

IDENTIFICATION OF THIOREDOXIN-FOLD PROTEINS WITH LOW  
PRIMARY STRUCTURE CONSERVATION

Xiaoping Wen

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

December, 2002

IDENTIFICATION OF THIOREDOXIN-FOLD PROTEINS WITH LOW  
PRIMARY STRUCTURE CONSERVATION

Xiaoping Wen, M.S.

University of Nebraska - Lincoln, 2002

Advisor: Stephen D. Scott

An exciting and potentially far-reaching development in contemporary computer science is the invention and application of methods of machine learning. These enable a computer program to automatically analyze a large body of data and decide what information is most relevant. This crystallized information can then be used to help people make decisions faster and more accurately. The purpose of this study was to develop a new machine learning tool to identify Thioredoxin (Trx)-fold proteins.

Many efforts have been put forth to find efficient ways to identify members of the thioredoxin superfamily. However, the lack of amino acid conservation among thioredoxin-fold proteins makes use of conventional research techniques difficult to identify new members of this protein superfamily. In this study, we characterized the statistical values of amino acid properties such as polarity, solubility, hydrophobicity, molecular weight, and amino acid count frequency. We applied the machine learning tool C4.5 to evaluate the properties of amino acids and to separate Trx-fold proteins from other proteins. We built a search engine which allows users to search any database in

FASTA format to find new Trx-fold proteins. We evaluated this new program via several experiments, including multiple database searches.

## **Acknowledgements**

I wish to thank Dr. Stephen Scott for his guidance, encouragement, and support throughout the course of this research work. I am also grateful to Dr. Vadim Gladyshev, Dr. Etsuko Moriyama, and Dr. Jitender Deogun, who served on my advisory committee and examining committee, for their advice and useful contributions to my work.

I also thank my fellow graduate student Haifeng Ji for his friendship and assistance with technical problems. Thanks are also extended to Dr. Dimitri Fomenko for his input in this project.

I thank my husband Joe Zhou and my two kids for their love and support.

# Table of Contents

---

	<b>Page #</b>
<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Table of Contents</b>	<b>iv</b>
<b>List of Tables</b>	<b>vi</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Motivation of this research	1
1.2 Thesis Outline	3
<b>Chapter 2 Background Review and Prior Work</b>	<b>4</b>
2.1 Biological Background	4
2.2 Thioredoxin-fold Proteins and Their Significance in Biology	5
2.3 Related Work	6
2.4 Computational Biology and Machine Learning Tools	7
2.5 Sliding Window and Search Engine	12
2.6 Experiment Design Approach	14
<b>Chapter 3 Experiment design and analysis</b>	<b>16</b>
3.1 Experiment 1: Testing on data randomly selected from PDB database	17
3.2 Experiment 2: Jack-knife tests on 47 Trx-fold proteins	17

3.3 Experiment 3: Testing the effect of homology	18
3.4 Experiment 4: Jack-knife tests on 9 Trx-fold proteins	18
3.5 Experiment 5: HMMER test to find 12 distant related Trx-fold proteins and Jack-knife tests on the 12 Trx-fold proteins	19
3.6 Experiment 6: Search database to find Trx-fold proteins	20
<b>Chapter 4 Results and discussion</b>	<b>21</b>
4.1 Results on Experiment 1: Testing on data randomly selected from PDB database	21
4.2 Results on Experiment 2: Jack-knife tests on 47 Trx-fold proteins	22
4.3 Results on Experiment 3: Testing the effect of homology	24
4.4 Results on Experiment 4: Jack-knife tests on 9 Trx-fold proteins	26
4.5 Results on Experiment 5: HMMER test to find 12 distant related Trx-fold proteins and Jack-knife tests on the 12 Trx-fold proteins	28
4.6 Results on Experiment 6: Search database to find Trx-fold proteins	30
<b>Chapter 5 Conclusions and Future Work</b>	<b>32</b>
<b>References</b>	<b>35</b>
<b>Appendix A: Positive data set used for jack-knife test</b>	<b>38</b>

## List of Tables and Figures

---

Table	Page #
1. <b>Table 2.1</b> Training data set for the baseball game example	10
2. <b>Table 2.5.1</b> A partial protein sequence and its mapped value.	13
4. <b>Table 4.1</b> Results of identifying Trx-fold protein by using Sliding Window method and C4.5 decision tree.	21
5. <b>Table 4.2</b> Results of jack-knife test on 47 Trx-fold proteins.	23
6. <b>Table 4.3</b> Results of identifying Trx-fold proteins with different homology as positive training data sets and tests data set by using Sliding Window method and C4.5 decision tree.	25
7. <b>Table 4.4</b> Results of Jack-knife tests on 9 Trx-fold proteins.	27
8. <b>Table 4.5</b> Results of HAMMER trained on primary structure and C4.5 Jack-knife tests on 12 Trx-fold proteins.	29
9. <b>Table 4.6.1</b> lists the Trx-fold proteins found from <i>Methanococcus jannaschii</i> database	30
10. <b>Table 4.6.2</b> lists the Trx-fold proteins found from <i>Campylobacter jejuni</i> database	30

**Figure**

**Figure 2.1** The output of the decision tree for the baseball game example

10

# Chapter 1 Introduction

## *1.1 Motivation of This Research*

Cellular reduction and oxidation (redox) status regulates various aspects of cellular functions such as proliferation, activation, growth inhibition and cell death. More and more evidence is accumulating that a proper balance between oxidants and antioxidants is involved in maintaining health and longevity, and altering this balance may cause functional disorders and disease [1]. Thioredoxin-fold protein (Trx) is a superfamily of proteins that contain the thioredoxin fold that is a distinct structural motif [2]. Trx-fold proteins are also redox proteins and are reported to be one of the important factors to maintain the redox environment of the cell [3, 4]. A large amount of research has been performed to understand the detailed mechanisms and role of thioredoxin systems in redox reactions and control of physiological processes. One of the major limitations to understanding the mechanism of cellular redox regulation is the lack of information on the identity of most redox proteins and on the specific functions of these proteins [5, 6]. Therefore, many efforts have been put forth to find efficient ways to identify members of the thioredoxin superfamily. However, the lack of amino acid conservation among thioredoxin-fold proteins makes use of conventional search techniques difficult to identify new members of this protein superfamily.

The large amount of nucleotide sequence and expression data, together with a recent trend of looking at a biological system as a whole, increases the demand for developing efficient bioinformatics tools. Because of large projects like the Human

Genome Project in the United States, there has been an exponential increase in the quantity of data available about proteins, DNA, and RNA. Traditional lab methods of studying the structure and function of these molecules are no longer able to keep up with the rate of new information. As a result, molecular biologists have turned to statistical methods capable of analyzing large amounts of data, and to computer programs which implement these methods.

A handful of other methods have been used to search for new Trx-fold proteins. Ji demonstrated that while effective in finding sequences related to known Trx-fold proteins, the popular hidden Markov model tool HMMER [8] when trained on primary sequences is ineffective in identifying new Trx-fold proteins for which no relatives are known. But when he used HMMER trained on structural information, he got stronger results. However, his work relied on accurate structure predictions of the proteins and required building multiple alignments of the sequences, both of which can be error-prone. In other work, research conducted in Dr. Gladyshev's laboratory [9] developed computational tool that automatically identifies conserved CxxS sequences in large sequence databases. This program first identifies all proteins containing CxxS motif and performs crude data reduction. It further eliminates unwanted protein by using metal-binding domains. The program searched for conservation of CxxS sequences and dramatically reduces unwanted data by using secondary structure prediction. The selected candidate redox proteins were then analyzed individually for homologies to proteins of known function, structure or domain. Since many proteins containing the CxxS motif had a thioredoxin fold, this tool can be used as a general approach of genome-wide identification of redox proteins including Trx-fold proteins.

All currently known Trx-fold protein families have been defined on the basis of experimentally determined structures of proteins that are representative members of the known families. Our aim is to develop a new bioinformatics method which uses the characteristics of protein amino acid sequences to identify new Trx-fold proteins for which no relatives are known, without the use of structure prediction or multiple alignments. During our research, we employed machine learning tools and developed a computer program to identify thioredoxin-fold proteins. We extract statistical data from known protein structural properties such as polarity, hydrophobicity, solubility, molecular weight, and amino acid frequency counts and mapped them to attributes of a machine learning algorithm C4.5 to identify Trx-fold proteins. We also built a computer search engine with known thioredoxin proteins and non-thioredoxin proteins and applied the search engine to search all completely sequenced genomes and other sequence databases. This work has led to the identification of a number of potential thioredoxin-fold proteins.

## ***1.2 Thesis Outline***

The remainder of this thesis is organized as follows. In Chapter 2 we described previous work and background, including the Sliding Window [10] and C4.5 Decision Tree algorithms [11] in our work. Chapter 3 describes experimental data preparation, experiment design, and the building of a Trx-fold protein search engine. Chapter 4 presents and discusses experimental results and the database search results. In Chapter 5, we present avenues for future work.

## ***Chapter 2 Background and Review***

### ***2.1 Biological Background***

*Proteins* are macromolecules made up from amino acids, also referred to as residues [12, 13]. There are 20 naturally occurring amino acids, each amino acid is represented by a capital letter, such as C = cysteine. All amino acids have a common chemical structure: a carbon atom ( $C_{\alpha}$ ) to which four asymmetric groups are connected: an amino group ( $NH_2$ ), a carboxy group ( $COOH$ ), a H atom and another chemical group (denoted by R) which carries from one amino acid to another. In a protein, amino acids are connected by peptide bonds. Two or more amino acids covalently linked by peptide bonds to form a peptide. Therefore, proteins are polypeptidic chains.

The *primary structure* of proteins is the sequence of amino acids from which it is constructed. In fact, proteins fold in three dimensions and form secondary, tertiary and quaternary structures. The two most common *secondary structure* arrangements are the right-handed  $\alpha$ -helix and the  $\beta$ -sheet, which can be connected into a larger *tertiary structure* by turns and loops of a variety of types. An  $\alpha$ -helix is a helix that makes a complete turn every 3.6 amino acids. The helix is right-handed and it twists in a clockwise direction. A  $\beta$ -sheet consists of pairs of chains lying side-by-side. Each chain is called a  $\beta$ -strand. A  $\beta$ -sheet is stabilized by hydrogen bonds between the carbonyl oxygen atom on one chain and the -NH group on the adjacent chain. Loops are

sequences that connect the other two types of secondary structure. A group of different proteins packed together results in the name *quaternary structure*.

We refer a *motif* to a small specific combination of secondary structure elements (such as helix-turn-helix) in this thesis. *Fold* refers to a global type of arrangement, like helix-bundle.

Proteins have many properties, such as size and shape, charge, polarity, hydrophilicity or hydrophobicity (attracts or repels water), solubility. Proteins that have evolved from a common ancestor are said to be homologous. Generally speaking, unrelated protein sequences show about 6% random identity. For long protein sequences 20% identity is usually taken to indicate homology. When homologous proteins are compared, one often finds regions that are conserved among many proteins. This is usually indicative of an essential common function for those regions – active sites in enzymes are often located this way.

## ***2.2 Thioredoxin-fold proteins and their significance in Biology***

The thioredoxins are small, dithiol proteins that play an important role in maintaining the redox balance in cells [14]. They have been identified in a number of organisms and are thought to be ubiquitous [15]. The best characterized is *Escherichia coli* thioredoxin, whose chemical properties and three-dimensional configuration have been studied in great detail [16]. The *Escherichia Coli* oxidized form was found to consist of a single domain with a central five-stranded  $\beta$ -sheet with four flanking  $\alpha$ -helices and a dithiol/disulphide group in the active site [17]. The thioredoxin superfamily

includes proteins all having the same basic folding as thioredoxin with the active site located at the C-terminal end and followed by an  $\alpha$ -helix [18]. There are some common features that can be used to identify Trx-fold proteins [19]. For example, despite any sequence differences, the three-dimensional structures are highly conserved across species[20]. At the active site, there is a characteristic CxxC motif. Typically, the sequence is -Cys-Gly-Pro-Cys-, but occasionally is -Cys-Ala-Pro-Cys- [21].

The role of thioredoxin in redox control of transcription factor activity affecting cell growth is an interest of biological research. In a series of ongoing studies stemming from the fundamental work, thioredoxin has emerged as a new tool in technology and medicine. The rapidly expanding knowledge of thioredoxin will lead to applications in industry and health. The large number of known Trx-fold proteins and the desire to find new thioredoxin family make this superfamily a promising target for protein identification by machine learning approaches. Our research involves using the machine learning algorithm C4.5 to identify thioredox based on known data in order to make predictions about unseen data.

### ***2.3 Related Work***

A few methods have been used to search for new Trx-fold proteins. Ji developed three approaches in identifying Trx-fold proteins. First he used the popular hidden Markov tool trained on primary structure. He demonstrated that while effective in finding sequences related to known Trx-fold proteins, the popular hidden Markov model tool HMMER [8] when trained on primary sequences is ineffective in identifying new Trx-

fold proteins for which no relatives are known. Then, he applied hidden Markov models to secondary structure patterns. He also applied Markov models to the remapped sequences with reduced alphabet that captures structural properties of amino acid. When he used HMMER trained on structural information, he got stronger results. However, his work relied on accurate structure predictions of the proteins and required building multiple alignments of the sequences, both of which can be error-prone.

Dr. Gladyshev's laboratory [9] developed computational tool that automatically identifies conserved CxxS sequences in large sequence databases through four steps. This program first identifies all proteins containing CxxS motif. It then filters out metal-binding proteins. At the third step, it searches for conservation of CxxS sequences. Finally it predicts secondary structure for proteins with conserved CxxS sequences. The selected candidate redox proteins are analyzed individually. This tool has been applied to recognize disulfide oxidoreductases, to identify such proteins in several completely sequenced bacterial, archaeal and eukaryotic genomes, and to provide examples of functional characterization of disulfide oxidoreductases and Trx-fold proteins.

## ***2.4 Computational Biology and Machine Learning Tools***

Computational biology is a fast growing field of computer science, synthesizing both the information and biological sciences. Its emergence as an independent field was driven by the development, in the late 1970s, of efficient experimental techniques in biology and by the resulting exponential growth in genomic data. Molecular biology is currently experiencing another period of fast technological progress, producing vast

amounts of valuable genomic and proteomic data. This growing body of data calls for efficient, integrated approaches to its storage, management and analysis. Therefore, large databases and sophisticated algorithms have become essential tools for scientists to understand complex biological systems, determine the functions of nucleotide and protein sequences, or reconstruct the course of evolution. Computational tools can be extremely useful for pruning the search space of experiments and deals with computational problems arising from biological systems.

Machine learning is the study of adaptive computational systems that improve their performance with experience [13]. An exciting and potentially far-reaching development in contemporary computer science is the invention and application of methods of machine learning. These enable a computer program to automatically analyze a large body of data and decide what information is most relevant. This crystallized information can then be used to help people make decisions faster and more accurately. One of the central problems of the information age is dealing with the enormous explosion in the amount of raw information that is available. Machine learning has the potential to sift through this mass of information and convert it into knowledge that people can use.

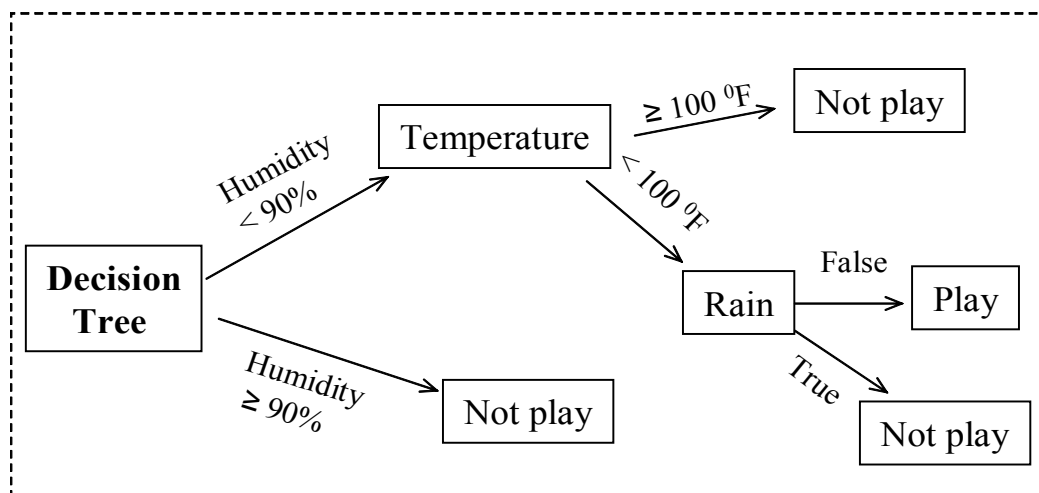
The methods of machine learning include supervised and unsupervised learning methods such as learning decision and regression trees, rules, connectionist networks, probabilistic networks and other statistical models, genetic algorithms, genetic programming, inductive logic programming, a case-based methods. Machine learning also deals with reinforcement learning, explanation-based leaning, and automated knowledge acquisition, leaning from instruction, visualization of patterns in data, and learning in integrated architectures [23].

C4.5 is a machine learning algorithm designed by Ross Quinlan [11]. It is a program for inducing classification rules in the form of decision trees from a set of given examples. C4.5 decision trees are excellent tools for helping to choose between several courses of action. They provide a highly effective structure within which one can lay out options and investigate the possible outcomes of choosing those options. Here we give a simple example on children's summer camp weekly baseball games to illustrate how the C4.5 and C4.5 rules work. Two summer camps want to decide whether there will be baseball games between these two camps every Wednesday during the whole summer. The games will be held based on temperature, humidity, and precipitation. They want a program to automatically predict when they would choose to play and adjust their appointment calendars. Since they are too busy to program it themselves, the machine has to take weather information from every Wednesday of last summer and build its own predictor for the coming weekend. In the training data set, they recorded the weather conditions for each Wednesday of last summer and also recorded whether or not they played in that Wednesday (see Table 2.1). The C4.5 takes the known factors such as the range of the humidity and precipitation from training data set to predict whether the baseball game will be played or not.

**Table 2.1** Training data set for the baseball game example

Temperature	Humidity	Rainy	Play (+) / Not Play (-)
85	85	true	-
80	90	false	-
83	78	false	+
70	96	false	-
68	80	true	-
65	70	true	-
64	65	false	+
72	95	false	-
100	70	false	-
75	80	false	+
102	70	false	-

Figure 2.1 the output of the decision tree for the baseball game example



The C4.5 program generated two types of trees. One is called an unpruned tree and the other is called a pruned tree. Unpruned trees are raw trees which contains redundant information. All trees generated in the process are saved in `filestem.unpruned`. When the expected error rate in the subtree is greater than in the single leaf and the tree is over fitted, pruning of the decision tree is done by replacing a whole subtree by a leaf node. The "best" pruned tree (selected by the program if there is more than one trial) is saved in machine-readable form in `filestem.tree`. All trees produced, both pre- and post-simplification, are evaluated on the training data. If required, they can also be evaluated on unseen data in the file `filestem.test`.

The C4.5 algorithm expects input in a tabular format where the last column contains the target (defined as positive or negative). All files read and written by C4.5 are of the form *filestem.ext* where *filestem* is a file name stem that identifies the induction task and *ext* is an extension that defines the type of file. The program expects to find three files: a names file `filestem.names` defining the columns of data file. In the above baseball game example, the `filestem.names` is `baseball.name`. The column headers such as Temperature, Humidity, and rain become the attribute names in `baseball.name`. Data file `filestem.data` contains a set of training data such as `baseball.data`, and a test file `filestem.test` contains a set of testing data such as `baseball.test`, each of which is described by its values of each of the attributes and its class.

## 2.5 *Sliding Window Algorithm*

For some protein superfamilies, such as Trx-fold proteins and G protein-coupled receptors (GPCRs), protein sequences are generally conserved, but interfamily similarities is low and typically not distinguished from noise by available sequence analysis tools, such as profile hidden Markov models built on the primary sequences. Thus, similarity analyses are not sufficient to identify new families of these proteins. Indeed, most of the new families were defined on the basis of experimentally determined structures of proteins that are representative members of the known families. Using the HMMER to profile primary structure sequences of Trx-fold proteins did not result in identifying new families of Trx-fold proteins satisfactorily [8]. Ji built a training set of 9 very distantly related Trx-fold and used this data set in a primary sequence-based jack-knife test with HMMER. In this test, he built an HMM on 8 of the 9 positive sequences and tested this model on the last (held-out) sequence and a separate set of negative sequences. This was repeated for each positive sequence separately. In this test, HMMER could not identify any of the Trx-fold proteins, demonstrating that primary sequence-based modeling methods are inappropriate for this problem. In this study, we combined two algorithms to recognize proteins with low primary sequence conservation. The algorithms we used are quasi-periodic feature classifier (QFC) and C4.5 decision tree prediction.

The architecture of the QFC algorithm has been described previously by Kim et al. [10]. With the QFC algorithm, the physical-chemical properties of the amino acids in the molecules were statically characterized using various indices and standard measurements,

such as GES hydrophathy index [24, 25], solubility [26], polarity, Kyte-Doolittle index [27], and molecular weight. A protein sequence is described by a set of variables  $x_1 - x_n$ , and for each  $x_i$ , there is a value  $x_{ij}$  for the  $i$ th amino acid index value at the  $j$ th position.  $x_{i1}-x_{ik}$  constitutes a profile of the protein in terms of the  $i$ th amino-acid property index. Since the raw profile  $x_{i1}-x_{ik}$  is a very noisy characterization of the protein, Sliding Window Recognizer was used to reduce noise of data [28]. The mechanism of the Sliding Window is to use kernel window function to recognize features at the proper scale. Table 2.5.1 and 2.5.2 demonstrate the procedures of the Sliding Window algorithm.

**Table 2.5.1** A partial protein sequence and its mapped value.

Position	1	2	3	4	5	6
Sequence	S	N	I	H	K	C
Polarity Value	9.2	11.6	5.2	10.4	11.3	5.5

Here we use the first 4 amino acids (window size 4) to explain how  $x_{i1}-x_{ik}$  constitutes a profile of the protein in terms of the  $i$ th amino-acid property index such as polarity.

In table 2.5.1, the polarity value 9.2 for S, 11.6 for N, 5.2 for I, and 10.4 for H. If we set window size at 4, the sum of the sequence SNIH is as follows:

$$9.2 + 11.6 + 5.2 + 10.4 = 36.4$$

We then use 3 Guassain kernels 0.00132952, 0.004878402, 0.014655531 to scale the value from moving window. The output is calculated as follows:

$$36.4 * 0.00132952 + 36.4 * 0.004878402 + 36.4 * 0.014655531$$

After getting the profile of protein sequences, we calculated average periodicity. Periodicity means how frequently the sliding window profile crosses over a neutral value. We then count how many times the profile changes sign around the neutral value and divide the sum by the length of sequence.

The protein profile and the average periodicity values are mapped to attributes in C4.5.

## ***2.6 Experiment Design Approach***

To prepare our data, we followed a procedure similar to the Sliding Window method used by Kim, et al [10]. We computed moving window profiles of known protein sequences. The Gaussian Kernel method with window size of 16 was used to smooth and to reduce noise in the data. We examined amino acid properties such as frequency, GES hydrophathy, KD, polarity, pI, alpha helix, molecular weight, solubility as we described in section 2.4. After many tests, we found that C4.5 decision trees performed better with the following five properties: molecular weight, polarity, GES hydrophobicity, solubility, and amino acid frequency count. We tested cutoff points for each of the five index from 1 cutoff points to 5 cutoff points and settled with 5 cutoff points for each of the properties.

The protein profile derived from the QFC method is used as the set of attributes in the C4.5 program to classify Trx-fold proteins and non-Trx proteins. All the records have the same structure, consisting of a number of attributes/values pairs of protein sequences derived from the QFC analysis. The attributes in the C4.5 program are molecular weight, polarity, GES hydrophobicity, solubility, and amino acid frequency count

For our experiments from 1 to 5, Trx-fold proteins in the training data set are combined with non-Trx-fold proteins to evaluate the data in the test data set, which also contains both Trx-fold proteins and non-Trx-fold proteins. For experiment 6 (test on search engine), the 9 and 12 Trx-fold proteins combined with known non-Trx-fold proteins were used as training data sets to build decision trees respectively. These built decision trees were used to search database to identify Trx-fold proteins.

The thioredoxin-fold protein search engine was designed to find new thioredoxin-fold proteins. This program allows users to search any database in FASTA format, e.g. PDB, *Methanococcus jannaschii* database, and *Campylobacter jejuni* database. It formats the names of amino acids to the standard form of the program. Then it uses moving window method to characterize amino acids physico-chemical properties by using molecular weight, polarity, solubility, hydrophobicity, amino acids frequency counts. The intermediate data are tested by C4.5 decision trees which are trained on known thioredoxin protein sequences and non-thioredoxin protein sequences. After positively identifying thioredoxin proteins, the names of these newly discovered proteins are recovered from the database.

## Chapter 3 Experiment Design and Data Analysis

In this project, we designed and performed five experiments to approach the problem of identifying Trx-fold proteins. The five experiments are listed below.

Experiment 1: Testing on data randomly selected from PDB database

Experiment 2: Jack-knife tests on 47 Trx-fold proteins

Experiment 3: Testing the effect of similarities

Experiment 4: Jack-knife tests on 9 Trx-fold proteins

Experiment 5: HMMER test to find 12 distantly related Trx-fold proteins and Jack-knife tests on the 12 Trx-fold proteins.

Experiment 6: Search *Methanococcus jannaschii* database, and *Campylobacter jejuni* database to find Trx-fold proteins

We obtained our positive training data set both from Dr. Dmitri Formenko of Dr. Vadim Gladyshev's lab in Department of Biochemistry (University of Nebraska-Lincoln) and from the PDB database ([www.rcsb.org](http://www.rcsb.org)). The negative data set used in our experiments is selected from the data set provided by Dr. Formenko. Both positive and negative training data sets are in a FASTA format file. In FASTA format, the first line is preceded by a ">" followed by the name of the sequence, such as TRIOREDOXIN, and the remaining lines contain amino acid sequences, the primary structure of the protein.

### ***3.1 Experiment 1: Testing on data randomly selected from PDB database***

In order to find out whether QFC and C4.5 decision tree algorithms were capable of identifying Trx-fold proteins when related proteins are potentially in the training set, we randomly chose 166 Trx-fold proteins and 250 non-Trx-fold proteins to do preliminary test on our model. 30 and 50 Trx-fold proteins were randomly selected and used as positive data in test1 and test2 respectively. In Test 1, 30 Trx-fold proteins combined with 50 negatives were used as test data set. The remaining 136 Trx-fold proteins combined with 200 negatives were used as training data set. In Test 2, 50 Trx-fold proteins combined with 80 negatives were used as test data set. The remaining 126 Trx-fold proteins combined with 170 negatives were used as training data set.

### ***3.2 Experiment 2: Jack-knife tests on 47 Trx-fold proteins***

Since many Trx-fold proteins contain CxxC motif, we decided to only use this type of proteins as positive sequences for our current experiments. We used a C program which only chooses sequences with CxxC motif and with keywords like thioredoxin or glutaredoxin to search the PDB database and obtained 47 Trx-fold proteins with CxxC motif. The 47 protein sequences are separated into 17 groups by sequence similarities and were used as Jack-knife with our program, i.e. we removed from the file all sequences in one of the 17 groups, built our classifier on the sequences from remaining 16 groups, and tested our classifier on the held out group. Each testing set contains one

group of positive Trx fold protein sequences and 42 negative non-Trx fold proteins. Each of 16 groups was combined with 60 negatives as training data set to build decision tree to evaluate the remaining testing data set.

### ***3.3 Experiment 3: Testing the effect of similarities***

Our next test was to determine our program's sensitivity to varying levels of sequence similarity in the data set. From the set of Trx-fold proteins provided by Dr. Gladyshev's lab, we selected 168 sequences with CxxC motifs with amino acids in length between 200 and 500 and divided them into 3 sets. The first set of sequences was filtered such that no two sequences were more than 90% similar when pairwise aligned. The second set of sequences was filtered such that no two sequences were more than 80% similar when pairwise aligned. The third set of sequences was filtered such that no two sequences were more than 70% similar when pairwise aligned. Within each set, Trx-fold proteins were divided into three sets labeled as A, B, C, respectively. When set A was used as test set, set B and set C were combined as training data set, likewise for set B and set C. 200 negatives were used in testing data sets and 400 negatives were used in training data sets.

### ***3.4 Experiment 4: Jack-knife tests on 9 Trx-fold proteins***

Our next experiment was designed to measure how well our program is in finding new, highly dissimilar Trx-fold proteins. We filtered our set of proteins via the following procedure. First we used the alignment tool ClustalW [29] to align those proteins.

ClustalW takes a file with multiple sequences as the input, and outputs the aligned file. After aligning them, a dendrogram tree is also generated with the level of similarities between each pair of proteins listed. The closely related proteins have short lines to connect them together and the distantly related proteins have long lines to connect them or no lines to connect them together. We selected 160 negatives from the negative data set obtained from Vadim Gladyshev's lab. We chose 9 positives out of the 47 Trx-fold proteins which have the minimum level of similarities between each pair of them.

These 9 positive protein sequences and 120 negative proteins sequences are used for the Jack-knife tests with C4.5 decision program. For the 9 positive Trx-fold proteins, 8 positives are used as training and one remaining is held out as testing data. We divide 120 negatives into 10 training groups. We trained the 8 positives with each of the 10 negative groups to build 10 decision trees. The held out positive was evaluated by each of the 10 decision trees.

### ***3.5 Experiment 5: HMMER test to find 12 distant related Trx-fold proteins and Jack-knife tests on the 12 Trx-fold proteins***

We also built an additional set of Trx-fold proteins to experiment with the method similar to those in section 3.4. We selected 12 positives out of 200 Trx-fold proteins obtained from the PDB database. When used in a jack-knife test with HMMER trained on primary sequence, HMMER again failed to identify any of the 12. This demonstrates the high diversity of the data set.

These 12 positive protein sequences and 140 negative proteins sequences are used for the Jack-knife tests with C4.5 decision tree program. For the 12 positive Trx-fold proteins, 11 positives were used as training and the remaining one was tested. 140 negatives were divided into 13 groups. We trained the 12 positives with each of the 12 negative groups to build 12 decision trees. The held out positive was evaluated by each of the 12 decision trees.

### ***3.6 Experiment 6: Search database to find Trx-fold proteins***

We built a thioredoxin search engine based on the two sets of known Trx-fold proteins used in section 3.4 and 3.5 and a set of known non-Trx-fold proteins. After running Sliding Window procedure, the statistic values of the positive and negative proteins are used as training data to build the C4.5 decision tree which is used to search *Methanococcus jannaschii* database to find Trx-fold proteins.

## Chapter 4 Experimental Results and Discussion

### *4.1 Results on Experiment 1: Testing on data randomly selected from PDB database*

In this experiment, 166 Trx-fold proteins obtained from PDB database search were randomly divided into two subsets. The first subset contained 30 Trx-fold proteins and the second subset contained 50 Trx-fold proteins. In Test 1, the set of 30 Trx-fold proteins was combined with 50 negative (non-Trx-fold proteins) to form the training set. In Test 2, the set of 50 Trx-fold proteins was combined with 80 negative Trx-fold proteins to form the testing set. The training data set for Test 1 contained 136 positive and 200 negative sequences. The training data set for Test 2 contained 126 positives and 170 negatives. Table 4.1 is constructed with the confusion matrix of the testing results with pruned trees. The second row represents the actual positive or negative. The first column represents the label of predicted value.

**Table 4.1** Results of identifying Trx-fold protein by using Sliding Window method and C4.5 decision tree.

	TEST 1		TEST 2	
	Positive	Negative	Positive	Negative
Positive	30	1	45	0
Negative	0	49	5	80
Accuracy	100%	98%	90%	100%

From Table 4.1, we can see the results of QFC and C4.5 on identifying Trx-fold proteins. For test 1, 30 out of 30 positives and 49 out of 50 negatives were identified. For test 2, 45 out of 50 positives and 80 out of 80 negative were identified. The false positive and false negative rates are below 10% for both test.

From the result, we found that our model based on protein properties had potential to successfully separate Trx-fold proteins from non-Trx-fold proteins. These results suggest that Trx-fold proteins have physico-chemical characteristics that can be identified by C4.5. This assumption and our preliminary test results encouraged us to do further tests.

## ***4.2 Results on Experiment 2: Jack-knife tests on 47 Trx-fold proteins***

In this experiment, 47 Trx-fold proteins were separated into 17 families by the method in section 3.2. After running QFC procedure, C4.5 program were used to perform Jack-knife test with the 17 Trx-fold protein families. Each testing set contained one Trx-fold protein family and the same 42 non-Trx-fold proteins. Each training set contained the remaining 16 Trx-fold protein families combined with the same 60 negatives.

**Table 4.2** Results of jack-knife test on 47 Trx-fold proteins.

Group Names	Positive identified	Accuracy (%)	Negative identified	Accuracy (%)
1	1/1	100%	41	97%
2	2/2	100%	41	97%
3	1/1	100%	41	97%
4	3/3	100%	41	97%
5	7/7	100%	36	88%
6	1/1	100%	41	97%
7	1/1	100%	41	97%
8	1/1	100%	41	97%
9	1/1	100%	41	97%
10	3/3	100%	41	97%
11	4/4	100%	41	97%
12	11/11	100%	41	97%
13	1/1	100%	41	97%
14	1/1	100%	41	97%
15	2/2	100%	41	97%
16	1/1	100%	41	97%
17	0/5	0%	39	95%

Table 4.2 shows the QFC and C4.5 decision tree model can identify homology families by using selected negative data set as training data set. From the Table, we can see that 16 out of the 17 Trx-fold proteins families are successfully identified. None of the 5 Trx-fold proteins in group 17 are identified. It is possible that these 5 proteins are homologous and their physical-chemical characteristics derived from the QFC are different from the other groups. Therefore, the C4.5 decision tree is unable to separate them from other proteins.

### ***4.3 Results on Experiment 3: Testing the effect of homology***

In this experiment, we separated Trx-fold proteins into three data groups. The three groups contained proteins that are at most 90% pairwise identical, 80% pairwise identical, and 70% pairwise identical respectively. Within each group, Trx-fold proteins were divided into three sets labeled as A, B, C, respectively. When set A was used as test set, set B and set C were combined as training data set, likewise for set B and set C. The number of positive and negative proteins used in the testing and training data sets is indicated in the following Table.

**Table 4.3** Results of identifying Trx-fold proteins with different similarities as positive training data sets and tests data set by using Sliding Window method and C4.5 decision tree.

Group Names	Sequence Identical (%)	Positive test set	Negative test set	Positive training set	Negative training set	Positive identified	Negative identified
A	90	59	61	126	113	54	53
B	90	63	55	122	119	57	51
C	90	63	58	122	116	57	54
<i>Average</i>						<b>90%</b>	<b>91%</b>
A	80	60	66	128	126	51	56
B	80	61	66	127	126	48	53
C	80	67	60	121	132	53	54
<i>Average</i>						<b>81%</b>	<b>85%</b>
A	70	49	61	98	121	33	52
B	70	49	61	98	121	32	51
C	70	49	60	98	122	33	39
<i>Average</i>						<b>67%</b>	<b>78%</b>

Table 4.3 shows that the similarities of the protein sequences have great effect at QFC and C4.5 model. With similarities at 90% pairwise identity, the false positive and false negative rates are both below 10%. As the similarities of proteins decreases, the false positive and false negative rates increase. With similarities at 80% pairwise identity, we can identify Trx-fold proteins with 81% accuracy. When the similarities further

decreases to 70% pairwise identity, only 67% positives and 78% negatives are identified. This is probably due to that eliminating of similarities will eliminate the bias of C4.5 decision trees. This suggests when sequences are similar, we can use c4.5 to identify Trx-fold proteins with high accuracy, but as similarity of sequences decrease, C4.5's ability to identify Trx-fold proteins also decreases.

#### ***4.4 Results on Experiment 4: Jack-knife tests on 9 Trx-fold proteins***

In this experiment, 9 dissimilar Trx-fold proteins were used as positives. Each positive sequence was combined with 12 negative sequences to be used as a testing data set. The remaining 8 positives were combined with each of the 10 negative data groups and used as training sets. Therefore, each testing set was evaluated 10 times with trees generated by 10 training data sets respectively. The reason for using 10 negative sets was to check sensitivity of C4.5 to choice of negative set.

Column 1 in Table 4.4 is the names of the 9 Trx-fold proteins. Column 4 represents how many times the tested positive protein is identified after being evaluated by 10 training data sets separately.

**Table 4.4** Results of Jack-knife tests on 9 Trx-fold proteins.

Names of Positive Protein	e-value	Times being identified (out of 10 tests)	Accuracy Percentage
>1A8L	0.83	5	50%
>1EEJ	2.9	5	50%
> 1BED	5.8	5	50%
>1QK8	>10	6	60%
>1F9M	6.2	8	80%
>1MEK	0.62	5	50%
>1EGO	6.7	3	30%
>1FOV	>10	4	40%
>1DE1	>10	6	60%

From table 4.4 we can see that Jack-knife tests can identify 7 out of the 9 Trx-fold proteins were identified at least half times of the 10 tests. 2 out of 9 Trx-fold proteins were identified less than half of the 10 test. This indicates that C4.5 does seem to be sensitive to the choice of negative set. Comparing this result with the results of HMMER trained on primary structure conducted by Haifeng Ji [8], we find that QFC and C4.5 perform better. For HMMER based on primary structures, none of the 9 positive proteins was identified in the jack-knife test. HMMER trained on primary structure could not find Trx-fold proteins with low level of similarities. Our results indicate that there is a possibility for C4.5 to identify remote Trx-fold proteins.

#### ***4.5 Results on Experiment 5:HMMER test to find 12 distantly related Trx-fold proteins and Jack-knife tests on the 12 Trx-fold proteins***

We used Jack-knife test on HMMER to select a new set of Trx-fold proteins with low similarity in their sequences by using the method in section 3.4. We did Jack-knife test on these 12 Trx-proteins using our model. Each testing set contained one positive Trx-fold protein sequence and 12 negative non-Trx fold proteins. Each positive sequence was tested with the remaining 11 positive combined with each of 12 negative data groups to training sets. Therefore, each testing data set was evaluated 12 times with trees generated by 12 training data sets respectively.

Column 1 in table 4.5 is the names of the 12 Trx-fold proteins. Column 2 is the e-value from HMMER tests. Column 3 represents how many times the tested positive protein is identified after being evaluated by 12 training data sets with C4.5 program. Column 4 is the percentage of accuracy for Jack-knife test with C4.5 program.

**Table 4.5** Results of HMMER trained on primary structure and C4.5 Jack-knife

tests on 12 Trx-fold proteins.

Names of Positive Protein	Primary e-value	Times being identified (out of 12 tests)	Accuracy Percentage
>1EEJ	3.8	0	0%
> 1BED	>10	7	58%
>1QK8	>10	10	83%
>1F9M	3.3	4	33%
>1EGO	4.3	9	75%
>1FOV	>10	3	25%
>1DE1	>10	7	58%
>gi:9989039	>10	6	50%
>gi:12324654	>10	8	67%
>gi:15610809	0.33	4	33%
>gi:1729945	>10	6	50%
>gi:7109697	>10	11	91%

The above table shows 8 out of the 12 Trx-proteins can be identified with more than 50% accuracy with one having 91% accuracy. The results of Jack-knife tests on this 12 positive Trx-fold proteins reinforced our assumption that the QFC and C4.5 algorithm can do better than HMMER trained on primary sequence.

## ***4.6 Results on Experiment 6: Search database to find Trx-fold proteins***

We used 9 and 12 Trx-fold proteins obtained by using methods in sections 3.4 and 3.5. We selected negative sets that performed well in sections 4.4 and 4.5 experiments. The 9 Trx-fold proteins were combined with 50 non-Trx-fold proteins to form one training set and the 12 Trx-fold proteins were combined with 50 non-Trx-fold proteins to form the other training set to build two decision trees with C4.5 program. Both trees were used to search *Methanococcus jannaschii* database to find Trx-fold proteins.

Table 4.6.1 list of the Trx-fold proteins found from *Methanococcus jannaschii* **database**.

<b>gi:1591289</b>	<b>thioredoxin</b>
<b>gi:2826305</b>	<b>methyltransferase</b>
<b>gi:1591909</b>	<b>Predicted methyltransferase</b>
<b>gi:1591150</b>	<b>Thiamine biosynthesis enzyme</b>
<b>gi:1590945</b>	<b>hypothetical protein</b>
<b>gi:1592229</b>	<b>coenzyme PQQ synthesis protein</b>
<b>gi:1498830</b>	<b>hypothetical protein</b>

Table 4.6.2 list of the Trx-fold proteins found from *Campylobacter jejuni* database

CAB72510.1	DsbV
CAB72631.1	thioredoxin
CAB75239.1	Thiol:disulfide interchange protein Trx domain

There are 7 Trx-fold proteins found in the *Methanococcus jannaschii* database and 3 Trx-fold proteins found in *Campylobacter jejuni* database. The search results show that our search engine has the potential of finding many new Trx-fold proteins from protein database.

## Chapter 5 Conclusions and Future Work

This study provides important insights into the role of C4.5 decision tree as a machine learning tool to identify thioredoxin-fold proteins based on the physico-chemical characteristics of amino acid sequences. A number of earlier studies had shown that hidden Markov models trained on primary structures cannot identify new Trx-fold proteins effectively [8]. The low in amino acid conservation among thioredoxin-fold proteins makes use of conventional research techniques difficult to identify new members of this protein superfamily.

In this study, we developed a model to identify Trx-fold protein superfamilies with low primary structure conservation. We used QFC method [10] to obtain the characteristics of the protein sequences, including molecular weight, polarity, GES hydrophobicity, and amino acids frequency counts. We then used C4.5 decision tree to separate Trx-fold proteins from non-Trx-fold proteins. In order to test preliminarily the performance of our model (see experiment 1), we randomly selected positive and negative data sets in our early experiments. The initial results showed that we could identify Trx-fold protein with above 90% accuracy, which encouraged us to do further evaluation on our model by using different training data sets and testing data sets. We applied our model to test protein sequences with different homologous. We trained and tested proteins with homology at least 90%, 80%, and 70% pairwise identical sequences respectively (see experiment 3). We discovered as the homology of proteins decreases, the percentage of accuracy in prediction decreases too. With homology at 90% pairwise

identical, the false positive and false negative rates are below 10%. As the similarity of proteins decreases, the false positive and false negative rates increase. With similarities at most 80% pairwise identity, we can identify Trx-fold proteins with 81% accuracy. When the similarity decreases to 70%, only 67% positives and 78% negatives are identified. This suggests that it is possible when the training data set contains protein sequences with high similarities, the proteins within the closely related family are easily identified, and when the training data set contains proteins sequences with low similarities, the bias of the training data is eliminated.

In order to compare our model with HMMER trained on primary structure, we did Jack-knife test on 47 Trx-fold proteins and 9 Trx-fold proteins, which have minimum level of similarity. We separated the 47 Trx-fold proteins into 17 families by using similarity. We successfully identified 16 Trx-fold protein families out of the 17 families (see experiment 2). For the 9 Trx-fold proteins, we evaluated each of the positive with 10 different training data sets. We identified 7 positive Trx-fold proteins with 50% or higher accuracy (see experiment 4). This result shows that our model preformed better than that of HMMER trained on primary structure, because the later can not identify any Trx-fold proteins with low level of similarity [8].

We also ran HMMER test to select additional 12 distantly related Trx-fold proteins with minimum level of similarities and ran Jack-knife test on the 12 Trx-fold proteins. In order to eliminate the bias in our training data set, we separated the negative training data set into 12 groups and tested each of the Trx-fold family with each of the 12 negatives data groups as training data set and averaged the testing results. We found that 8 out of

the 12 Trx-protein can be identified with more than 50% accuracy with one having 91% accuracy. This result further proved our discovery that C4.5 decision tree is more efficient in identifying Trx-fold proteins than HMM on primary structure.

We also built a search engine to find Trx-fold proteins from *Methanococcus jannaschii* database. We found 7 Trx-fold proteins from *Methanococcus jannaschii* database. This indicates that our tool has the potential of finding many new Trx-fold proteins from protein databases.

Since we used distinct positive Trx-fold proteins and low homology negative proteins for our training data set, we may have low hit rate and relatively high false positive in identifying Trx-fold proteins from a large database. In the future, we will develop and enhance search tools to identify Trx-fold proteins with higher accuracy on both high similarity and low similarity protein sequences

We will continue our database search to look for Trx-fold proteins in more databases. 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 functional classifications of discovered proteins.

## References

1. Spyrou G, Padilla, CA, Holmgren A, and Miranda-Vizueté A. (2001). A genome-wide survey of human thioredoxin and glutaredoxin family pseudogenes. *Human Genetics* 109, 429-439.
2. Martin J. (1995). Thioredox – a fold for all reasons. *Structure*. 3:245-250
3. Wakasugi N, Tagaya Y, Wakasugi H, Mitsui A, Maeda M, et al. (1990). Adult T-cell leukemia-derived factor/thioredoxin, produced by both human T-lymphotropic virus type I- and Epstein-Barr virus-transformed lymphocytes, acts as an autocrine growth factor and synergizes with interleukin 1 and interleukin 2. *Proc. Natl. Acad. Sci. USA*. 87: 8282-86
4. Nakamura H, Masutani H, Tagaya Y, Yamauchi A, Inamoto T, et al. (1992). Expression and growth-promoting effect of adult T-cell leukemia-derived factor. A human thioredoxin homologue in hepatocellular carcinoma. *Cancer* 69: 2091-97
5. Debarbieux L, Beckwith J. (1999). Electron avenue: pathways of disulfide bond formation and isomerization. *Cell* 99: 117-119
6. Aslund F, Beckwith J. (1999). The thioredoxin superfamily: redundancy, specificity, and gray-area genomics. *J. Bacteriol* 181: 1375-1379
7. Dmitriy F, Patrick A. (1997). Seventy-five percent accuracy in protein secondary structure prediction. *Proteins: Structure, Function, and Genetics* 27:329-335.
8. Haifeng J. (2002). On Modeling protein superfamilies with low primary sequence conservation. Master Thesis. University of Nebraska.

9. Fomenko D., Gladyshev V. (2002). CxxS: fold-independent redox motif revealed by genome-wide searches for thio/disulfide oxidoreductase function. *Protein Science*. 11: 2285-2296.
10. Kim J., Moriyama E., et al. (2000). Identification of novel multi-transmembrane proteins from genomic databases using quasi-periodic structural properties. *Bioinformatics*. 16: 767-775.
11. <http://www.cse.unsw.edu.au/~quinlan/>
12. Branden V and Tooze J. (1998). Introduction to Protein Structure (2<sup>nd</sup> Edition), Pub., Taylor & Francis inc.
13. Salzberg S. L., Searls D.B., Kasif S. (Eds.). (1998). Computational Methods in Molecular Biology. Elsevier Science B.V.
14. Buchanan BB, Schurmann P, Decottignies P, Lozano RM. (1994). Thioredoxin: a multifunctional regulatory protein with a bright future in technology and medicine. *Arch Biochem Biophys*. 314:257-260.
15. Holmgren, A. (1989). *Journal of Biochemistry*. 264, 13963-13966
16. Eklund, H, Gleason, F.K, Holmgren A. (1991). *Proteins: Structure, Function, and Genetics* 11, 13-28.
17. Follmann H, Haberlein I. (1995). Thioredoxins: universal, yet specific thiol-disulfide redox cofactors. *Biofactors*. 5:147-156.
18. Martin J. (1995). Thioredoxin – a fold for all reasons. *Structure* 3:245-250
19. Holmgren, A. 1989. Thioredoxin and glutaredoxin systems. Minireview. *Journal of Biochemistry*. 264, 13963-13966.
20. Eklund H, Gleason F.K, Holmgren A. (1991). Structural and functional relations

- among thioredoxins of different species. *Proteins*. 11:13-18.
21. Hollmann H, Haberlein I. (1995). Thioredoxins: universal, yet specific thiol-disulfide redox cofactors. *BioFactors*. 5:147-156.
  22. Salzberg, S. L., Searls D.B., Kasif S. (Eds.). (1998). Computational Methods in Molecular Biology. Elsevier Science B.V.
  23. Baldi P. and Soren B. (1998). Bioinformatics: A Machine Learning Approach. MIT press, Boston.
  24. Engelman D.M., Steitz, T.A. and Goldman A. (1986). Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Annual Review of biophysics and Biophysical Chemistry*. 15, 321-353.
  25. Von hajjne G. (1992). Membrane protein structure prediction. Hydrophobicity analysis and the positive-inside rule. *Journal of Molecular Biology*. 225, 487-494.
  26. Brown T. (1998). Molecular Biology labfax 2<sup>nd</sup> edition. Academic Press, San Diego.
  27. Kyte, J. and Doolittle, R.F. 1982. A simple method for displaying the hydrophobic character of a protein. *Journal of Molecular Biology*. 157, 105-132
  28. Tukey J.L. (1977). Exploratory data analysis. Addison-Wealey, Reading, MA.
  29. <http://www.ebi.ac.uk/clustalw/>

## Appendix A: Positive data set used for jack-knife test

>1EEJ:A THIOL:DISULFIDE INTERCHANGE PROTEIN

DDAAIQQTLAKMGIKSSDIQPAPVAGMKTVLTNSGVLYITDDGKHIIQGPMYDVS  
GTAPVNVNKNMLLKQLNALEKEMIVYKAPQEKHVITVFTDITCGYCHKLHEQM  
ADYNALGITVRYLAFPRQGLSDAEKEMKAIWCAKDKNKAFFDDVMAGKSVAP  
ASCDVDIADHYALGVQLGVSGLTPAVVLSNGTLVPGYQPPKEMKEFLDEHQKMT  
SGK

>1BED:\_ DSBA OXIDOREDUCTASE

AQFKEGEHYQVLKTPASSSPVSEFFSFYCPHCNTFEPHIAQLKQQLPEGAKFQKN  
HVSFMGGNMGQAMSKAYATMIALEVEDKMVPVMFNRIHTLRKPPKDEQELRQI  
FLDEGIDAAKFDAAAYNGFAVDSMVRREFDKQFQDSGLTGVPVAVVNNRYLVQG  
QSVKSLDEYFDLVNYLLTLK

>1QK8:A TRYPAREDOXIN-I

MSGLDKYLPGIEKLRRGDGEVEVKSLAGKLVFFYFSASWCPPCRGFTPQLIEFYD  
KFHESKNFEVVFCTWDEEEDGFAGYFAKMPWLAVPFAQSEAVQKLSKHFNVESI  
PTLIGVDADSGDVVTTRARATLVKDPEGEQFPWKDAP

>1F9M:A THIOREDOXIN F

MEAIVGKVTEVNKDTFWPIVKAAGDKPVVLDMFTQWCGPCKAMAPKYEKLAE  
EYLDVIFLKLDCNQENKTLAKELGIRVVPTFKILKENSVVGEVTGAKYDKLLEAI  
QAARS

>1EGO:\_ GLUTAREDOXIN (OXIDIZED) (NMR, 20 STRUCTURES) - CHAIN \_

MQTVIFGRSGCPYCVRAKDLAEKLSNERDDFQYQYVDIRAEGITKEDLQQKAGK  
PVETVPQIFVDQQHIGGYTDFAAWVKENLDA

>1FOV:A GLUTAREDOXIN

ANVEIYTKETCPYCHRAKALLSSKGVSFQELPIDGNAAKREEMIKRSGRTTVPQIF  
IDAQHIGGYDDLIALDARGGLDPLLK

>1DE1:A GLUTAREDOXIN

MFKVYGYDSNIHKCVYCDNAKRLTLVKKQPFEFINIMPEKGVFDDEKIAELLTKL  
GRDTQIGLTMPQVFAPDGSHIGGFDQLREYFK

>gi|9989039|gb|AAG10802.1|AC022284\_7 (AC022284) Thioredoxin-like

protein [Leishmania major]

MLKVSSKEHYAEIKKKAEDSLGLVVHFSATWCEPCTAVNEHLTKQAAEYGDNV  
VFAEVDCGELGDVCEAEGVESVPFVAYFRTPLVGDDRRVERVADVAGAKFDQI  
DMNTHSLFGEKGGNRGSAEGLCHSGRLPALPHEAARGRNVHHRHPISSALRLY  
WSAV

>gi|12324654|gb|AAG52290.1|AC019018\_27 (AC019018) putative

thioredoxin; 109829-109566 [Arabidopsis thaliana]

MFPVMVMFTARWCGPCRDMIPILNKMDSEYKNEFKFYTVNFDTEIRFTERFDISY  
LPTTLVFKGGEQMAKVTGADPKKLRELVKKYI

>gi|15610809|ref|NP\_218190.1| (NC\_000962) hypothetical protein Rv3673c

[Mycobacterium tuberculosis H37Rv]^Agi|15843290|ref| NP\_338327.1| (NC\_002755)

thioredoxin-related protein [Mycobacterium tuberculosis CDC1551] ^Agi|7477713|pir||

B70790 hypothetical protein Rv3673c - Mycobacterium tuberculosis (strain H37RV)

^Agi|2960097|emb|CAA17995.1| (AL022121) hypothetical protein Rv3673c

[Mycobacterium tuberculosis H37Rv]^Agi|13883649|gb|AAK48141.1| (AE007175)

thioredoxin-related protein [Mycobacterium tuberculosis CDC1551]

MPSLPTTPAETAMTTLTGKTRWTIAILAVVAALMAALVAQLHDYSASSTISQRPA  
PREHRDGDTPPEALAWSRQRANLPPCPAAGNGPGAAALRGVVVVCAGDGSAVD  
VARALAGR RVINLWAHWCAPCMTELPVMAEYQRRVGPVLLVTVHQGQNE  
AAALSRLADLGVRLPTLQDDRRRVAAALRVANVMPATVVLRPDGSVAQTLPRA  
FGSADEIVAAVGNDAG

>gi|1729945|sp|P21610|THIO\_EUBAC Thioredoxin (TRX)^Agi|629193|pir

||S38989 thioredoxin - Eubacterium acidaminophilum^Agi|388295|gb|

AAB93304.1| (L04500) thioredoxin [Eubacterium acidaminophilum]

MSALLVEIDKDQFQAEVLEAEGYVLVDYFSDGCVPCCKALMPDVEELAACYEGK  
VAFRKFNTSSARRLAISQKILGLPTITLYKGGQKVEEVTKDDATRENIDAMIAKH  
VG

>gi|7109697|gb|AAF36768.1| (L35043) thioredoxin [Mycoplasma gallisepticum]

MKHITNKAELDQLLTTNKKVVVDFYANWCGPCKILGPIFEEVAQDKKDWTFFVK  
VDVDQANEISSEYEIRSIPTIIFFDGK MADKRIGFIPKNELKELLK