeated patterns is called its *secondary structure*. These patterns include the α -*helices*, the β -*sheets*, and structures that are neither helices nor sheets, called *loops*. An α -helix is a helix that makes a complete turn every 3.6 amino acids. A β -sheet consists of pairs of chains lying side-by-side [1]. Examples of primary structure and secondary structure can be found in Appendix A.

2.2 Trx-fold Protein Superfamily

The majority of known disulfide oxidoreductases belong to a superfamily which has a thioredoxin *fold* (secondary structures clasping together). Most Trx-fold proteins contain a conserved CxxC (two cysteines separated by two other amino acids) *motif* (a

recurring pattern of protein structure) in the primary structure. This superfamily includes reductants such as thioredoxins and glutaredoxins and oxidants. Primary structures are generally conserved within protein families, but interfamily similarity is low. Thus, sequence analyses are not sufficient to identify new families of Trx-fold proteins. However, there are some common features in Trx-fold proteins. First, the majority of Trx-fold proteins contain the CxxC motif mentioned above. Second, three α -helices and four β -sheets are organized in a specific secondary structure pattern in Trx-fold. Specifically, it is a β - α - β - α - β - β - α motif. The CxxC motif is located between the first β -strand and the first α -helix in the fold. So the entire motif is β -CxxC- α - β - α - β - β - α [8,9]. Therefore, even though the protein primary structures are not conserved, we may be able to use protein secondary structures as well as the CxxC motif to discriminate Trx-fold proteins.

2.3 Machine Learning Approaches

Machine learning techniques are very suitable for processing large amounts of data and characterizing noisy patterns. Machine learning is a model of computation that lets computers learn from experience. The goal of machine learning is to learn a *classifier* (function) that correctly labels the data. To achieve this goal, we first need a *training set* in which the examples are already labeled. The computers can use the training set to learn a *hypothesis* that is a good approximation to the *target concept* (the function that labels the training data). We can then use the hypothesis to classify new (unlabeled) examples. The algorithm learns by altering its hypothesis. Examples of

learning algorithms include artificial neural networks (ANNs), evolutionary algorithms, clustering algorithms, hidden Markov models and decision trees [10]. C4.5 is a decision tree learner.

Artificial neural networks (ANNs) contain many neuron-like switching units, with many weighted interconnections. Normally, it is used when input is high-dimensional, the form of the target function is unknown, and long training times are acceptable. Some applications of ANNs are speech recognition, image classification, and protein structure prediction. The basic unit of an ANN is related to the *perceptron* [11]. A perceptron takes a vector of real-valued inputs, calculates a linear combination of these inputs, then outputs a 1 if the result is greater than some threshold and -1 otherwise. The perceptron learns by adjusting its weights (w): $w_i(new) = w_i(old) + n_{(t-o)}x_i$. Here t is the target output for the current training example, o is the output generated by the perceptron, and n is a positive constant called the learning rate. We can view the perceptron as representing a hyperplane decision surface in the n -dimensional space of points. The perceptron outputs a 1 for points lying on one side of the hyperplane and outputs -1 for points lying on the other side. Perceptrons can represent many functions, but they cannot represent functions that are not linearly separable. To represent those functions, we add hidden layers of perceptrons to re-map the linearly non-separable input space to linearly separable space.

Another study by Muggleton et al. [12] used grammatical representations for predicting members of biological sequence families. They used the Inductive Logic Programming (ILP) Bayesian approach to learning from positive sequences to generate a grammar for recognizing the human neuropeptide precursors (NPPs) protein family. They derived a group of features of NPP from ILP, and derived other groups of features

from other learning strategies. They built recognition models for the amalgams of these groups using C4.5 and measured its performance using both predictive accuracy and a new cost function, Relative Advantage (RA). A model that includes grammar-derived features achieved the highest RA. This study shows that grammatical representations are also useful for learning from biological sequence data.

Some other related work builds HMMs on non-primary structure. Luthy et al. [13] used mutation table for residues in α -helices, β -strands or other secondary structures to calculate the profile. Using this profile to scan the sequence database, they detected avidin. Another method, presented by Bowie et al. [14], used 3D profile method to find sequences that are most compatible with the environments of the residues in the 3D structure. This method detected the structural similarity of the actins and 70-kilodalton heat short proteins.

2.4 Hidden Markov Models

Hidden Markov models (HMMs) are a general statistical modeling technique for “linear” problems like sequences or time series [15]. HMMs were first used in speech recognition [16]. Later HMMs have been used in computational sequence analysis [17], including protein structure modeling [18].

A *Markov chain* is a stochastic process containing a countable or finite state space with transition probabilities associated with the states.

The probability parameters are called the *transition probabilities*, which we will write a_{st} :

$$a_{st} = P(x_i = t | x_{i-1} = s).$$

The key property of a Markov chain is that the probability of each symbol x_i depends only on the value of the preceding symbol x_{i-1} , not on the entire previous sequence. This is because of the one-to-one correspondence between symbols and states. Therefore we have:

$$P(x_i | x_{i-1}, \dots, x_1) = P(x_i | x_{i-1}).$$

So probability of sequence (x_1, \dots, x_L) is:

$$P(x) = P(x_1, \dots, x_L) = P(x_1) \prod_{i=2}^L p(x_i | x_{i-1}).$$

A Hidden Markov model, another stochastic process, is an extension of Markov Chains. While the representation of the *state*, called the *symbol*, is the state itself in a Markov chain, in an HMM the state sequence is hidden and we do not know which state a given symbol belongs to. We use HMMs in modeling Markovian processes whose inherent states have to be identified when the respective symbols of the states are known exactly.

Let's consider an example adapted from Durbin et al. [19], the occasionally dishonest casino. Assume that a casino is typically fair, but with probability 0.05 it switches to a loaded die, and switches back again with probability 0.1 (Figure 2.1).

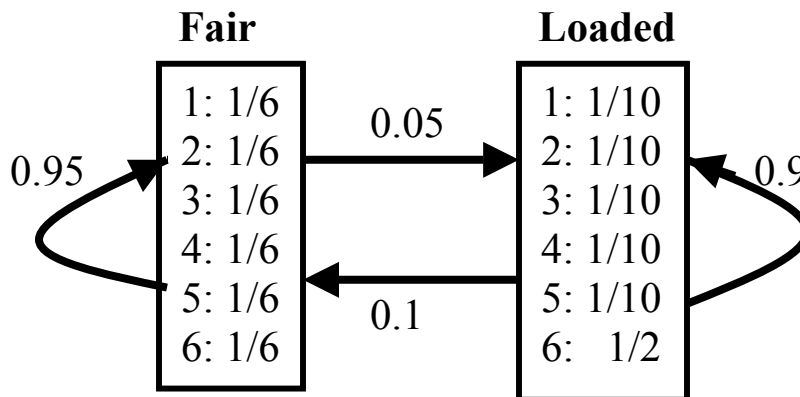


Figure 2.1 The occasionally dishonest casino [17]

In this model, given a sequence of rolls, we do not know which rolls used a loaded die and which used a fair die. It is no longer possible to tell what state the model was in when x_i was generated just by looking at x_i . This means the state sequence (π) is “hidden” from the symbol sequence (X). Therefore, we need to distinguish the symbol sequence X from the state sequence $\pi = (\pi_1, \dots, \pi_L)$. State transition probabilities follow a simple Markov chain, so the probability of a state depends only on the previous state:

$$a_{kl} = P(\pi_i = l \mid \pi_{i-1} = k).$$

Because we have decoupled the symbols b from the states k , we must introduce a new set of parameters for the model, $e_k(b)$. We therefore define the probability that symbol b is seen when in state k , which is called the *emission probability*:

$$E_k(b) = P(x_i = b \mid \pi_i = k).$$

Now it's easy to write down the joint probability of an observed sequence x and a state π :

$$P(x, \pi) = a_{0\pi_1} \prod_{i=1}^L e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}},$$

where a is the transition probability and e is the emission probability.

There are several algorithms to find out what the observed sequence means by considering the underlying states. The most common one is called the *Viterbi algorithm* [20]. It is a dynamic programming algorithm that takes a given HMM M and a sequence X and determines the most likely path through M that produced X . This tells us how likely X is modeled by M , and thus “scores” X with respect to how much it resembles the sequences used to build M . To build an HMM in the first place, we need to estimate its

parameters. We can use the *Baum-Welch algorithm* to solve this problem: Let A_{kl} be the number of transitions k to l , and $E_k(b)$ = number of emissions of b from k in training data.

We start with arbitrary $P(l | k)$ and $P(b | k)$, and use them to calculate A_{kl} and $E_k(b)$ as the expected numbers of time each transition or emission is used, given the training set.

Then these values are used to recompute $P(l | k)$ and $P(b | k)$, and the process continues until a convergence criterion is met.

A known HMM tool, HMMER 2.1.1, is used for this thesis. HMMER was designed by Sean Eddy from the Washington University School of Medicine [21]. HMMER uses profile hidden Markov models (profile HMMs), which are statistical models of the primary structure consensus of a sequence family. They are used to model a multiple alignment of a protein family, using position-specific scores for amino acids, insertion and deletion. Profile HMMs are useful for searching databases to find more homologies and for aligning sequences to the family. HMMER uses the Viterbi algorithm to build models. A program in HMMER, *hmmbuild*, reads a multiple sequence alignment file, builds a new profile HMM, and saves the HMM in a file. Then, another program, *hmmcalibrate*, calibrates the hidden Markov model. Finally, the program *hmmsearch* in HMMER reads an HMM and searches sequences database for significantly similar sequence matches. The output file has a list of ranked top hits, sorted by E-value, most significant hit first. Score is used to measure the significance of similarities between the sequence in the database and the HMM. The E-value (expectation value) is the expected number of hits to have the corresponding score or more just by chance in a sequence database of the current size.

2.5 Protein Secondary Structure Prediction

One of our approaches of finding new Trx proteins requires us to obtain the secondary structure of the protein we want to classify. There are four different types of protein structures: primary, secondary, tertiary, and quaternary. A protein's linear sequence of amino acids is called its primary structure. The regular folding of a protein in repeated patterns is called its secondary structure. Protein secondary structure includes α -helices, β -sheets and other structures. They can be exactly determined by nuclear magnetic resonance (NMR) or X-ray crystallography [22, 23]. However, these are very time-consuming, expensive processes. To predict it we can use Chou-Fasman method [24], Artificial neural networks (ANNs) [25, 26], Hidden-Markov Models (HMMs) [27], or a combination of multiple classifiers [28, 29].

The tool we used to predict secondary structure is PREDATOR, which uses the primary structure as the input. PREDATOR uses an artificial neural network for prediction [30]. PREDATOR does not use multiple sequence alignment. Instead, it relies on careful pairwise local alignments of the sequences in the set with the query sequence to be predicted. Only significant alignment fragments are subsequently considered. The secondary structure propensities of the auxiliary-related sequences are combined with (projected onto) those of the base sequence and weighted according to their degree of similarity. PREDATOR has 75% prediction accuracy, which is very high for this problem. The input for PREDATOR's ANN is the amino acid sequence of the protein. It is represented as a binary encoding of the amino acids in the window. The output is a prediction of α , β or coil for each residue in the sequence.

Chapter 3 Data Analysis Approaches

In this study, we developed two approaches to identify new Trx-fold proteins in the Trx superfamily. First, we applied hidden Markov models to secondary structure patterns. Second, we applied hidden Markov models to the remapped sequences with a reduced alphabet that captures structural properties of the amino acids.

3.1 Using Large Data Sets to Train and Test HMMER

To test how well our techniques identify homologous sequences in similar sequences that it has trained on, we first applied HMMER on a large data set for training and testing. We built a HMM on 47 Trx-fold sequences, and used it to test a data set with 226 positive sequences and 320 negative sequences. (Details of this experiment will be explained in Section 3.2) Some sequences in the training set are similar to the sequences in the test set. We first used the primary structures of the above data set for training and testing. We then used HMMs built on both true secondary structures and predicted secondary structures, and tested and compared the results between HMMs on primary structures and HMMs on secondary structures.

3.2 Hidden Markov Models on Secondary Structure Patterns

The positive data set we used for training and testing HMMER is chosen from the PDB database, which is obtained from the website <http://www.rcsb.org/>. The PDB database contains about 30,000 sequences. These sequences are in a FASTA format file. In a FASTA file, each sequence is preceded by a line starting with “>”. The first word on

this line is the name of the sequence. The rest of the line is a description of the sequence (free format). The remaining lines contain the sequence itself. The sequence of alphabet is the primary structure of the protein, which represents amino acid sequence.

Figure 3.1 summarizes the data flow in our experiment.

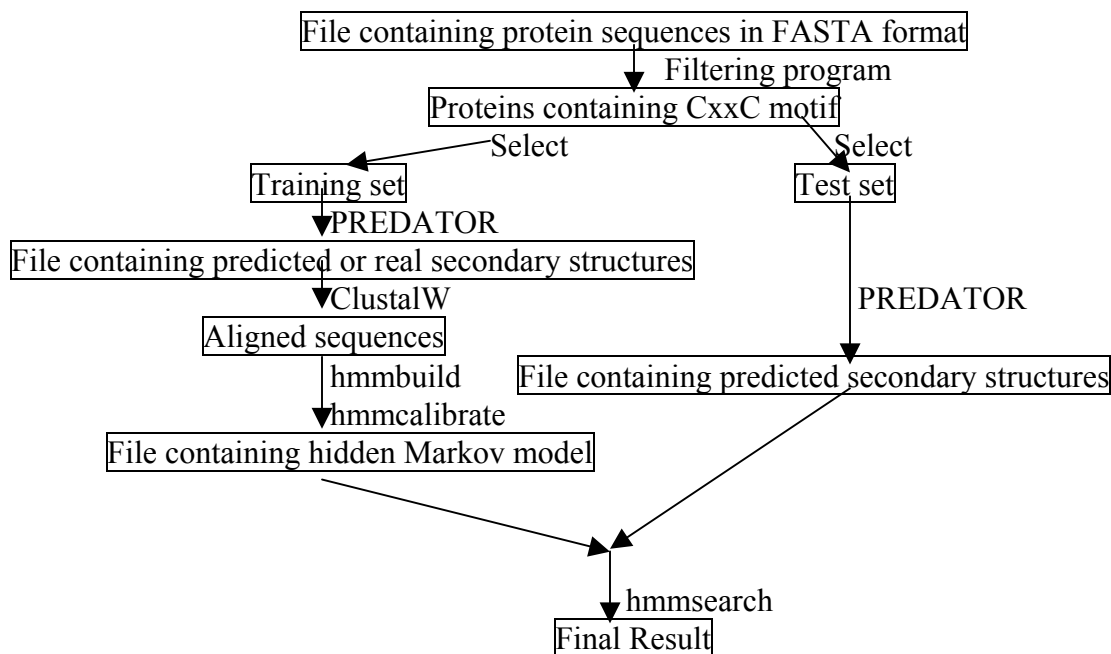


Figure 3.1 A Flowchart summarizing the data flow

The negative data set we used is chosen from the data set provided by Dr. Dmitri Fomenko from Professor Vadim Gladyshev's lab in Department of Biochemistry (University of Nebraska-Lincoln). This data set is selected from the Non-redundant Database, which can be accessed at <http://www.ncbi.nlm.nih.gov>. It is also in FASTA format.

Then we used the hidden Markov model package HMMER as a tool for training and test. To train HMMER, we used sequences containing Trx-fold. (HMMER does not need negative data for training.) Since most Trx-fold proteins contain CxxC motif, we

decided to only use this type of protein as positive sequences for our current experiments. We did so by writing a C program which only chooses sequences with CxxC motif, and running that program on the PDB database. Then, we obtained a Trx-fold protein database by searching within the proteins with CxxC motif with keywords like “thioredoxin” or “glutaredoxin”, and then verifying Trx superfamily membership with a human expert. As a result, we got a data set with 47 Trx-fold proteins with CxxC motif. Within the 47 sequences, there are many sequences similar to each other. Since the goal is to identify new families, we filtered the database so that only proteins with low primary structure conservation between each other are remained. We first used an alignment tool ClustalW to align those proteins. ClustalW takes a file with multiple sequences as the input, and outputs the aligned files. After aligning them, a dendrogram tree is also generated with the level of similarity between each pair of proteins listed. We selected the 9 of the 47 proteins which have the minimum level of similarity between each pair of them. They can be found at Appendix B. These 9 positive sequences were put in a file.

Now we need to obtain the secondary structures of these sequences. Since our final goal is to build a search engine to do database search, and true secondary structures are not available in those databases, we used PREDATOR to predict the secondary structures of the sequences in the positive and negative data set. True secondary structures are available in the PDB database, where Trx-fold proteins are selected and used to train HMMER. For comparison, we also used true secondary structures to train HMMER.

After that we used this data set for jack-knife test on HMMER, i.e., removing one sequence at a time, and each time training on the remaining 8 sequences. We then tested on the held out sequence and a set of negative sequences. (This technique is also called leave-one-out test.) The negative data set was selected from the Non-redundant Database. First, we chose all the sequences that contain CxxC motif from the negative database. Then, we used ClustalW to align the sequences and remove the sequences which were similar to another sequence in the data set. This resulted in 168 negative sequences for testing. As indicated previously, we used PREDATOR to get the secondary structures of those negative sequences.

After obtaining both the positive data set and the negative data set, we trained HMMER on true secondary structure and predicted secondary structure. This enabled us to compare the results between these two models. We first aligned the protein sequences using ClustalW. After aligning, most sequences' CxxC motifs are aligned. However, there is no conservation in other parts of the sequences. After aligning, we applied a program in HMMER, hmmbuild, to the aligned sequence to build a hidden Markov model. This program reads a multiple sequence alignment file, builds a new profile HMM, and saves the HMM in a file. Then we used a program hmmscalibrate in HMMER to calibrate the hidden Markov model, so that the search results will be more accurate. hmmscalibrate determines appropriate statistical significance parameters for a profile HMM prior to doing database searches. This program reads an HMM file, scores a large number of synthesized random sequences with it, fits an extreme value distribution (EVD) to the histogram of those scores, and re-saves the HMM file now including the EVD parameters.

Then we tested HMMER on the predicted secondary structures. We used a program `hmmsearch` in HMMER to search for the database of the predicted secondary structures to find Trx-fold proteins. We also applied HMMER on primary structures and compared the results to the results of HMMER on true secondary structures and HMMER on predicted secondary structures. The results are reported in Chapter 4.

3.3 Hidden Markov Models on Remapped Sequences

Andorf et al. [31, 32] remapped the 20-character AA alphabet to a reduced one that captures structural properties. They based the remappings on hydrophobicity, charge, volume and mass. They used the reduced alphabet representations of protein sequences in the data-driven discovery of sequence motif-based decision trees for classifying protein sequences into functional families. They constructed the classifiers using motifs generated using a multiple sequence alignment-based motif discovery tool. Results of their experiments on a data set of 11 protease families show that the classification performance of the resulting decision trees based on several reduced alphabets is comparable to that of trees based on the 20-letter amino acid alphabet. This raises the possibility that the use of different alphabets might provide different, but complementary, insights into protein structure-function relationships.

In our studies, we applied hidden Markov models to the remapped sequences with reduced alphabet that capture structural properties of amino acids. For this experiment, the positive and negative data sets are the same as in Section 3.2. The training and testing processes are also the same as in Section 3.2. The difference is, instead of mapping the primary structures to a sequence of secondary structure elements, we

remapped the 20-character amino acid alphabet to a reduced one. We tried remappings based on hydrophobicity, charge, volume and mass. Then we trained and tested HMMER based on the remapped sequences. But only the remappings based on hydrophobicity yielded good results (Section 4.1). Thus we think hydrophobicity is an important feature of this superfamily. We built and tested HMMER based on remapping the 20-character amino acid alphabet to 4 characters and 6 characters, respectively. The details of remapping are shown in Table 3.1 and Table 3.2.

Table 3.1 Remapping to 3 characters based on charge. The characters in each pair of curly braces are remapped to one single character.

Amino acid sets	Charge
{A C F G I L M N P Q S T V W Y}	No charge
{D E H}	Negative
{K R}	Positive

Table 3.2 Remapping to 4 characters based on volume. The characters in each pair of curly braces are remapped to one single character.

Amino acid sets	Volume
{F W Y}	Large
{E H I K L M Q V R}	Medium-Large
{C D N P T}	Medium
{A G S}	Small

Table 3.3 Remapping to 4 characters based on mass. The characters in each pair of curly braces are remapped to one single character.

Amino acid sets	Mass
{F R W Y}	Large
{D E H I K L M N Q}	Medium-Large
{C P S T V}	Medium
{A G}	Small

Table 3.4 Remapping to 4 characters based on hydrophobicity. The characters in each pair of curly braces are remapped to one single character.

Amino acid sets	Hydrophobicity range
{R D K E}	8.2 – 12.3
{N Q H}	3.0 – 4.8
{Y P S G T A W C}	-2.0 – 0.7
{V L I M F}	-3.7 – -2.6

Table 3.5 Remapping to 6 characters based on hydrophobicity. The characters in each pair of curly braces are remapped to one single character.

Amino acid sets	Hydrophobicity range
{R}	12.3
{D K E}	8.2 – 9.2
{N Q H}	3.0 – 4.8
{Y P}	0.2 – 0.7
{S G T A W C}	-0.6 – -2.0
{V L I M F}	-3.7 – -2.6

3.4 Database Search

We built HMMs on 9 positive sequences with true and predicted secondary structures and reduced alphabets by remapped sequences based on hydrophobicity, charge, volume and mass. We used that model to search for *Campylobacter jejuni* database to find Trx-fold proteins. The reason to use the data set with 9 sequences of the large data set with 47 sequences (in Section 3.3) is to avoid bias. In the data set with 47 sequences, there are many sequences similar to each other. Some sequences has more sequences similar to them than other sequences. So we only used the data set with 9 sequences to make sure these sequences are not similar at all.

Chapter 4 Experimental Results and Discussion

4.1 Results of HMMER on Large Data Sets

We built an HMM on 47 trx-fold sequences, and used it to test a data set with 226 positive sequences and 320 negative sequences. The sequences in the test set are filtered at the level of 80% or less similarity by a program provided by Dr. Stephen Scott from the Department of Computer Science & Engineering at the University of Nebraska-Lincoln. Some sequences in the training set are similar to the test set. We first used the primary structures of the above data set for training and testing. We found that HMMER trained on primary structure can achieve a true positive rate and a true negative rate at the level of more than 99%. This shows that HMMER trained on primary structure is very effective at finding sequences so long as the model was built on other, related sequences (related in primary structure), even if the relationship was remote. However, Section 4.2 shows that if the proteins are unrelated in primary structure, HMMER is not very effective.

We then used HMMs built on both true secondary structures and predicted secondary structures, and tested and compared the results between these two experiments. The true positive rate and the true negative rate are dependent upon the cut-offs of the scores or e-values of the result of the HMMER search. By lowering the cut-offs, the true positive rate will increase, but the true negative rate will decrease. So we plotted a graph based on the true positive and true negative rates verses the cut-off scores.

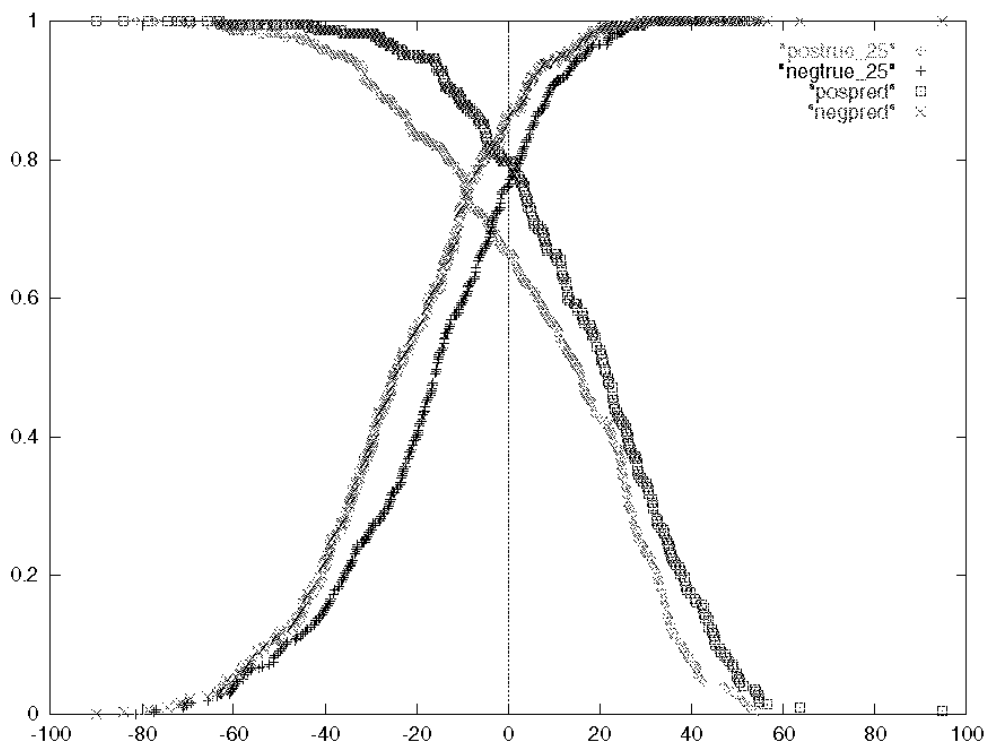


Figure 4.1. HMMER on large data set. The x-coordinate is the score. The y-coordinate is the true positive rate or $1 - [\text{true negative rate}]$. To make plot of true secondary structure and predicted secondary structure aligned, we add 25 to the score of each sequence of true secondary structure.

From this graph, we can see that HMMER trained on predicted secondary structures produces a better result than HMMER trained on true secondary structures. HMMER trained on predicted secondary structures can achieve both true positive and false positive rate at about 82%, while HMMER trained on true secondary structure can only achieve both true positive and false positive rate at about 70%. The reason is training HMMER on predicted secondary structure can reduce the noise introduced by the false prediction of PREDATOR. Since HMMER on primary structure can achieve true positive rate and true negative rate at the level of more than 99%, it is good enough

to identify new sequences in similar sequences that it has trained on. However, in the next section we will show that this is not the case for the jack-knife test.

4.2 Jack-knife Test Results

We used HMMER to attempt to identify new Trx-fold protein families based on primary structure alone (Section 3.3). In this test, 0% of distinct Trx-fold protein families were identified (Table 4.2), indicating that primary structure alone is insufficient to find distinct Trx-fold families. Therefore, in this study, we use structural properties to identify these Trx-fold families. These structural properties include secondary structure patterns, as well as remapped sequences with a reduced alphabet that captures structural properties. To select the best structural property for remapping sequences with a reduced alphabet, we tried remappings based on hydrophobicity, charge, volume and mass. We used remapped alphabet to train and test HMM, using the jack-knife method. The results are shown in Table 4.1. From Table 4.1 we can see that the remapping based on hydrophobicity shows the best result.

Table 4.1 Jack-knife test on different types of remapped alphabet.

Remapping methods	By charge	By volume	By mass	By hydrophobicity (in alphabet size 4)
True positive rate	0%	0%	22.2%	44.4%
False positive rate	0%	1.4%	2.6%	4.6%

We trained HMMER on true secondary structure, predicted secondary structure, and remapped alphabets. The results are shown in Table 4.2.

Table 4.2 Jack-knife test. All the cutoffs are set to 0.1, except the mapping cut-off is set to 0.01. This is because the overall scores of mapping is less than that of primary or secondary structures. TP means true positive rate. FP means average false positive rate. The meaning of E-value was explained in section 2.6.

PDB ID	Primary		Secondary (true)		Secondary (predicted)		Mapping (Hydrophobicity, 4 groups)		Mapping (Hydrophobicity, 6 groups)	
	E-value	FP	E-value	FP	E-value	FP	E-value	FP	E-value	FP
1a8l	0.83	0	9.9e-3	9	3.5e-4	25	>10	12	>10	1
1eej	2.9	0	5.3	9	3.3e-3	22	>10	1	>10	3
1bed	5.8	0	>10	12	0.22	23	7.9e-3	10	9.8	0
1qk8	>10	0	6.2e-3	15	4e-7	27	0.016	3	0.036	5
1f9m	6.2	0	0.55	10	3.9e-4	26	1.2e-3	11	2.9e-3	3
1mek	0.62	0	0.019	18	6.2e-6	25	6.4e-3	3	3.5e-3	5
1ego	6.7	0	>10	14	0.41	16	8e-3	20	0.081	6
1fov	>10	0	>10	9	7.2e-4	34	0.075	5	0.16	5
1del	>10	0	>10	9	2.4e-3	21	0.26	5	0.026	15
TP	0%		33.3%		77.8%		44.4%		22.2%	
FP	0%		6.9%		14.5%		4.6%		2.8%	

From this result, we can see that HMMER trained on primary structure cannot find Trx-fold proteins with very low similarity. HMMER trained on secondary structure or remapped alphabets can identify new families of Trx-fold proteins. In particular, two out of three glutaredoxin proteins are found. The result of HMMER on secondary structure is better than HMMER on remappings. This is probably due to the highly conserved secondary structure elements in the Trx-fold. However, there is not much correlation between these two methods on which sequences were missed. Therefore, some sequences which were missed by one classifier were found by the other classifier. This raises the possibility of combining two classifiers to find more Trx-fold proteins. We also found that HMMER trained on predicted secondary structure generates better results than HMMER trained on true secondary structures. We believe that this is due to

the noise introduced by PREDATOR. In the test data set, the secondary structures of the negative sequences are predicted by PREDATOR. This software has an error of about 25%. Therefore, about 25% of the predicted secondary structure is wrong. This results in some noise in the test set. But if we use the predicted secondary structure for training, since both the training and test data set has secondary structures predicted by PREDATOR, it seems that the noise is correlated between testing and training. Therefore, HMMER will produce better results when trained and tested on predicted secondary structure. Ideally, we should use true secondary structure for both training and test. But this is not feasible because the ultimate objective of this research is to build a search engine to find new Trx-fold protein families and the true secondary structures for many proteins are unavailable currently, since finding them via nuclear magnetic resonance (NMR) or X-ray crystallography is a time-consuming, expensive process.

We can also see that although the true positive percentage has raised 77.7%, the false positive percentage has also been raised by a noticeable degree — 14.5%. This is a drawback of using secondary structure. But this increase is tolerable given the increase in true positive rate.

The letters in the alphabet for the secondary structure has different meanings compared to the 20 letters in the alphabet for the primary structure. And the default protein weight matrix used by ClustalW is constructed based on the 20 amino acid alphabet, with the consideration of the hydrophobicity and other properties of the amino acids. So changing the amino acid alphabet to a reduced alphabet might affect the correctness of the protein weight matrix. Therefore, we used an identity matrix for ClustalW when we align the remapped sequences. The identity matrix gives a score of

1.0 to two identical symbols and a score of zero otherwise. We found that the difference between the old alignments using PAM matrix and the new alignments using the identity matrix. We also used the new alignments to train HMMER, and used the HMM to do jack-knife test and database search. First, we did jack-knife test on the data set of 9 Trx-fold proteins with predicted secondary structure, and found that HMMER identified 6 sequences out of the 9 sequences (compared to identifying 7 sequences out of the 9 sequences from the old test). Second, we searched the *Campylobacter jejuni* database using the predicted secondary structure. We found 5 Trx-fold proteins in the database (compared to finding 6 Trx-fold proteins in the database). We also searched the *Campylobacter jejuni* database using the remapped sequence based on hydrophobicity with alphabet size 6. We found 2 Trx-fold proteins in the database (compared to finding 4 Trx-fold proteins in the database). These results indicate that our choice of protein weight matrix in our experiment will not affect the experiment results significantly.

Another issue is of the insertion at the secondary structure level. Some Trx-fold proteins have insertions in some areas of the Trx-fold. This makes it more difficult to discover them. In our future work we will try to solve this problem.

4.3 Results of Database Search

To test our tool on protein databases, we built an HMM based on 9 positive sequences with predicted secondary structures, and used that model to search for the *Campylobacter jejuni* database to find Trx-fold proteins. The results are listed in Table 4.3.

Table 4.3. Trx-fold proteins found at the Campylobacter jejuni database using HMM built on predicted secondary structures

CAB73132.1	protein disulphide isomerase
CAB73652.1	lipoprotein thiredoxin
CAB73651.1	periplasmic thioredoxin
CAB73130.1	disulfide oxidoreductase

From Table 4.3, we can find that there are 4 Trx-fold proteins found at the *Campylobacter jejuni* database. (These proteins are verified by a human expert.) This indicates that our tool has the potential of finding many new Trx-fold proteins in protein databases. We also used HMMER to search primary structures. None of the above 4 proteins were found.

We also built HMMs on 9 positive sequences with true secondary structures and with mappings based on hydrophobicity, charge, volume and mass, and used those models to search for the *Campylobacter jejuni* database to find Trx-fold proteins. The results are listed in Table 4.4.

Table 4.4. Numbers of Trx-fold proteins found at the Campylobacter jejuni database using different models

	Primary	Secondary (true)	Secondary (predicted)	Mapping (hydrophobicity, 4 groups)	Mapping (hydrophobicity, 6 groups)	Mapping (charge)	Mapping (volume)	Mapping (mass)
TP	1	3	4	4	6	3	2	4
ID	CAB72631.1	CAB73361.1 CAB73461.1 CAB73652.1	CAB73132.1 CAB73652.1 CAB73651.1 CAB75273.1	CAB73461.1 CAB73652.1 CAB72631.1 CAB73651.1	CAB73361.1 CAB73652.1 CAB72631.1 CAB73651.1 CAB73807.1 CAB73627.1	CAB73461.1 CAB73652.1 CAB73651.1	CAB73461.1 CAB73651.1	CAB73461.1 CAB73652.1 CAB72631.1 CAB73651.1

From Table 4.4, we can see that remappings based on hydrophobicity with alphabet size 6 is better than using other structural properties for remapping, because using this method we found 6 Trx-fold proteins. There is not much correlation between these two methods on which sequences were missed. Therefore, some sequences which were missed by one classifier were found by the other classifier. So we can combine two classifiers to find more Trx-fold proteins.

Chapter 5 Conclusions and Future Development

In this thesis, we used hidden Markov models to identify new Trx-fold proteins in protein superfamilies with low primary structure conservation. Our techniques are based on exploiting conserved structural patterns, including secondary structure patterns, as well as the remapped sequences with a reduced alphabet that captures structural properties. We evaluated our methods on protein sequences with a thioredoxin fold, a very important protein superfamily. To predict the secondary structure of candidate sequences, we used a protein secondary structure prediction tool, PREDATOR. We also wrote a program which can remap the protein primary structures to a reduced alphabet. We used HMMER to model and search for Trx-fold proteins. From the result, we can see that hidden Markov models on primary structures cannot identify new Trx-fold proteins effectively. Hidden Markov models on secondary structures and hidden Markov models on remapped sequences can identify a considerable amount of new Trx-fold protein families. We also found that the former approach is better than the latter approach in our experiments.

We also used our new classifier to search a larger database. Even though by lowering the cut-offs, both the true positive rate and the false negative rate will increase, a good cutoff exists that can make both the true positive rate and the true negative rate reasonably high. We also found that HMMER trained on predicted secondary structures produces better results than HMMER trained on true secondary structures. This is because training HMMER on predicted secondary structure generates a model based on the noise introduced by the false predictions of HMMER.

We also built an HMM to search the *Campylobacter jejuni* database for Trx-fold proteins. We found 6 such proteins. This indicates that our tool has the potential to find many new Trx-fold proteins in protein databases.

We will continue our database search to look for Trx-fold proteins in protein databases of different species, including *Methanococcus jannaschii*, *Escherichia coli*, and *Saccharomyces cerevisiae*. Besides Trx-fold proteins, we also want to find other redox proteins. We want to identify the majority of disulfide oxidoreductases in completely sequenced genomes. Eventually, we will also do function classifications of discovered proteins.

References

1. Salzberg, S. L., D. B. Searls and S. Kasif (Eds.). (1998). *Computational Methods in Molecular Biology*. Elsevier Science B.V.
2. Finkel, T. and N. J. Holbrook. (2000). Oxidants, oxidative stress and the biology of ageing. *Nature* 408: 239-247.
3. Thannickal, V. J. and B. L. Fanburg. (2000) Reactive oxygen species in cell signaling. *Am. J. Physiol. Lung Cell. Mol. Physiol.* 279: L1005-28.
4. Zheng, M., F. Aslund and G. Storz. (1998) Activation of the OxyR transcription factor by reversible disulfide bond formation. *Science* 279: 1718-1721.
5. Holmgren, A. (1989). Thioredoxin and glutaredoxin systems. *J. Biol. Chem.* 264: 13963-13966.
6. Debarbieux, L. and J. Beckwith. (1999). Electron avenue: pathways of disulfide bond formation and isomerization. *Cell* 99: 117-119.
7. Aslund, F., and J. Beckwith. (1999). The thioredoxin superfamily: redundancy, specificity, and gray-area genomics. *J. Bacteriol* 181: 1375-1379.
8. Martin, J. (1995). Thioredoxin – a fold for all reasons. *Structure* 3: 245-250.
9. Holmgren, A. and M. Bjornstedt. (1995). Thioredoxin and Thioredoxin Reductase. *Methods in Enzymology* 252: 199-208.
10. Theodoridis, S. and K. Koutroumbas. (1999). *Pattern Recognition*. Academic Press, London.

11. Rosenblatt F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65: 386-408.
12. Muggleton, S. H., C. H. Bryant, A. Srinivasan, A. Whittaker, S. Topp and C. Rawlings (2001). Are grammatical representations useful for learning from biological sequence data? – A case study. *Journal of Computational Biology* 8: 493-521.
13. Luthy, R., A. McLachlan and D. Eisenberg (1991). Secondary Structure-Based Profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities. *Proteins: Structure, Function, and Genetics* 10: 229-239.
14. Bowie, J. U., R. Luthy and D. Eisenberg (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science* 253:164-168.
15. Eddy, S. R. (1996). Hidden Markov Models. *Current Opinion in Structural Biology*, 6: 361-365.
16. Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE*, 77: 257-286.
17. Churchill, G. A. (1989). Stochastic models for heterogeneous DNA sequences. *Bull Math Biol*, 51: 79-94.
18. Stultz, C. M., J. V. White and T. F. Smith (1993). Structural analysis based on state-space modeling. *Protein Sci*, 2: 305-314.

19. Durbin, R., S. Eddy, A. Krogh and G. Mitchison. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.
20. Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 13: 260-269.
21. Eddy, S. (1998). *HMMER User's Guide*. 4566 Scott Ave., St. Louis, MO 63110, USA. Version 2.1.1 edition.
22. Blundell, T. and L. Johnson (1976). *Protein crystallography*. Academic Press, London.
23. Kaptein, R., R. Boelens, R. Scheek and W. van Gunsteren (1988). Protein structures from NMR. *Biochemistry* 27: 5389-5395.
24. Chou, P.Y., and G. D. Fasman (1974). Prediction of protein conformation. *Biochemistry* 13: 222-245.
25. Holley L. H. and M. Karplus. (1989). Protein secondary structure prediction with a neural network. *Proc Natl Acad Sci USA*, 86: 152-156.
26. Rost, B. and C. Sander. (1993). Prediction of protein secondary structure at better than 70% accuracy. *J Mol Biol*, 232: 584-599.
27. Asai, K., S. Hayamizu and K. Handa. (1993). Prediction of protein secondary structure by the hidden Markov model. *CABIOS*, 9: 141-146.
28. Zhang, X., J. P. Mesirov and D. L. Waltz. (1992). Hybrid system for protein secondary structure prediction. *J Mol Biol*, 225: 1049-1063.

29. Rost, B. and C. Sander. (1994). Combining evolutionary information and neural networks to predict protein secondary structure. *PROTEINS: Structure, Function, and Genetics*, 19: 55-72.
30. Frishman, D. and P. Argos (1997). Seventy-five percent accuracy in protein secondary structure prediction. *Proteins* 27: 329-335.
31. Andorf, C. M., D. L. Dobbs and V. G. Honavar. (2002). Discovering protein function classification rules from reduced alphabet representations of protein sequences. *4th Conference on Computational Biology and Genome Informatics* 1200-1206.
32. Wang, X., D. Schroeder, D. Dobbs, and V. G. Honavar. (2002). Data-driven discovery of protein function classifiers: Decision trees based on MEME motifs outperform Prosite patterns and profiles on peptidase families. *4th Conference on Computational Biology and Genome Informatics* 1193-1199.

Appendix A: Examples of Protein Primary And Secondary Structure

1. Protein primary structure:

-M-G-L-I-S-D-A-D-K-K-V-I-K-

This is a segment from a protein disulfide oxidoreductase (PDB ID 1A8L). Each letter represents an amino acid residue.

2. Protein secondary structure:

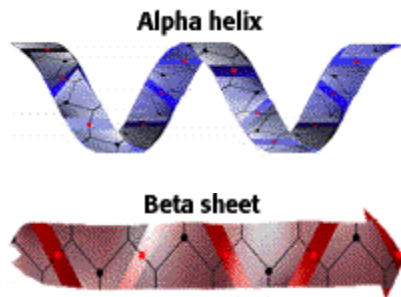


Figure A Protein secondary structure: α -helix and β -sheet, from http://www.biology.arizona.edu/biochemistry/problem_sets/large_molecules/03t.html

Appendix B: Positive Data Set Used for Jack-knife

Test (in FASTA Format)

```
>1A8L:_ PROTEIN DISULFIDE OXIDOREDUCTASE
MGLISDADKKVIKEEFFFSKMNPNVKLIVFVRKDHCOYCDQLKQLVQELSELTDKLSYEIV
DFDTPEGKELAKRYRIDRAPATTITQDGKDFGVRYFGLPAGHEFAAFLEDIVDVSREETN
LMDETKQAIRNIDQDVRILVFVTPTCPYCP LAVRMAHKFAIENTKAGKGKILGDMVEAIE
YPEWADQYNVMAVPKIVIQVNGEDRVEFEGAYPEKMFLEKLLSALS
```

```
>1EEJ:A THIOL:DISULFIDE INTERCHANGE PROTEIN
DDAAIQQTLAKMGIKSSDIQPAPVAGMKTVLTNSGVLYITDDGKHIIQGPMYDVSGTAPV
NVTNKMLLKQLNALEKEMIVYKAPQEKHVITVFTDITCGYCHKLHEQMADYNALGITVRY
LAFPRQGLDSDAEKEMKAIWCAKDKNKAFDDVMAGKSVAPASCDVDIADHYALGVQLGVS
GTPAVVLSNGTLVPGYQPPKEMKEFLDEHQKMTSGK
```

```
>1BED:_ DSBA OXIDOREDUCTASE
AQFKEGEHYQVLKTPASSSPVVSEFFSFYCPHCNTFEPPIIAQLKQQLPEGAKFQKNHVSF
MGGNMGQAMSKAYATMIALEVEDKMVPVPMFNR IHTLRKPPKDEQELRQIFLDEGIDAAKF
DAAYNNGFAVDSMVRFRDKQFQDSGLTGVPVAVVNNRYLVQGQSVKSLDEYFDLVNYLLTL
K
```

```
>1QK8:A TRYPAREDOXIN-I
MSGLDKYLPGIEKLRRGDGEVEVKSLAGKLVFFYFSASWCPPCRGFTPQLIEFYDKFHES
KNFEVVFCTWDEEEDGFAGYFAKMPWLAVPFAQSEAVQKLSKHFNVESIPTLIGVDADSG
DVVTTTRARATLVKDPEGEQFPWKDAP
```

```
>1F9M:A THIOREDOXIN F
MEAIVGKVTEVNKDTFWPIVKAAGDKPVVLDMFTQWCGPCKAMAPKYEKLAEEYLDVIFL
KLDCNQENKTLAKELGIRVVPTFKILKENSVVGEVTGAKYDKLLEAIQAARS
```

```
>1MEK:_ mol:protein length:120 Protein Disulfide
Isomerase
DAPEEEDHVLVLRKSNFAEALAAHKYLLVEFYAPWCGHCKALAPEYAKAAGKLLKAEGSEI
RLAKVDATEESDLAQQYGVRYPTIKFFRNGDTASPKEYTAGREADDIVNWLKRTGPAA
```

```
>1EGO:_ GLUTAREDOXIN (OXIDIZED) (NMR, 20 STRUCTURES) - CHAIN
-
MQTVIFGRSGCPYCVRAKDLAEKLSNERDDFQYQYVDIRAEGITKEDLQQKAGKPVETVP
QIFVDQQHIGGYTDFAAWKENLDA
```

```
>1FOV:A GLUTAREDOXIN
ANVEIYTKETCPYCHRAKALLSSKGVSFQELPIDGNAAKREEMIKRSGRTTVPQIFIDAQ
HIGGYDDLALDARGGLDPLLK
```

```
>1DE1:A GLUTAREDOXIN
MFKVYGYDSNIHKCVYCDNAKRLLTVKKQPFEFINIMPEKGVFDDEKIAELLTKLGRDTQ
IGLTMPQVFAPDGS HIGGFQDLREYFK
```