University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

CSE Technical reports | Computer Science and Engineering, Department of

1-1-2003

On Modeling Protein Superfamilies with Low Primary Sequence Conservation

Stephen Scott
University of Nebraska - Lincoln, sscott2@unl.edu

H. Ji
University of Nebraska - Lincoln

P. Wen
University of Nebraska - Lincoln

Dmitri E. Fomenko
University of Nebraska - Lincoln, dfomenko2@unl.edu

Vadim N. Gladyshev
University of Nebraska-Lincoln, vgladyshev1@unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/csetechreports

Part of the Computer Sciences Commons

Scott, Stephen; Ji, H.; Wen, P.; Fomenko, Dmitri E.; and Gladyshev, Vadim N., "On Modeling Protein Superfamilies with Low Primary Sequence Conservation" (2003). *CSE Technical reports.* Paper 43.
http://digitalcommons.unl.edu/csetechreports/43

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Technical reports by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

University of Nebraska-Lincoln, Computer Science and Engineering
Technical Report # TR-UNL-CSE-2003-0004, 01/05/2003

On Modeling Protein Superfamilies with Low Primary Sequence Conservation

S. D. Scott*, H. Ji*, P. Wen*, D. E. Fomenko†, and V. N. Gladyshev†

January 5, 2003

Abstract

Motivation: *Development of tools for identification of new thioredoxin-fold proteins as well as other proteins belonging to superfamilies with low primary sequence conservation.*

Results: *We present several algorithms for identifying thioredoxin (Trx)-fold proteins containing a conserved CxxC motif (two cysteines separated by two residues). The low conservation of primary sequence in this protein superfamily makes conventional methods difficult to use. Therefore, we use structural properties to build our classifiers. These structural properties include secondary structure patterns as well as various properties of the residues in the protein sequences. We use this information to model Trx-fold proteins via hidden Markov models, decision trees, and algorithms in the multiple-instance learning model. In 9-fold and 12-fold jack-knife tests, some of our models performed quite well, with high true positive and true negative rates. In addition, By combining a small number of our classifiers, we can identify 100% of the Trx-fold proteins in these jack-knife tests with moderate false positive rates. We also identified several candidate Trx-fold proteins in the* C. jejuni, M. jannaschii, E. coli *and* S. cerevisiae *genomes. Since our techniques are very general, they should be applicable to other superfamilies with low primary sequence conservation.*

Availability: *C code available via email from contact author.*

Contact: *Stephen Scott, Dept. of Computer Science, 115 Ferguson Hall, University of Nebraska, Lincoln, NE 68588-0115, USA, sscott@cse.unl.edu, (402) 472-6994, fax: (402) 472-7767*

Keywords: *low primary sequence conservation, hidden Markov models, C4.5, multiple-instance learning, thioredoxin-fold proteins, redox proteins*

1 Introduction

Oxidation-reduction reactions in cells are catalyzed by various redox proteins, many of which use catalytic cysteine residues. Thiol-dependent redox proteins regulate many basic cellular processes, such as DNA synthesis, apoptosis, signal transduction and transcription [12, 5]. To understand the mechanism of cellular redox regulation, the first step is to identify redox proteins and to characterize the specific functions of these proteins [5, 2]. The thioredoxin superfamily is the major family of thiol-dependent oxidoreductases involved in cellular regulation, and its characterization is important for understanding of redox processes. In addition to thioredoxin, it includes protein disulfide isomerases, glutaredoxins, nucleoredoxins, peroxiredoxins, glutathione peroxidases and other redox enzymes.

Inter-family similarity within the Trx-fold superfamily is generally low, and sequence analysis tools such as HMMER cannot easily identify new families in the Trx-fold superfamily. For example, in Figure 1,

*Dept. of Computer Science & Engineering, University of Nebraska, 115 Ferguson Hall, Lincoln, NE, 68588-0115, USA, sscott@cse.unl.edu

†Dept. of Biochemistry, N151 Beadle Center, 1901 Vine St., Lincoln, NE, 68588-0664, USA

```
                    *   *
1A8L:   KLIVFVRKDHCQYCDQLKQLVQEL
1BED:   PVVSEFFSFYCPHCNTFEPIIAQL
1QK8:A  LVFFYFSASWCPPCRGFTPQLIEF
1F9M:A  PVVLDMFTQWCGPCKAMAPKYEKL
1MEK:   YLLVEFYAPWCGHCKALAPEYAKA
```

Figure 1: Alignment of segments of five Trx-fold proteins, indexed by PDB ID.

active site segments of five Trx-fold proteins are shown. Only the two cysteines (C, marked by asterisks) are conserved in the alignment. These two cysteines form a redox motif designated the CxxC motif. This motif is conserved in the majority members of the superfamily, including thioredoxins, glutaredoxins, protein disulfide isomerases and other proteins. However, some of the Trx-fold proteins conserve only a single cysteine (e.g. peroxiredoxins and glutathione peroxidases).

In a more rigorous evaluation of the low primary sequence conservation of this superfamily, we used HMMER to attempt to identify distinct Trx-fold protein families based on primary sequence alone (Section 3.2) by running jack-knife tests on sets of highly dissimilar sequences. In these tests, 0% of distinct Trx-fold proteins were identified, indicating that primary structure alone is insufficient in identification of Trx-fold protein families. In our study, we use structural properties to identify new Trx-fold protein families. These structural properties include secondary structure patterns, as well as various properties of the residues in the protein sequences. We use this information to model Trx-fold proteins via hidden Markov models, the decision tree learning algorithm C4.5 [17], and algorithms [10, 21] in the *multiple-instance learning model* [7]. Our strongest results came from applying hidden Markov models to predicted secondary structure patterns (predicted by PREDATOR [9]) and to reduced alphabets that captured hydrophobicities of the individual residues. In 9-fold and 12-fold jack-knife tests, the models based on predicted secondary structure achieved true positive rates above 0.75 and true negative rates above 0.85. Hydrophobicity-based models had much lower true positive rates, but they often identified the sequences that were missed by Predicted Secondary. Indeed, by combining Predicted Secondary with the hydrophobicity-based models (and in one case, a third model), we identified 100% of the Trx-fold proteins in both jack-knife tests with moderate false positive rates. We also identified several new Trx-fold proteins in the databases of *C. jejuni*, *M. jannaschii*, *E. coli*, and *S. cerevisiae*. Finally, since our techniques are very general, they should be applicable to other superfamilies with low primary sequence conservation beyond the Trx-fold superfamily.

In addition to the CxxC motif mentioned above, Trx-fold proteins conserve secondary structure, such that three α-helices and four β-sheets are organized in a specific pattern (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 [16, 13]. Therefore, even though the protein primary sequences are not conserved, one can use structural information as well as the CxxC motif to discriminate Trx-fold proteins. It should be noted, however, that some Trx-fold proteins allow insertions of secondary structures, which complicate the searches.

The rest of this paper is as follows. In Section 2 we describe the algorithms we employ in our study. Then in Section 3 we summarize our experimental results. Finally, we conclude in Section 4 with a discussion of future work.

2 Our Algorithms

We apply three fundamental approaches to this problem. The first employs hidden Markov models (HMMs), but the models are built on structural information rather than on primary sequence. The second approach involves deriving summary statistics on structural information on the sequences (similar to that used in the QFC algorithm [14]) and using these statistics as attributes to C4.5 [17], which is a robust algorithm for learning decision trees. In our third approach we treat this problem as a *multiple-instance* problem in machine learning [7] and apply two different algorithms [10, 21] to learn classifiers that will separate Trx-fold proteins containing the CxxC motif from non-Trx-fold proteins.

2.1 HMM-Based Models

Given the high conservation of secondary structure in the Trx-fold superfamily and a defined position of the CxxC motif within the secondary structure pattern, we could use these features to develop HHMs. We use known secondary structure when available, and in the other instances we predict secondary structure with PREDATOR [9]. These sequences of structural elements (one element per residue in the primary sequence) are aligned with ClustalW, and then the alignments are used to infer an HMM using HMMER[1]. After building and calibrating our model, we extract every sequence in the database that contains the CxxC motif in the primary structure, predict their secondary structures, and then score them with our model.

While predicted secondary structure is a natural first approach, PREDATOR (like other structure prediction algorithms) has a fairly high per-residue error rate. This introduces significant noise in remapped sequences and thus affects our model. Hence we also looked at other sequence mappings. Andorf et al. [1] and Wang et al. [20] remapped the 20-character amino acid alphabet to a reduced one that captures structural properties. 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. Their results raise the possibility that the use of different alphabets might provide different, but complementary, insights into protein structure-function relationships. So in addition to the remapping to secondary structure elements as outlined above, we remapped our sequences from the 20-character amino acid alphabet to a reduced one. We tried remappings based on hydrophobicity, charge, volume and mass. The details of the remappings are shown in Table 1. Each column shows a criterion for remapping and the class that the particular residue was remapped to based on that criterion. Then for each remapping, we aligned the remapped sequences with ClustalW, built a HMM with HMMER, and searched databases of remapped sequences for hits.

2.2 QFC/C4.5

In the QFC algorithm [14], the physical-chemical properties of the amino acids in the molecules are statically characterized using various indices and standard measurements, such as GES hydropathy index [8, 11], solubility [3], polarity, pI, Kyte-Doolittle index [15], α helix index [6], and molecular weight. A protein sequence is described by a set of variables x_1 through x_n, and for each x_i, there is a value x_{ij} for the ith amino acid index value at the jth position. Thus x_{i1} through x_{ik} constitutes a profile of the protein in terms of the ith amino-acid property index (see Figure 2). Then each raw profile is smoothed by applying the Sliding Window Recognizer [19], which transforms the profile as follows: $x'_{ij} = \sum_{k=-d}^{d} w_{j-k} x_{j-k}$, where d is the kernel size (16 in our tests) and w is the kernel window (a Gaussian function in our tests).

We followed a procedure similar to the method used by Kim et al. [14]. We first computed moving window profiles of known Trx-fold (for positive training data) and non-Trx-fold (for negative training data) proteins based on each property, and then smoothed the profiles with a width-16 Gaussian kernel. We then mapped each sequence's set of smoothed profiles to a set of attributes associated with that sequence. The *average periodicity* attributes describe how often each property's profile crosses a neutral value. For example, in Figure 2, we count the number of times the Kyte-Doolittle index crosses the neutral value 2.0 (44) and then divide this by the length of the sequence (104). So the value of attribute "crosscv-KD2.0" for 1fb0 is $44/104 = 0.423$. For each property, we chose a distinct set of five such neutral values. The second type of attributes used were based on the *first-order and second-order derivatives* of the profiles, where the first-order derivative of profile i at position j is $x_{ij} - x_{ij-1}$ and the second-order derivative of profile i is the derivative of its derivative. The attributes computed from the derivatives were the average values and the variances of the first- and second-order derivatives. Finally, we also added 20 attributes that represent the *frequencies* of each amino acid.

[1]Since ClustalW and HMMER expect symbols from the 20-amino acid alphabet, their default scoring matrices and priors are inappropriate for a 4-symbol secondary structure alphabet. So we replaced ClustalW's 20×20 scoring matrix with a 4×4 identity matrix and we replaced HMMER's default Blocks9 prior with a prior that is uniform over our 4 symbols and 0 elsewhere.

Table 1: Summary of the remappings of the 20 residue alphabet.

Residue	Charge	Volume	Mass	Hydro-4	Hydro-6
A	None	Small	Small	$[-2.0, 0.7]$	$[-0.6, -2.0]$
C	None	Medium	Medium	$[-2.0, 0.7]$	$[-0.6, -2.0]$
D	Neg	Medium	Med-Large	$[8.2, 12.3]$	$[8.2, 9.2]$
E	Neg	Med-Large	Med-Large	$[8.2, 12.3]$	$[8.2, 9.2]$
F	None	Large	Large	$[-3.7, -2.6]$	$[-3.7, -2.6]$
G	None	Small	Small	$[-2.0, 0.7]$	$[-0.6, -2.0]$
H	Neg	Med-Large	Med-Large	$[3.0, 4.8]$	$[3.0, 4.8]$
I	None	Med-Large	Med-Large	$[-3.7, -2.6]$	$[-3.7, -2.6]$
K	Pos	Med-Large	Med-Large	$[8.2, 12.3]$	$[8.2, 9.2]$
L	None	Med-Large	Med-Large	$[-3.7, -2.6]$	$[-3.7, -2.6]$
M	None	Med-Large	Med-Large	$[-3.7, -2.6]$	$[-3.7, -2.6]$
N	None	Medium	Med-Large	$[3.0, 4.8]$	$[3.0, 4.8]$
P	None	Medium	Medium	$[-2.0, 0.7]$	$[0.2, 0.7]$
Q	None	Med-Large	Med-Large	$[3.0, 4.8]$	$[3.0, 4.8]$
R	Pos	Med-Large	Large	$[8.2, 12.3]$	$[12.3, 12.3]$
S	None	Small	Medium	$[-2.0, 0.7]$	$[-0.6, -2.0]$
T	None	Medium	Medium	$[-2.0, 0.7]$	$[-0.6, -2.0]$
V	None	Med-Large	Medium	$[-3.7, -2.6]$	$[-3.7, -2.6]$
W	None	Large	Large	$[-2.0, 0.7]$	$[-0.6, -2.0]$
Y	None	Large	Large	$[-2.0, 0.7]$	$[0.2, 0.7]$

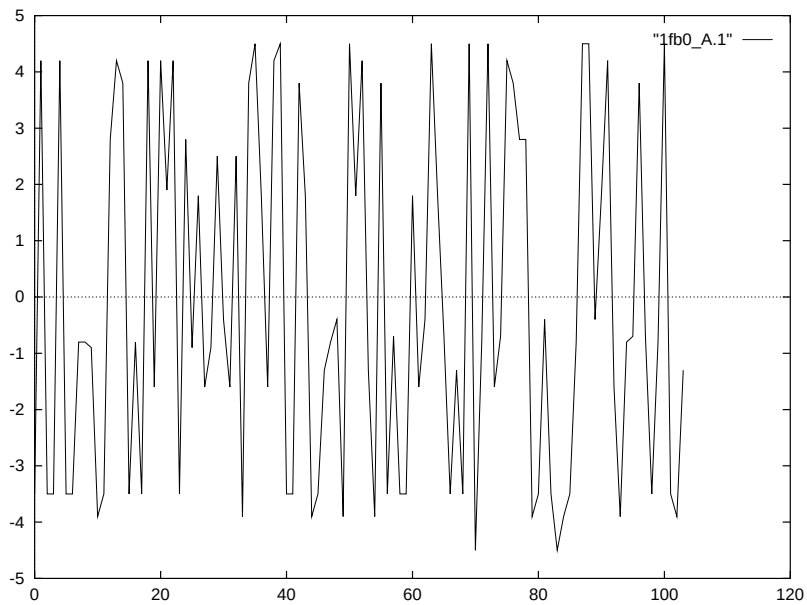

Figure 2: Plot of a profile of 1fb0 from PDB, based on Kyte-Doolittle index. On the x axis is the amino acid position and on the y axis is the value of the index.

2.3 Multiple-Instance Learning Approaches

C4.5 is an example of an algorithm in the conventional machine learning model. As such, the sequence profiles as described in Section 2.2 must be summarized into a single set of numbers for C4.5 to use, as described in the previous section. To use the profiles directly, one must use the *multiple-instance learning model* [7], in which each example is represented as a multiset (called a *bag*) of attribute vectors rather than as a single attribute vector as in the conventional learning model. Simply put, in this new model a bag is labeled as positive (Trx-fold) if and only if the attribute vectors in it satisfy some function. For example, the algorithm EM-DD [21] might assume all Trx-fold proteins have a point in their profiles near each other and this point is not near any in non-Trx-fold proteins. E.g. this algorithm might find that all Trx-fold proteins take values of -4.5 in Kyte-Doolittle around position 85, and that few non-Trx-fold proteins do. In contrast, the algorithm of Goldman et al. [10] generalizes EM-DD by looking for a *set* of points S such that each Trx-fold protein has a point near each point of a size-k subset $S' \subseteq S$ and that all non-Trx-fold proteins have points near at most $k-1$ points of S. E.g. this algorithm might find that all Trx-fold proteins satisfy one of the following conditions and that few non-Trx-fold proteins satisfy any of them: (1) a Kyte-Doolittle value of -4.5 around position 85 *and* a Kyte-Doolittle value of -0.75 near position 10; (2) a Kyte-Doolittle value of 3.5 near position 55 *and* Kyte-Doolittle value of 4.25 near position 92 *and* Kyte-Doolittle value of 1.5 near position 25; etc. Thus intuitively, Goldman et al.'s algorithm can represent more specific models than EM-DD, and thus should be able to better discriminate between Trx-fold and non-Trx-fold proteins. However, its time complexity is much worse, which limits its applicability until faster versions are available.

We mapped our data to the multiple-instance learning model in the following way. We first found the CxxC motif in each (Trx-fold and non-Trx-fold) sequence and extracted a window of size 215 around it (31 residues upstream, 180 downstream). We then mapped all sequences to their profiles based on the properties of Kim et al. [14]. We then aligned the CxxC motifs in all the sequences and used these profiles as inputs to the multiple-instance learning algorithms. Due to computational complexity issues, Goldman et al.'s algorithm could only handle 2-dimensional data of this size, so we ran it seven times, one for each property. We also tried the same thing for EM-DD, but then we also combined all seven profiles into one 8-dimensional multiple-instance data set.

3 Experimental Results

We constructed our data sets as follows. First we extracted 47 Trx-fold proteins from PDB[2], including thioredoxins and glutaredoxins, all containing the CxxC motif. Since these 47 had structural information, they were particularly important in building models on true secondary structure. We then combined these 47 proteins with a set of 226 other known Trx-fold proteins (for which secondary structure is not known) and 320 known non-Trx-fold proteins from the Non-redundant Database[3]. These data sets were then filtered for various tests, as described below.

3.1 Random Data Sets

As a first test of our techniques, we applied them on large, random data sets. We filtered our positive and negative sets such that no two sequences were more than 80% similar when pairwise aligned, yielding 183 positives and 195 negatives. We then built three HMMs: one on the sequences' primary structure, one on predicted secondary structure, and one on true secondary structure[4]. In all three cases, the sequences were aligned with ClustalW before models were built and calibrated in HMMER. Then the test set (consisting of all Trx-fold and non-Trx-fold proteins not used for training) was searched with each model. In the secondary structure test sets, only predicted structure was used since when performing database searches, the true secondary structures would not be known.

[2]http://www.rcsb.org/

[3]http://www.ncbi.nlm.nih.gov

[4]Since true secondary structure was used for one test, we used the PDB sequences for building all three models and the remaining positive and negative sequences for testing.

Table 2: Results of applying multiple-instance learning algorithms to random data sets.

Property	Goldman et al.		EM-DD	
	TP	TN	TP	TN
GES hydropathy	0.9188	0.9836	0.7817	0.8852
Kyte-Doolittle	0.9746	0.9672	0.9188	0.8579
Polarity	0.9746	0.9617	0.9036	0.8962
pI	0.9594	0.9672	0.9188	0.8907
α helix	0.9797	0.9454	0.7868	0.8579
Mol Wt	0.9797	0.9617	0.6954	0.8743
Solubility	0.9492	0.9563	0.9188	0.8743

We found that HMMER trained on primary structure can achieve true positive and true negative rates of more than 0.99. 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. In contrast, HMMER trained on predicted secondary structures can achieve both true positive and true negative rates at about 0.82, while HMMER trained on true secondary structure can only achieve both true positive and true negative rates at about 0.70. The reason for True Secondary's worse performance is that PREDATOR's errors in predicting secondary structure adds noise to the test sequences. Thus an HMM built on predicted secondary structure is also training on the noise introduced by PREDATOR, which makes it less sensitive to structure prediction inaccuracies in the database.

To test QFC/C4.5, we split our filtered data set into three sets of approximately equal sizes and ran three tests. For each test, we built a decision tree on two sets and tested on the third. In this experiment we averaged 0.81 for the true positive rate, and 0.85 for the true negative rate. We tried the same test for the multiple-instance learning algorithms, one property at a time. The results are in Table 2.

Since HMMER on primary structure can achieve true positive and true negative rates at over 99%, it is superior to our methods in identifying new sequences that are similar to the sequences it was trained on (the only exception is Goldman et al.'s algorithm, but it is significantly slower than HMMER). However, in the next section we will show that this is not the case when sequences are highly dissimilar.

3.2 Jack-Knife Tests

Within our data set, there are many similar sequences, which means that the experiments of Section 3.1 are inappropriate to evaluate our methods for the purpose they were designed. Since our goal is to identify new families, the sequences in our data set should be highly dissimilar to each other. Thus we filtered our set so that only proteins with low primary structure conservation between each other remained. We first used ClustalW to align those proteins, which also generated a dendogram that indicated the level of similarity between each pair of proteins. We generated two sets: one consisted of 9 Trx-fold proteins from PDB with very little similarity between them (1a8l, 1eej, 1bed, 1qk8, 1f9m, 1m3k, 1ego, 1fov, and 1de1), and the other consisted of 12 proteins (1eej, 1bed, 1qk8, 1f9m, 1ego, 1fov, 1de1, and some from outside PDB: GI:9989039, GI:12324654, GI:15610809, GI:1729945, GI:7109697), also of very low similarity. Pairwise identity of sequences from the set of 9 ranged from 37% to 60%, averaging 47%, and within the set of 12 it ranged from 35% to 55%, averaging 45%. The set of non-Trx-fold proteins remained the same.

Due to the small number of Trx-fold proteins available in our new data set, we performed a jack-knife (leave-one-out cross-validation) test. We held out one Trx-fold protein for use in testing and used the rest for training, repeating once for each of the 9 (12) Trx-fold proteins in the data set. So for each HMMER-based experiment, the model was built on 8 (11) Trx-fold proteins and the test set (the one that is searched by the model) consisted of all 9 (12) Trx-fold proteins and all our non-Trx-fold proteins[5]. Since QFC/C4.5

[5]We used the 8 (11) sequences from the training set in our test set for two reasons. The first was to increase the size of the test set to improve the statistical validity of our tests. The second was to compare the E-values of the hold-out to those of sequences that the model was built on. However, all error rates reported are only on sequences that were not used to build the

Table 3: Summary of results on the jack-knife tests on the set of 9 Trx-fold proteins. "EM-DD (MW)" refers to EM-DD run on only the molecular weight profiles (its best result), "EM-DD (All)" refers to EM-DD run on all profiles combined, and "Goldman (K-D)" refers to Goldman et al.'s algorithm run on the Kyte-Doolittle profiles (its best). "TP" is true positive rate and "TN" is true negative rate.

	TP	TN		TP	TN
Primary	0.000	1.000	Hydro-4	0.444	0.954
True Second.	0.333	0.931	Hydro-6	0.222	0.972
Pred. Second.	0.778	0.855	QFC/C4.5	0.667	0.671
Volume	0.000	0.986	EM-DD (MW)	1.000	0.654
Mass	0.222	0.974	EM-DD (All)	0.889	0.602
Charge	0.000	1.000	Goldman (K-D)	0.667	0.853

and the multiple-instance learning algorithms require both Trx-fold and non-Trx-fold proteins for training, we split our set of non-Trx-fold proteins into 10 equal-sized sets. We then trained our algorithms on the 8 (11) Trx-fold proteins plus one of the 10 sets of non-Trx-fold proteins, and tested on the held-out Trx-fold protein plus the remaining 9 sets of non-Trx-fold proteins. We repeated this for each of the 10 sets of non-Trx-fold proteins. Thus we ran $9 \times 10 = 90$ ($12 \times 10 = 120$) experiments for each algorithm, compared to the 9 (12) experiments for each remapped HMMER.

Our results[6] are in Tables 3 and 4. For HMMER-based experiments, we used an E-value cutoff of 0.1 as in Section 3.1. Since each jack-knife round for QFC/C4.5 and multiple-instance learning involved 10 experiments (one for each non-Trx-fold set), we gave the algorithm credit for correctly classifying the held-out Trx-fold protein if it successfully identified it at least half the time. The TP rates in the tables are the fractions (out of 9 or 12) of the set of Trx-fold proteins that each algorithm correctly identified. For the HMMER-based algorithms, TN is the fraction of non-Trx-proteins that had E-values above 0.1. For QFC/C4.5 and multiple-instance learning, TN is that algorithm's accuracy on the non-Trx-proteins over all 90 or 120 experiments. To save space, Table 3 only reports one result (the best over all 7 properties) for each of EM-DD and Goldman et al., and EM-DD over all 7 properties taken at once is also given. EM-DD on other properties performed comparably to the results in the table, with TP rates in the range $[0.5556, 1.000]$ (average 0.7222) and TN rates in the range $[0.5793, 0.6560]$ (average 0.6362). Goldman et al.'s algorithm had TP rates in the range $[0.3333, 0.6667]$ (average 0.5000) and TN rates in the range $[0.6614, 0.8526]$ (average 0.7645). Thus EM-DD was consistently better than Goldman et al. for TP rate but consistently worse for TN rate. The model built on predicted secondary structure was the overall best performer, correctly identifying 7/9 and 9/12 positives and over 0.85 of the negatives. Further, over the two experiments, four of the five positive sequences that were missed by Predicted Secondary had E-values between 0.1 and 1.0, which could be considered near-hits (the fifth missed positive had an E-value > 10).

Interestingly, there is little correlation among the methods we tested in terms of the positive sequences they found. Table 5 summarizes each algorithm's performance on each of the 9 Trx-fold proteins from the first jack-knife test. So an "H" or a large number in an entry indicates that the algorithm was successful in finding that protein. We see that the two proteins missed (but nearly hit) by Predicted Secondary are hit by Hydrophobicity-4. Further, there are some positive results for Goldman et al.'s algorithm on 1ego (for 3 properties). Since both of these other algorithms have high TN rates, it suggests that taking a union of these classifiers' hits (or at least Predicted Secondary and Hydrophobicity-4) would completely cover all 9 positives while not incurring a high false positive penalty. Indeed, if we neglect non-Trx proteins that were falsely identified in two or fewer of the nine tests, then taking the union of Predicted Secondary and Hydrophobicity-4 yields a TN rate of over 0.773. For the 12-fold jack-knife test, one of Predicted Secondary's three missed positives is picked up by Hydrophobicity-4, which also nearly hits the other two positives missed by Predicted Secondary. These last two are hit by Volume, Mass, and QFC/C4.5. So by

models.

[6] 12-fold jack-knife results for the multiple-instance learning algorithms are pending. No 12-fold jack-knife results are reported for True Secondary since structural information is unavailable for part of the training set.

Table 4: Summary of results on the jack-knife tests on the set of 12 Trx-fold proteins. Results for the multiple-instance learning algorithms are pending, and no results are reported for True Secondary since structural information is unavailable for part of the training set.

	TP	TN		TP	TN
Primary	0.000	1.000	Charge	0.000	0.999
Pred. Second.	0.750	0.889	Hydro-4	0.417	0.962
Volume	0.167	0.925	Hydro-6	0.417	0.977
Mass	0.083	0.963	QFC/C4.5	0.750	0.642

Table 5: Summary of which sequences were found by each classifier in the 9-fold jack-knife test. "H" indicates a hit, "M" a miss, and "NH" a near hit (E-value in $(0.1, 1.0]$ for HMMER-based algorithms or successfully identifying the protein 4 out of 10 times for QFC/C4.5 and multiple-instance algorithms). The "min" columns for EM-DD and Goldman et al. are the results for the best properties as reported in Table 3. The "sum" columns indicate how many properties (out of 7) allowed each of EM-DD and Goldman et al.'s algorithms to identify that protein.

ID	Tr Se	Pr Se	Vol	Mass	Chrg	H-4	H-6	QFC	EMmin	EMall	EMsum	Gmin	Gsum
1a8l	H	H	M	M	M	M	M	NH	H	H	7	H	5
1eej	M	H	M	H	NH	M	M	H	H	H	7	H	7
1bed	M	NH	M	M	M	H	M	H	H	M	3	M	0
1qk8	H	H	M	NH	M	H	H	H	H	H	5	H	5
1f9m	NH	H	M	H	M	H	H	H	H	H	6	H	4
1m3k	H	H	M	M	M	H	H	H	H	H	7	H	2
1ego	M	NH	M	M	M	H	H	NH	H	H	4	M	3
1fov	M	H	M	M	M	H	NH	M	H	H	3	M	2
1de1	M	H	M	M	M	NH	H	H	H	H	5	H	4

adding one of Volume and Mass, we can again cover 100% of the positives while maintaining a reasonable TN rate.

A final item of note about these experiments is that 1ego, 1fov, and 1de1 are glutaredoxins (Grx), which differ in secondary structure from Trx. Specifically, Trx proteins have an additional helix upstream of a typical Trx-fold structure and Grx proteins have an insertion of a helix in the middle of the Trx fold. This makes Predicted Secondary's performance (Table 5) that much more impressive since it relies strictly on secondary structure for its models.

3.3 Database Search Results

The final test of our algorithms was to search genomic databases. In this work, we analyzed the completely sequenced genomes of *Campylobacter jejuni*, *Methanococcus jannaschii*, *Escherichia coli*, and *Saccharomyces cerevisiae*. For the HMMER-based algorithms, we built our models both on the 9 Trx-fold sequences and on the 12 Trx-fold sequences from Section 3.2. For QFC/C4.5, we used these 9 and 12 Trx-fold sequences for positives, and for the negatives, we used the one (of ten) non-Trx-fold set from Section 3.2 that yielded the best performance (in our jack-knife tests, some negative training sets consistently afforded higher TP and TN rates than others).

Our search results so far[7] are listed in Table 6. In addition, QFC/C4.5 found GI:15668239 in *M.*

[7] Some hits have not yet been verified as true positives or false positives, so more hits might be added to Table 6, particularly

Table 6: Verified Trx-fold proteins found in the *C. jejuni*, *M. jannaschii*, *E. coli*, and *S. cerevisiae* databases. Proteins are listed by GI number when available. Results for QFC/C4.5 are pending.

Organism	Primary	Secondary (true)	Secondary (pred)	Hydro-4	Hydro-6	Charge	Volume	Mass
C. jejuni	6967641	6968540 6968640 6969081	6968312 6969081 6969080 6968640 6967749 6968311 6968540	6968640 6969081 6967641 6969080	6968540 6969081 6967641 6969080 6968814 6969056	6968640 6969081 6969080	6968640 6969080 6967641 6968540	6968640 6969081 6967641 6969080
M. jannaschii			15668482					15668239
E. coli	16128817		16128817	16128817	16128817		16128817	16128817
S. cerevisiae	1360373 1323375		5328 1332638 P25372 Q12404 4120 1323375 1360373 P47091	1360373 1323375 927781			1360373 1323375	

jannaschii (search results for QFC/C4.5 on the other databases and searches by the multiple-instance learning algorithms are pending). Consistent with the results of Section 3.2, some of the strongest results are with Predicted Secondary and the hydrophobicity-based mappings.

4 Discussion and Conclusion

The Trx-fold superfamily is a very important set of proteins with very low similarity in primary sequence. We have proposed numerous methods to identify new Trx-fold proteins, focusing on structural information rather than primary sequence. Jack-knife tests and database searches indicate that the most efficient and most accurate methods use predicted secondary structure and remapped alphabets based on residue hydrophobicity. They also suggest that taking a union of the results from these three models (and perhaps Volume and Mass as well) can maximize the number of accurate hits. The most credible hits would of course be those that are detected by multiple models.

Future and current work include testing our results on proteins that have a CxxS motif rather than CxxC and using the identities of the two residues embedded in the CxxC and CxxS motifs. We are also exploring building new Dirichlet mixture priors [18] on our remapped alphabets to replace the uniform priors mentioned in Footnote 1. We are also looking at using other reduced alphabets such as those produced by applying the algorithm of Cannata et al. [4]. Preliminary jack-knife tests of various alphabets produced by Cannata et al.'s algorithm (both based on the PAM70 scoring matrix) yielded TP and TN rates of 0.778 and 0.758 for a 6-letter alphabet, and 0.556 and 0.960 for a 9-letter alphabet. Further work is needed to refine these results and thoroughly compare them with our existing ones. Finally, our techniques are very general, and should be applicable to other superfamilies with low primary sequence conservation.

for *E. coli*. Thus we cannot report on the false positive rate yet.

Acknowledgments

The authors thank Adam Cook, Skanth Ganesan, Praveen Guddeti, Kevin Karplus, Gregory Kryukov, Hasan Otu, Sarathkumar Polireddy, and Subhani Shaik for their help with the experiments, and Etsuko Moriyama for help with QFC. The project described was supported by NIH Grant Number RR-P20 RR17675 from the IDeA program of the National Center for Research Resources. It was also supported in part by NSF grants CCR-0092761, CCR-9877080, and EPS-0091900.

References

[1] C. M. Andorf, D. L. Dobbs, and V. G. Honavar. Discovering protein function classification rules from reduced alphabet representations of protein sequences. In *Proceedings of the Fourth Conference on Computational Biology and Genome Informatics*, pages 1200–1206, 2002.

[2] F. Aslund and J. Beckwith. The thioredoxin superfamily: redundancy, specificity, and gray-area genomics. *Journal of Bacteriology*, 181:1375–1379, 1999.

[3] T. Brown. *Molecular Biology Labfax*. Academic Press, second edition, 1998.

[4] N. Cannata, S. Toppo, C. Romualdi, and G. Valle. Simplifying amino acid alphabets by means of a branch and bound algorithm and substitution matrices. *Bioinformatics*, 18:1102–1108, 2002.

[5] L. Debarbieux and J. Beckwith. Electron avenue: pathways of disulfide bond formation and isomerization. *Cell*, 99:117–119, 1999.

[6] G. Deleage and B. Roux. An algorithm for protein secondary structure prediction based on class prediction. *Protein Engineering*, 1:289–294, 1987.

[7] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71, 1997.

[8] D. M. Engelman, T. A. Steitz, and A. Goldman. Identifying nonpolar traasbilayer helices in amino acid sequences of membrane proteins. *Annual Review of Biophysics and Biophysical Chemistry*, 15:321–353, 1986.

[9] D. Frishman and P. Argos. Seventy-five percent accuracy in protein secondary structure prediction. *Proteins*, 27:329–335, 1997.

[10] S. A. Goldman, S. K. Kwek, and S. D. Scott. Agnostic learning of geometric patterns. *Journal of Computer and System Sciences*, 6(1):123–151, February 2001.

[11] G. Von Haijne. Membrane protein structure prediction: Hydrophobicity analysis and the positive-inside rule. *Journal of Molecular Biology*, 225:487–494, 1992.

[12] A. Holmgren. Thioredoxin and glutaredoxin systems. *J. of Biological Chemistry*, 264(24):13963–13966, 1989.

[13] A. Holmgren and M. Bjornstedt. Thioredoxin and thioredoxin reductase. *Meth. in Enzy.*, 252:199–208, 1995.

[14] J. Kim, E. N. Moriyama, C. G. Warr, P. J. Clyne, and J. R. Carlson. Identification of novel multi-transmembrane proteins from genomic databases using quasi-periodic structural properties. *Bioinformatics*, 16:767–775, 2000.

[15] J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157:105–132, 1982.

[16] J. Martin. Thioredoxin—a fold for all reasons. *Structure*, 3:245–250, 1995.

[17] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, 1993.

[18] K. Sjölander, K. Karplus, M. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *Computer Applications in the Biosciences*, 12(4):327–345, 1996.

[19] J. L. Tukey. *Exploratory data analysis*. Addison Wesley, 1977.

[20] X. Wang, D. Schroeder, D. L. Dobbs, and V. G. Honavar. Data-driven discovery of protein function classifiers: Decision trees based on meme motifs outperform prosite patterns and profiles on peptidase families. In *Proc. of the Fourth Conference on Computational Biology and Genome Informatics*, pages 1193–1199, 2002.

[21] Q. Zhang and S. A. Goldman. EM-DD: An improved multiple-instance learning technique. In *Neural Information Processing Systems 14*, 2001.