Global sequence properties for superfamily prediction: a machine learning approach

Richard JB Dobson [1] *, Patricia B Munroe [1], Mark J Caulfield [1], Mansoor Saqi [2], [3]

[1]The William Harvey Research Institute, Bart's and the London School of Medicine and Dentistry, Queen Mary University of London, Charterhouse Square, London EC1M 6BQ, UK

[2]Institute of Cell and Molecular Science, Bart's and the London School of Medicine and Dentistry, Queen Mary University of London, Charterhouse Square, London EC1M 6BQ, UK

[3]Present address: Centre for Mathematical and Computational Biology, Rothamsted Research,Harpenden, AL5 2JQ, UK

Summary

Functional annotation of a protein sequence in the absence of experimental data or clear similarity to a sequence of known function is difficult. In this study, a simple set of sequence attributes based on physicochemical and predicted structural characteristics were used as input to machine learning methods. In order to improve performance through increasing the data available for training, a technique of sequence enrichment was explored. These methods were used to predict membership to 24 and 49 large and diverse protein superfamiles from the SCOP database.

We found the best performance was obtained using an enriched training dataset. Accuracies of 66.3% and 55.6% were achieved on datasets comprising 24 and 49 superfamilies with LibSVM and AdaBoostM1 respectively.

The methods used here confirm that domains within superfamilies share global sequence properties. We show machine learning models used to predict categories within the SCOP database can be significantly improved via a simple sequence enrichment step. These approaches can be used to complement profile methods for detecting distant relationships where function is difficult to infer.

Background

Functional annotation of a new protein sequence is often obtained by database searching. If the search reveals a sequence which shares a large degree of similarity with the target sequence, an annotation can usually be transferred with some confidence. In the absence of any clear similarity with a sequence of known function, annotation becomes difficult although new algorithms which make use of information contained within groups of related sequences have helped to detect distant sequence relationships [1, 2, 3, 4].

Wilson et al. (2000) [5] have estimated that broad biological function can be conserved down to about 25% sequence identity. However, there are a large number of sequences that cannot be annotated with current methods. The annotation of proteins that currently have no functional

*To whom correspondence should be addressed. E-mail: richarddobson@gmail.com

assignment due to lack of clear sequence similarity with a protein of known function remains an important challenge [6]. This lack of annotation hinders the exploitation of some genome data and it also impacts on the understanding of biological systems as we do not have sufficient understanding of constituent parts and how they might interact.

Machine learning methods have been used to explore the problem of protein function annotation. Rather than considering the sequences as strings to be compared at a character by character level, most of these methods seek to identify global features of the sequences that might be discriminative of function. Measures of function include the enzyme commission database [7], expert classifications from Riley for *Escherichia coli* [8], the Gene Ontology [9] and categories from the Munich Information Centre for Protein Sequences (MIPS) [10].

Ding & Dubchak (2001) [11] have explored the use of support vector machines (SVMs) for protein fold prediction using the SCOP [12] protein structure database as a benchmark. The SCOP database is a manually curated resource supported by a host of automated methods to provide comprehensive and accurate descriptions of the structural relationships between proteins where the structure is known. It is a hierarchical categorization of protein structural domains where levels in the hierarchy correspond to class (reflecting the overall secondary structure composition of the protein, all α for example), fold (a general description of the spatial arrangement of the secondary structure elements), superfamily (related proteins) and family (closely related proteins where relationships are usually obvious from sequence similarity alone). Proteins in the same SCOP superfamily are believed to be related from structural and other considerations and would therefore be expected to have the same general functional role. However they include proteins which are very diverse at the level of sequence similarity and for which relatedness would not be apparent from consideration of sequence alone.

Support vector machines have also been used by Cai et al. (2003) [13] to predict protein function for 54 functional families using attributes similar to those used by Ding & Dubchak (2001) [11]. The potential of the method for the prediction of distantly related proteins has also been explored by testing the method on 24 randomly selected distantly related proteins [13]. This analysis achieved a prediction accuracy of 58.3%. Related studies on enzyme functional prediction found 72% of a set of 50 enzymes could be correctly assigned where no sequence homolog was available [14]. SVMs have also been used to distinguish enzyme structures from non-enzyme structures [15]. The most useful features included secondary structure content and amino acid frequencies. Recently, Melvin et al. (2007) [16] used SVM's for superfamily classification of distantly related proteins, but did not report the specific performance for each superfamily. They employed SVM string kernels based on PSI-BLAST hits and devised a novel approach for multiclass problems. In further work they showed improved performance over single classifiers by combining a Nearest Neighbour and SVM approach using profile kernel [17]. If the Nearest Neighbour prediction was below a threshold the instance was 'punted' to a SVM and vice-versa. Kernel functions were based upon output from either structural alignments using MAMMOTH or sequence alignment using PSI-BLAST. Wieser & Mahesan (2009) [18] used a kernel based remote homology detection method and SVMs combining sequence alignment scores and secondary structural similarity scores for predicting superfamily membership. They showed that joint sequence similarity and secondary structure similarity scores improve performance over sequence-only classifiers.

Clare & King (2003) [19] and Clare et al. (2006) [20] used decision trees with GO and MIPS functional categories for mining data on the *Saccharomyces cerevisiae* and *Arabidopsis*

thaliana genomes. Predictions achieved 75% accuracy in the *Saccharomyces cerevisiae* study and 85% precision in the *Arabidopsis thaliana* study. Attributes used were those derived from PSI-BLAST, phenotypic properties, expression data, sequence and secondary structure. Other sequence attributes that have been used for functional prediction relate to predicted sequence properties such as post translational modifications, subcellular localization and secondary structure [21] using the Riley functional classification [8] and Gene Ontology [9].

A number of novel approaches have recently been applied. Such an approach named RankProp [22] was used to create an all versus all protein similarity network based on PSI-BLAST results. Analysis of the network highlights relationships between the proteins. Autoscop [23] computes unique sequence patterns and pattern combinations for SCOP classifications. A SCOP superfamily is assigned to a pattern found in its members whenever the pattern is unique to that superfamily. It is designed to be part of a prediction pipeline and only produces predictions when very confident, preferring to pass the prediction task to an alternative method when not confident.

In this study we focused on the performance of machine learning methods to predict function for distantly related protein sequences. We used membership of a SCOP superfamily as a measure of functional relatedness [12]. We restricted the study to large and diverse SCOP superfamilies, namely those with more than 15 sequences that do not share more than 20% sequence identity. We employed a range of popular machine learning methods as implemented in the Weka workbench [24] and a web based clustered computing infrastructure was built to enable rapid identification of optimal classifiers and configurations. This tool parses and stores results in a MySQL database, whilst sending a summary to the user by email. A sequence enrichment step was introduced in order to increase the number of sequences available for training. The dataset provides a challenging benchmark but one which is very relevant to enhanced genome annotation strategies.

Methods

Domain Dataset

Two datasets were created for analysis from SCOP version 1.69. The primary analysis and main focus of the study was on the first dataset which comprised domains from single domain proteins exclusively. A second dataset was created to include domains from multi domain proteins. The inclusion of SCOP domains from multi domain protein structures may present problems with functional characterisation of a protein. Namely, the function of a multi domain protein (say composed of 2 SCOP domains A and B) may not necessarily be the sum of the functions associated with the individual constituent domains, A and B. However including SCOP domains from multi domain protein structures does lead to many more examples.

Domain sequences were obtained from the Astral20 database which contains SCOP domain sequences sharing less than 20% sequence identity [25]. Superfamilies containing fewer than 16 domains at this level of sequence redundancy were excluded because we were interested specifically in studying large, diverse superfamilies. We expected that it could be difficult to train the model with a figure much lower than this, however the choice of 16 is somewhat arbitrary. We randomly extracted 2/3 of domains from each superfamily. This subset of domains

became the training dataset upon which any enrichment steps were performed. The training datasets were used to create the models in Weka and the remaining 1/3 of domains were used as the test dataset to evaluate the models. We report the accuracies of the models using this test dataset. A single test dataset was used to allow direct comparison of model performance.

Superfamily enrichment

The SCOP database provides a gold standard structural resource with reliable comprehensive annotation meaning that we could be fairly certain that domains were accurately classified at the level of superfamily despite being diverse at the sequence level. We wished to extend the diverse set of domain sequences in the training datasets by including entries from sequence databases without known structure and therefore missing the SCOP annotation. The reason for this was to boost the numbers of instances available for training the machine learning algorithms. It was necesary to be cautious, however, because if very remote relatives were included there was a danger they may not actually be part of the same superfamily.

The following steps were performed to enrich the number of examples in each superfamily: (i) A BLAST [26] search using each of the domain sequences from the diverse SCOP superfamilies was performed against the UniRef50 database [27]. (ii) In order for a hit to be retained, the E value had to be <0.0005. (iii) Hits were excluded where $<80\%$ of the domain was aligned (iv) Hits were also excluded where the length of the aligned section of the UniRef50 hit was $<80\%$ of the length of the aligned section of the domain (to exclude hits that had long gaps within the alignment.) (v) UniRef50 hits were further excluded that matched domains of more than one superfamily in order to reduce ambiguity in superfamily membership of the hit. (vi) BLASTClust [28] was then run against the resulting SCOP domains and UniRef50 hits for each superfamily to remove redundancy. For each cluster the SCOP domains were retained as the cluster representative when present. (vii) Results were compared where BLASTClust was used to remove redundant sequences at $>20\%$ and then $>30\%$ sequence identity. BLASTClust was set at these levels of sequence identity because below 25% similar function can not confidently be inferred by sequence alone [5]. It was considered that 30% was a conservative cutoff where similar function could be confidently inferred. At a cutoff of 20%, confidence in assumption of function was lower but it was considered to be of interest to compare to the 30% cutoff.

Attributes

Attributes selected for machine learning were based upon the properties explored by Ding & Dubchak (2001) [11] who analysed protein folds in the context of the SCOP classification. These attributes relate to the hydrophobicity, Van Der Waals volume, polarity, polarizibility and predicted secondary structure of the amino acid sequence. The secondary structure (C=Coli, H=Helix, E=Strand) was predicted using PSIPRED [29]. Each amino acid was labelled as belonging to one of three groups for each of these descriptors.

All descriptors were analyzed in the context of their composition, distribution and transition along the amino acid sequence. Taking hydrophobicity as an example, the composition element comprised 3 attributes; the percentage composition of polar (P), neutral (N), hydrophobic (H) amino acids in the domain sequence. The transition was also composed of 3 hydrophobicity related attributes; the percentage frequency of P followed by N or N followed by P, the

percentage frequency of P followed by H or H followed by P and the percentage frequency of N followed by H or H followed by a N. The distribution comprised 15 hydrophobicity related attributes describing the amino acid sequence in terms of the proportion of the length of the domain sequence that contained the first, 25%, 50%, 75%, 100% of each of the groups of amino acids (P, N or H).

In addition to these previously studied properties, the amino acid sequence length (bins of length 20 amino acids) and amino acid composition were added as attributes. A total of 126 attributes were included in the machine learning analysis.

Single attribute analysis

All machine learning analysis was performed using the Weka collection of tools and algorithms [24]. In order to identify the most effective attribute in the machine learning prediction, the 1R classifying algorithm and the information gain attribute evaluator were used. The 1R classifying algorithm creates single level decision tree for each attribute and measures the prediction error rate. The information gain is the information required after using the attribute as a classifier subtracted from the information required before using the attribute as a classifier. In both algorithms the attributes were ranked in terms of their effectiveness as predictors using the default ranker search method [24].

Attribute set analysis

The performance of 32 machine learning classifiers in a total of 96 configurations (table 1) were compared for the prediction of superfamily membership based on assignment to one of 24 superfamilies using 126 amino acid based sequence attributes. The clustered implementation of Weka [24] was used to rapidly identify the optimal classifier and configuration. The resource can be accessed by contacting the author.

The enrichment process was assessed by comparing the performance using the non-enriched resources and enriched resources using a BLASTClust cut-off of 20% and 30% sequence identity. In order to assess the performance of the length of the domain as an attribute, the prediction performances using all variables were compared with the performance of all variables excluding the length of the domain sequence.

Table 1: The lineup of classifiers and configurations chosen to run as a batch job on the clustered implementation of Weka. This lineup was used to identify the best classifier configuration for predicting SCOP superfamily.

ID	Classifier
1	weka.classifiers.bayes.BayesNet – -D -Q weka.classifiers.bayes.net.search.local.HillClimber – -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator – -A 0.5
2	weka.classifiers.bayes.BayesNet – -D -Q weka.classifiers.bayes.net.search.local.K2 – -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator – -A 0.5
3	weka.classifiers.bayes.BayesNet – -D -Q weka.classifiers.bayes.net.search.local.RepeatedHillClimber – -U 10 -A 1 -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator – -A 0.5

Continued on Next Page...

Table 1 – Continued

ID	Classifier
4	weka.classifiers.bayes.BayesNet – -D -Q weka.classifiers.bayes.net.search.local.TabuSearch – -L 5 -U 10 -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator – -A 0.5
5	weka.classifiers.bayes.BayesNet – -D -Q weka.classifiers.bayes.net.search.local.TAN – -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator – -A 0.5
6	weka.classifiers.bayes.ComplementNaiveBayes
7	weka.classifiers.bayes.NaiveBayes
8	weka.classifiers.bayes.NaiveBayesMultinomial
9	weka.classifiers.functions.LibSVM – -S 0 -K 0 -D 1 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 0.1 -E 0.0010 -P 0.1 -B
10	weka.classifiers.functions.LibSVM – -S 0 -K 0 -D 1 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1 -B
11	weka.classifiers.functions.LibSVM – -S 0 -K 0 -D 1 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1 -B
12	weka.classifiers.functions.LibSVM – -S 0 -K 0 -D 1 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 10.0 -E 0.0010 -P 0.1 -B
13	weka.classifiers.functions.LibSVM – -S 0 -K 1 -D 2 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 0.1 -E 0.0010 -P 0.1 -B
14	weka.classifiers.functions.LibSVM – -S 0 -K 1 -D 2 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1 -B
15	weka.classifiers.functions.LibSVM – -S 0 -K 1 -D 2 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 10.0 -E 0.0010 -P 0.1 -B
16	weka.classifiers.functions.LibSVM – -S 0 -K 1 -D 4 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1 -B
17	weka.classifiers.functions.LibSVM – -S 0 -K 2 -D 1 -G 0.001 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1 -B
18	weka.classifiers.functions.LibSVM – -S 0 -K 2 -D 1 -G 0.005 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1 -B
19	weka.classifiers.functions.LibSVM – -S 0 -K 2 -D 1 -G 0.01 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1 -B
20	weka.classifiers.functions.LibSVM – -S 0 -K 2 -D 1 -G 0.1 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.0010 -P 0.1 -B
21	weka.classifiers.functions.MultilayerPerceptron – -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
22	weka.classifiers.functions.MultilayerPerceptron – -L 0.6 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
23	weka.classifiers.functions.MultilayerPerceptron – -L 0.8 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
24	weka.classifiers.functions.RBFNetwork – -B 2 -S 1 -R 1.0E-8 -M -1 -W 0.1
25	weka.classifiers.functions.RBFNetwork – -B 2 -S 1 -R 1.0E-8 -M -1 -W 0.3
26	weka.classifiers.functions.RBFNetwork – -B 4 -S 1 -R 1.0E-8 -M -1 -W 0.1
27	weka.classifiers.functions.RBFNetwork – -B 4 -S 1 -R 1.0E-8 -M -1 -W 0.3
28	weka.classifiers.functions.SimpleLogistic – -I 0 -M 500 -H 50 -W 0.0
29	weka.classifiers.functions.SimpleLogistic – -I 0 -M 500 -H 50 -W 0.0 -A
30	weka.classifiers.functions.SMO – -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K 'weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0'

Continued on Next Page…

Table 1 – Continued

ID	Classifier
31	weka.classifiers.functions.SMO – -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K 'weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 2.0'
32	weka.classifiers.functions.SMO – -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K 'weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 4.0'
33	weka.classifiers.functions.SMO – -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K 'weka.classifiers.functions.supportVector.RBFKernel -C 250007 -G 0.001'
34	weka.classifiers.functions.SMO – -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K 'weka.classifiers.functions.supportVector.RBFKernel -C 250007 -G 0.01'
35	weka.classifiers.functions.SMO – -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K 'weka.classifiers.functions.supportVector.RBFKernel -C 250007 -G 0.05'
36	weka.classifiers.functions.SMO – -C 1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K 'weka.classifiers.functions.supportVector.RBFKernel -C 250007 -G 0.1'
37	weka.classifiers.lazy.IB1
38	weka.classifiers.lazy.IBk
39	weka.classifiers.lazy.KStar
40	weka.classifiers.lazy.LWL
41	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART – -M 2 -C 0.25 -Q 1
42	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48
43	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.RandomForest – -I 10 -K 107 -S 1
44	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.RandomForest – -I 10 -K 67 -S 1
45	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.RandomForest – -I 10 -K 87 -S 1
46	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.REPTree
47	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 30 -W weka.classifiers.rules.PART – -M 2 -C 0.25 -Q 1
48	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 30 -W weka.classifiers.trees.J48
49	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 30 -W weka.classifiers.trees.REPTree
50	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 50 -W weka.classifiers.rules.PART – -M 2 -C 0.25 -Q 1
51	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 50 -W weka.classifiers.trees.J48
52	weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 50 -W weka.classifiers.trees.REPTree
53	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART – -M 2 -C 0.25 -Q 1
54	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 – -C 0.25 -M 2
55	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.RandomForest – -I 10 -K 0 -S 1

Continued on Next Page…

Table 1 – Continued

ID	Classifier
56	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.REPTree – -M 2 -V 0.0010 -N 3 -S 1 -L -1
57	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.AdaBoostM1 – -P 100 -S 1 -I 10 -W weka.classifiers.trees.SimpleCart – -S 1 -M 2.0 -N 5 -C 1.0
58	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART
59	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48
60	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.trees.RandomForest – -I 10 -K 107 -S 1
61	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.trees.RandomForest – -I 10 -K 67 -S 1
62	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.trees.RandomForest – -I 10 -K 87 -S 1
63	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.trees.REPTree
64	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 30 -W weka.classifiers.rules.PART
65	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 30 -W weka.classifiers.trees.J48
66	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 30 -W weka.classifiers.trees.REPTree
67	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 50 -W weka.classifiers.rules.PART
68	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 50 -W weka.classifiers.trees.J48
69	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.ClassBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 50 -W weka.classifiers.trees.REPTree
70	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART

Continued on Next Page…

Table 1 – Continued

ID	Classifier
71	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48
72	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 10 -W weka.classifiers.trees.REPTree
73	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 30 -W weka.classifiers.rules.PART
74	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 30 -W weka.classifiers.trees.J48
75	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 30 -W weka.classifiers.trees.REPTree
76	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 50 -W weka.classifiers.rules.PART
77	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 50 -W weka.classifiers.trees.J48
78	weka.classifiers.meta.END – -S 1 -I 10 -W weka.classifiers.meta.nestedDichotomies.DataNearBalancedND – -S 1 -W weka.classifiers.meta.Bagging – -P 100 -S 1 -I 50 -W weka.classifiers.trees.REPTree
79	weka.classifiers.misc.FLR
80	weka.classifiers.misc.HyperPipes
81	weka.classifiers.rules.ConjunctiveRule
82	weka.classifiers.rules.DecisionTable
83	weka.classifiers.rules.JRip
84	weka.classifiers.rules.NNge
85	weka.classifiers.rules.OneR
86	weka.classifiers.rules.PART
87	weka.classifiers.rules.ZeroR
88	weka.classifiers.trees.DecisionStump
89	weka.classifiers.trees.J48
90	weka.classifiers.trees.LMT – -I -1 -M 5 -W 0.0
91	weka.classifiers.trees.LMT – -I -1 -M 5 -W 0.0 -A
92	weka.classifiers.trees.NBTree
93	weka.classifiers.trees.RandomForest
94	weka.classifiers.trees.RandomTree
95	weka.classifiers.trees.REPTree
96	weka.classifiers.trees.SimpleCart – -S 1 -M 2.0 -N 5 -C 1.0

Measure of performance

The performance of the machine learning methods was assessed using the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The false positive rate (FPR) is

$$FPR = \frac{FP}{TN + FP}$$

The precision (P) of predictions can be described by

$$P = \frac{TP}{TP + FP}$$

and recall (R) (sensitivity) is considered to be

$$R = \frac{TP}{TP + FN}$$

The F-measure (F) combines the precision and recall measurements in the following manner

$$F = \frac{2(P * R)}{(P + R)}$$

Benchmarking

SCOP superfamilies chosen as classes in this study were specifically large diverse superfamilies whose domains shared no more than 20% sequence identity. It is typically difficult to classify members of such superfamilies using conventional sequence homology methods.

As a comparison to the machine learning methods applied the non-enriched dataset, the PSI-BLAST program was run using a similar method to Melvin et al. (2007) [16]. A database of UniRef90 sequences was initially used to create profiles for each of the studied SCOP domain sequences [27]. Each profile was then matched separately against a database composed solely of the Astral20 proteins from the studied superfamilies. Matches with an E-value < 0.0005 over 5 iterations were identified.

A definitive comparison of PSI-BLAST with a model created by an SVM (no sequence enrichment) was difficult as various measures of performance could be used. For each PSI-BLAST query, we observed the number of matches with SCOP domains from the correct superfamily and incorrect superfamily using the defined threshold. We considered a query to be correctly assigned to a superfamily (TP) when the number of hits to domains from the true superfamily exceeded the number of hits from a false superfamily.

Results

Domain Dataset

The exclusion of multi domain protein sequences in the first (single domain) dataset reduced the number of domains from 4931 to 2867 (contained within 1136 superfamilies). Excluding superfamilies that contained fewer than 16 domains at this level of sequence redundancy further reduced the number of domains to 573 contained within 24 superfamilies (columns 1 and 2 of table 2). Excluding superfamilies that contained fewer than 16 domains in the second dataset (which included domains from multi domain proteins) resulted in 49 superfamilies (1448 domains).

Superfamily enrichment

Tables 2 and 3 show the number of instances per superfamily in the single and multi domain training datasets before and after the enrichment process using BLASTClust at 20% and 30% redundancy. The periplasmic binding protein-like II superfamily (id 53850) exhibited the biggest increase (10.38 fold single domain, 10.13 fold multi domain) in the number of instances after enrichment and the restriction endonuclease-like superfamily (52980) had the smallest increase (1.15 fold single domain, 1.13 fold multi domain).

Table 2: Number of domains per superfamily (in the analysis that excluded multi domain proteins) from Astral20 before enrichment (D) and after enrichment at 20% (20E) and 30% (30E) sequence identity cutoffs

Superfamily	D	20E	30E	30E/D
46458 a.1.1 sf Globin-like	11	22	31	2.82
46689 a.4.1 sf Homeodomain-like	12	35	35	2.92
46785 a.4.5 sf “Winged helix” DNA-binding domain	25	81	114	4.56
47266 a.26.1 sf 4-helical cytokines	15	19	27	1.8
48371 a.118.1 sf ARM repeat	11	35	50	4.55
49785 b.18.1 sf Galactose-binding domain-like	13	17	21	1.62
49899 b.29.1 sf Concanavalin A-like lectins/glucanases	14	19	28	2
50249 b.40.4 sf Nucleic acid-binding proteins	19	37	58	3.05
50729 b.55.1 sf PH domain-like	11	27	27	2.45
51182 b.82.1 sf RmlC-like cupins	11	16	20	1.82
88633 b.121.4 sf Positive stranded ssRNA viruses	11	23	23	2.09
51445 c.1.8 sf (Trans)glycosidases	15	30	45	3
51735 c.2.1 sf NAD(P)-binding Rossmann-fold domains	13	24	56	4.31
52540 c.37.1 sf P-loop containing nucleoside triphosphate hydrolases	43	89	138	3.21
52833 c.47.1 sf Thioredoxin-like	17	36	41	2.41
52980 c.52.1 sf Restriction endonuclease-like	13	14	15	1.15
53335 c.66.1 sf S-adenosyl-L-methionine-dependent methyltransferases	25	71	92	3.68
53383 c.67.1 sf PLP-dependent transferases	15	36	72	4.8
53448 c.68.1 sf Nucleotide-diphospho-sugar transferases	11	20	45	4.09
53474 c.69.1 sf alpha/beta-Hydrolases	23	41	130	5.65
53850 c.94.1 sf Periplasmic binding protein-like II	13	31	135	10.38
55729 d.108.1 sf Acyl-CoA N-acyltransferases (Nat)	15	37	49	3.27
57059 g.3.6 sf omega toxin-like	15	19	19	1.27
57095 g.3.7 sf Scorpion toxin-like	12	22	22	1.83

Table 3: Number of domains per superfamily (in the analysis that included multi domain proteins) from Astral20 before enrichment (D) and after enrichment at 20% (20E) and 30% (30E) sequence identity cutoffs

Superfamily	D	20E	30E	30E/D
46458 a.1.1 sf Globin-like	11	19	24	2.18
46626 a.3.1 sf Cytochrome c	15	20	20	1.33
46689 a.4.1 sf Homeodomain-like	23	114	115	5
46785 a.4.5 sf "Winged helix" DNA-binding domain	55	144	175	3.18
47266 a.26.1 sf 4-helical cytokines	15	20	24	1.6
47473 a.39.1 sf EF-hand	13	25	34	2.62
48371 a.118.1 sf ARM repeat	17	43	53	3.12
48726 b.1.1 sf Immunoglobulin	36	103	105	2.92
49265 b.1.2 sf Fibronectin type III	21	60	61	2.9
81296 b.1.18 sf E set domains	26	52	53	2.04
49503 b.6.1 sf Cupredoxins	15	33	40	2.67
49785 b.18.1 sf Galactose-binding domain-like	21	37	43	2.05
49899 b.29.1 sf Concanavalin A-like lectins/glucanases	22	41	56	2.55
50249 b.40.4 sf Nucleic acid-binding proteins	39	68	91	2.33
50729 b.55.1 sf PH domain-like	19	47	48	2.53
51011 b.71.1 sf Glycosyl hydrolase domain	19	24	24	1.26
51182 b.82.1 sf RmlC-like cupins	12	19	22	1.83
88633 b.121.4 sf Positive stranded ssRNA viruses	15	23	23	1.53
51445 c.1.8 sf (Trans)glycosidases	33	64	99	3
51569 c.1.10 sf Aldolase	12	16	36	3
51735 c.2.1 sf NAD(P)-binding Rossmann-fold domains	47	87	137	2.91
51905 c.3.1 sf FAD/NAD(P)-binding domain	21	37	42	2
52317 c.23.16 sf Class I glutamine amidotransferase-like	11	19	23	2.09
52374 c.26.1 sf Nucleotidylyl transferase	13	24	31	2.38
52540 c.37.1 sf P-loop containing nucleoside triphosphate hydrolases	70	150	227	3.24
52833 c.47.1 sf Thioredoxin-like	28	56	71	2.54
52980 c.52.1 sf Restriction endonuclease-like	15	16	17	1.13
53067 c.55.1 sf Actin-like ATPase domain	17	27	36	2.12
53098 c.55.3 sf Ribonuclease H-like	15	28	37	2.47
53335 c.66.1 sf S-adenosyl-L-methionine-dependent methyltransferases	29	69	110	3.79
53383 c.67.1 sf PLP-dependent transferases	16	32	66	4.13
53448 c.68.1 sf Nucleotide-diphospho-sugar transferases	12	19	45	3.75
53474 c.69.1 sf alpha/beta-Hydrolases	27	54	145	5.37
53850 c.94.1 sf Periplasmic binding protein-like II	15	33	152	10.13
56784 c.108.1 sf HAD-like	11	23	47	4.27
54001 d.3.1 sf Cysteine proteinases	18	28	33	1.83
54211 d.14.1 sf Ribosomal protein S5 domain 2-like	11	21	23	2.09
54236 d.15.1 sf Ubiquitin-like	11	21	24	2.18
54373 d.16.1 sf FAD-linked reductases, C-terminal domain	11	19	19	1.73
54593 d.32.1 sf Glyoxalase/Bleomycin resistance protein/Dihydroxybiphenyl dioxygenase	12	19	19	1.58
55347 d.81.1 sf Glyceraldehyde-3-phosphate dehydrogenase-like, C-terminal domain	11	23	32	2.91
55486 d.92.1 sf Metalloproteases ("zincins"), catalytic domain	18	44	54	3
55729 d.108.1 sf Acyl-CoA N-acyltransferases (Nat)	11	25	41	3.73
56672 e.8.1 sf DNA/RNA polymerases	11	23	56	5.09
57059 g.3.6 sf omega toxin-like	15	19	19	1.27
57095 g.3.7 sf Scorpion toxin-like	12	20	20	1.67

Continued on Next Page…

Table 3 – Continued

Superfamily	D	20E	30E	30E/D
57196 g.3.11 sf EGF/Laminin	11	55	55	5
57667 g.37.1 sf C2H2 and C2HC zinc fingers	15	40	40	2.67
57716 g.39.1 sf Glucocorticoid receptor-like (DNA-binding domain)	13	22	22	1.69

Single attribute analysis

The top ten single attributes in the non-enriched datasets and at levels of 20% and 30% enrichment comprised attributes relating to the composition, transition and distribution of secondary structure elements (coil, helix, strand) and the length of the domain. The domain length was in the top 5 attributes in all but one training set and algorithm combinations.

Attribute set analysis

Single domain dataset

Generally performance of models in predicting superfamily membership increased with the increasing level of training dataset enrichment (figure 1). Three LibSVM classifiers (18, 19, 20 in table 1) failed to complete for both enrichment levels due to a memory shortage. Best performing classifiers were the END classifier achieving 64.2% correctly classified instances on the non-enriched dataset, AdaBoostM1 obtaining 64.2% correctly classified instances on the dataset enriched at a level of 20% and LibSVM achieving 66.3% correctly classified instances with a dataset enriched at a level of 30%. END is a meta classifier for handling multi class datasets with 2-class classifiers by building an ensemble of nested dichotomies [30]. AdaBoostM1 is a class for boosting a nominal class classifier using the Adaboost M1 method [31]. The prediction performance was largely unaffected by the exclusion of length as an attribute (not shown).

The classifiers varied greatly in predicting each superfamily. Table 4 shows the performance per superfamily of the single domain dataset using LibSVM at enrichment of 30%. The model achieved the best performance in predicting membership to the ARM repeat superfamily (id 48371) (all alpha proteins class (id 46456)) with an F-measure of 0.91. The poorest performing superfamily was the nucleotide-diphospho-sugar transferases superfamily (id 53448) (alpha and beta proteins class a/b (id 51349)) with an F-measure of 0.

Figure 1: Plot showing the performance of each of the 96 classifier configurations on the dataset of 24 superfamilies (excluding multi domain proteins). Classifiers on the *x* axis are defined in table 1.

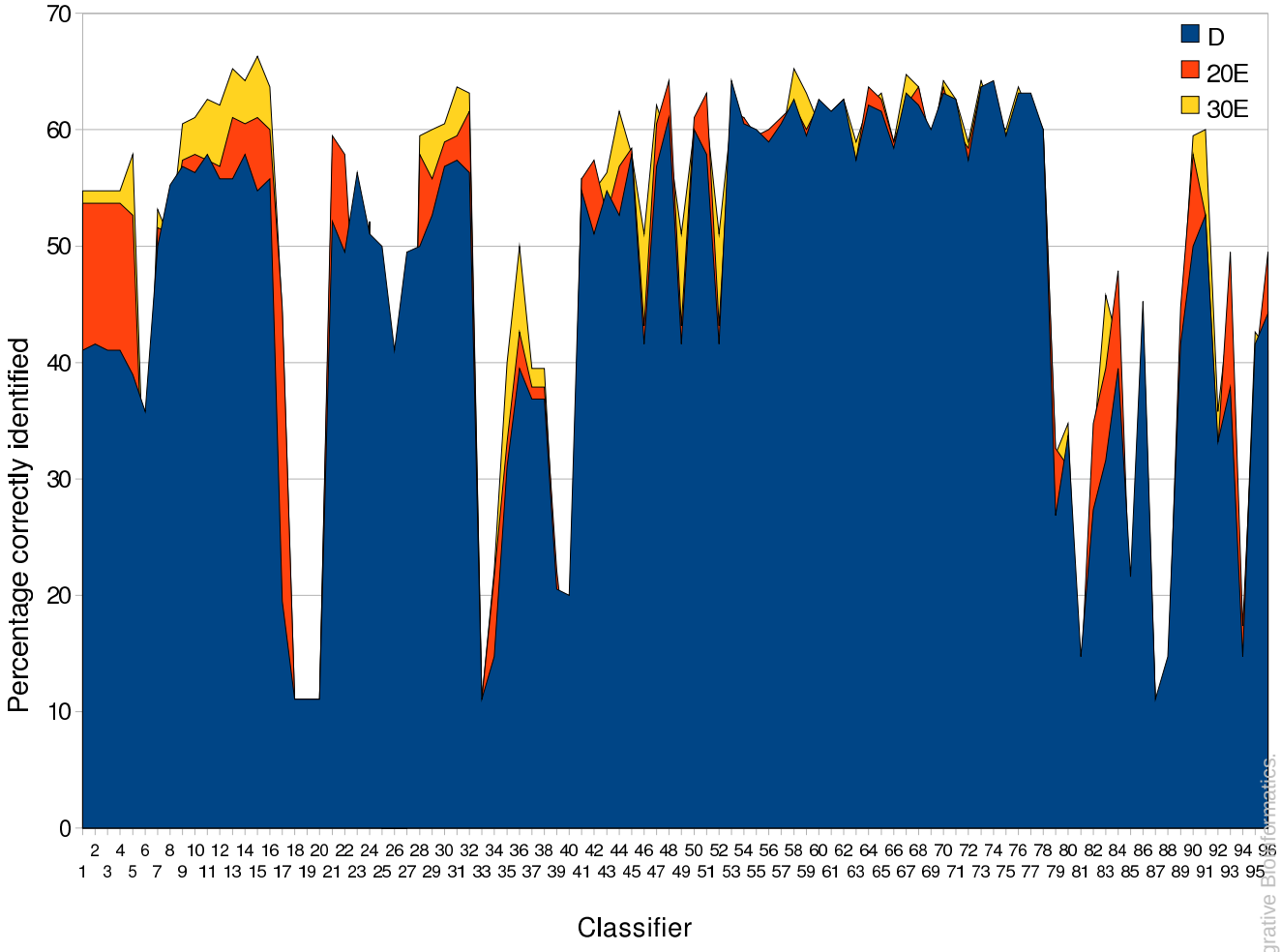


Table 4: Performance in prediction of each of 24 SCOP superfamilies (analysis that excluded multi domain proteins) studied using Support Vector Machines (LibSVM) with enrichment at a redundancy cutoff of 30% (FP=false positive).

Superfamily	FP Rate	Precision	Recall	F-Measure
46458 a.1.1 sf Globin-like	0.01	0.8	0.8	0.8
46689 a.4.1 sf Homeodomain-like	0.01	0.67	0.67	0.67
46785 a.4.5 sf Winged helix DNA-binding domain	0.02	0.77	0.83	0.8
47266 a.26.1 sf 4-helical cytokines	0.01	0.86	0.75	0.8
48371 a.118.1 sf ARM repeat	0	1	0.83	0.91
49785 b.18.1 sf Galactose-binding domain-like	0.01	0.75	0.5	0.6
49899 b.29.1 sf Concanavalin A-like lectins/glucanases	0.01	0.67	0.57	0.62
50249 b.40.4 sf Nucleic acid-binding protein	0.03	0.57	0.89	0.7
50729 b.55.1 sf PH domain-like	0.01	0.75	0.5	0.6
51182 b.82.1 sf RmlC-like cupins	0	1	0.2	0.33
88633 b.121.4 sf Positive stranded ssRNA viruses	0.02	0.57	0.8	0.67
51445 c.1.8 sf (Trans)glycosidases	0	1	0.43	0.6
51735 c.2.1 sf NAD(P)-binding Rossmann-fold domains	0.01	0.8	0.67	0.73
52540 c.37.1 sf P-loop containing nucleoside triphosphate hydrolases	0.09	0.52	0.76	0.62
52833 c.47.1 sf Thioredoxin-like	0.01	0.86	0.67	0.75
52980 c.52.1 sf Restriction endonuclease-like	0	1	0.14	0.25

Continued on Next Page...

Table 4 – Continued

Superfamily	FP Rate	Precision	Recall	F-Measure
53335 c.66.1 sf S-adenosyl-L-methionine-dependent methyltransferases	0.06	0.35	0.46	0.4
53383 c.67.1 sf PLP-dependent transferases	0.01	0.78	0.88	0.82
53448 c.68.1 sf Nucleotide-diphospho-sugar transferases	0	0	0	0
53474 c.69.1 sf alpha/beta-Hydrolases	0.02	0.75	0.82	0.78
53850 c.94.1 sf Periplasmic binding protein-like II	0.03	0.55	0.86	0.67
55729 d.108.1 sf Acyl-CoA N-acyltransferases (Nat)	0.02	0.56	0.71	0.63
57059 g.3.6 sf omega toxin-like	0.01	0.8	0.57	0.67
57095 g.3.7 sf Scorpion toxin-like	0.02	0.63	0.83	0.71

Multi domain dataset

There were 49 superfamilies (1448 domains) that had more than 15 domains within Astral20 when including multi domain proteins. The class alpha and beta proteins a+b (id 53931) is better represented in this analysis.

The same configurations and evaluation method were applied to this dataset. Best performing classifiers for each dataset were AdaBoostM1 achieving 48.5% correctly classified instances using the non-enriched dataset, END obtaining 53.7% correctly classified instances on the training dataset enriched at 20%, and AdaBoostM1 achieving 55.6% on the training dataset enriched at 30% (figure 2). The RBFNetwork classifier configurations (24, 25, 26, 27 in table 1) failed to complete for all datasets due to a memory shortage. This classifier implements a normalized Gaussian radial basisbasis function network [24]. The results of machine learning for each superfamily using AdaBoostM1 and enrichment at sequence identity of 30% can be seen in table 5.

Again, the success of the machine learning methods in predicting SCOP superfamily varied greatly depending on the superfamily with F-measure ranging from 0 to 0.92. The top performing superfamilies were the globin-like (id 46458) (all alpha protein class (id 46456)), and C2H2 and C2HC zinc finger (id 57667) (small proteins class (id 56992)) superfamilies, both with an F-measure of 0.92. The ARM repeat superfamily (id 48371) still performed well being ranked 4th in terms of F-measure (0.82). The poorest performing superfamilies were the restriction endonuclease-like (id 52980), the nucleotidylyl transferase (id 52374) (both belonging to the alpha and beta proteins class a/b (id 51349)) and the cysteine proteinases (id 54001) (Alpha and beta proteins a+b (id 53931)) superfamilies, all with F-measures of 0. The ARM repeat superfamily (id 48371) still performed well being ranked 4th in terms of F-measure (0.82).

Figure 2: Plot showing the performance of each of the 96 classifier configurations on the dataset of 49 superfamilies (including multi domain proteins). Classifiers on the *x* axis are defined in table 1.

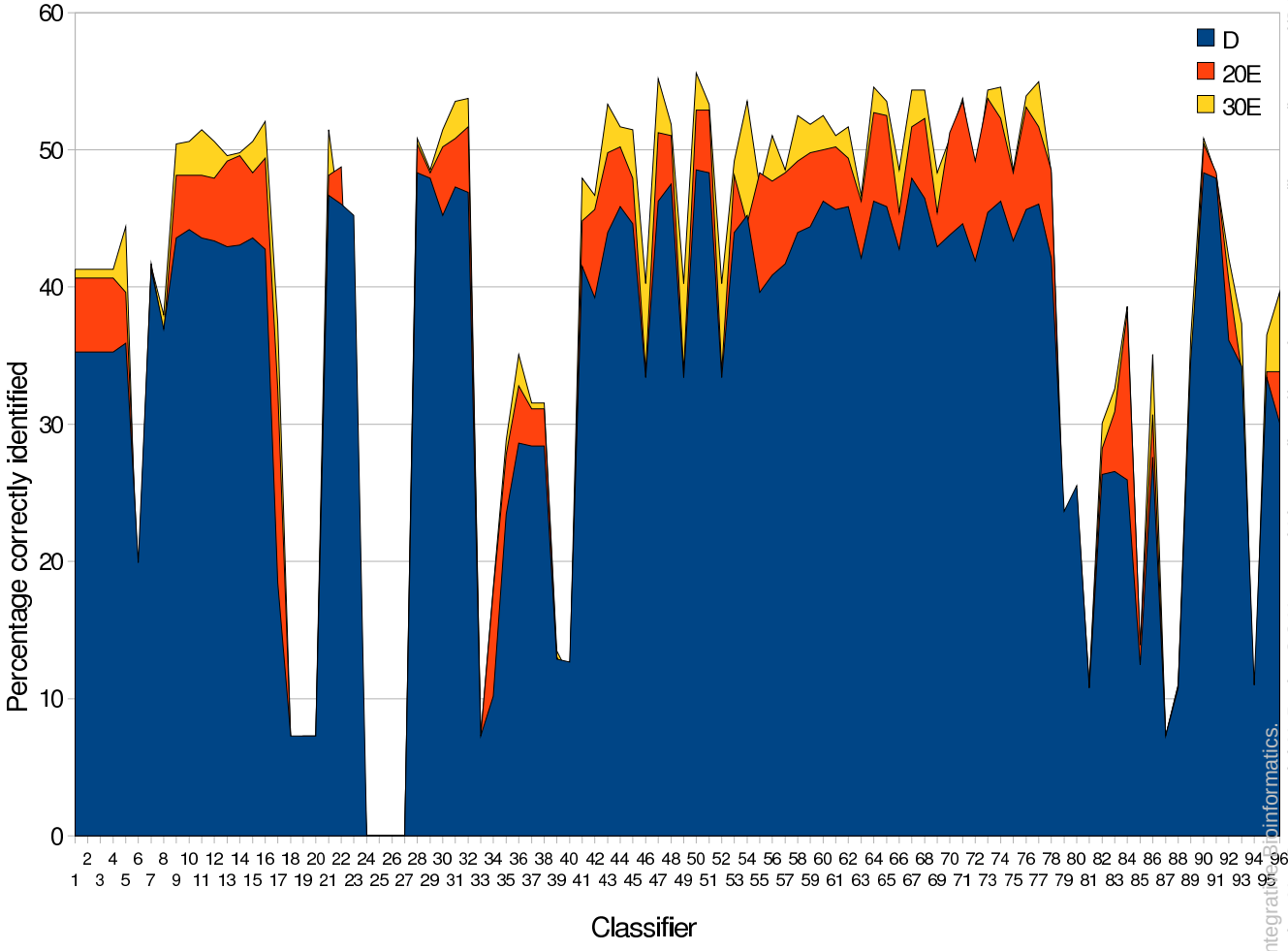


Table 5: Performance in prediction of each of 49 SCOP superfamilies (analysis that included multi domain proteins) studied using AdaBoostM1 with enrichment at a redundancy cutoff of 30% (FP=False positive).

Superfamily	FP Rate	Precision	Recall	F-Measure
46458 a.1.1 sf Globin-like	0	0.86	1	0.92
46626 a.3.1 sf Cytochrome c	0.01	0.67	0.75	0.71
46689 a.4.1 sf Homeodomain-like	0.01	0.57	0.67	0.62
46785 a.4.5 sf “Winged helix” DNA-binding domain	0.02	0.65	0.74	0.69
47266 a.26.1 sf 4-helical cytokines	0	1	0.38	0.55
47473 a.39.1 sf EF-hand	0	0.5	0.17	0.25
48371 a.118.1 sf ARM repeat	0	0.78	0.88	0.82
48726 b.1.1 sf Immunoglobulin	0.02	0.68	0.83	0.75
49265 b.1.2 sf Fibronectin type III	0.02	0.42	0.5	0.46
81296 b.1.18 sf E set domains	0.02	0.22	0.15	0.18
49503 b.6.1 sf Cupredoxins	0	0.67	0.5	0.57
49785 b.18.1 sf Galactose-binding domain-like	0.01	0.57	0.8	0.67
49899 b.29.1 sf Concanavalin A-like lectins/glucanases	0.01	0.57	0.73	0.64
50249 b.40.4 sf Nucleic acid-binding proteins	0.02	0.38	0.3	0.33
50729 b.55.1 sf PH domain-like	0.01	0.55	0.6	0.57

Continued on Next Page…

Table 5 – Continued

Superfamily	FP Rate	Precision	Recall	F-Measure
51011 b.71.1 sf Glycosyl hydrolase domain	0.01	0.4	0.4	0.4
51182 b.82.1 sf RmlC-like cupins	0	0.67	0.67	0.67
88633 b.121.4 sf Positive stranded ssRNA viruses	0	0.75	0.43	0.55
51445 c.1.8 sf (Trans)glycosidases	0.01	0.77	0.81	0.79
51569 c.1.10 sf Aldolase	0	0.67	0.33	0.44
51735 c.2.1 sf NAD(P)-binding Rossmann-fold domains	0.04	0.5	0.7	0.58
51905 c.3.1 sf FAD/NAD(P)-binding domain	0	0.6	0.27	0.38
52317 c.23.16 sf Class I glutamine amidotransferase-like	0	0.5	0.4	0.44
52374 c.26.1 sf Nucleotidylyl transferase	0	0	0	0
52540 c.37.1 sf P-loop containing nucleoside triphosphate hydrolases	0.07	0.36	0.49	0.42
52833 c.47.1 sf Thioredoxin-like	0.02	0.61	0.79	0.69
52980 c.52.1 sf Restriction endonuclease-like	0	0	0	0
53067 c.55.1 sf Actin-like ATPase domain	0.01	0.43	0.33	0.38
53098 c.55.3 sf Ribonuclease H-like	0.01	0.56	0.63	0.59
53335 c.66.1 sf S-adenosyl-L-methionine-dependent methyltransferases	0.02	0.39	0.36	0.37
53383 c.67.1 sf PLP-dependent transferases	0.01	0.7	0.88	0.78
53448 c.68.1 sf Nucleotide-diphospho-sugar transferases	0	0.33	0.17	0.22
53474 c.69.1 sf alpha/beta-Hydrolases	0.02	0.42	0.62	0.5
53850 c.94.1 sf Periplasmic binding protein-like II	0.02	0.38	0.75	0.5
56784 c.108.1 sf HAD-like	0	0.5	0.4	0.44
54001 d.3.1 sf Cysteine proteinases	0	0	0	0
54211 d.14.1 sf Ribosomal protein S5 domain 2-like	0	0.6	0.33	0.43
54236 d.15.1 sf Ubiquitin-like	0	0.71	0.83	0.77
54373 d.16.1 sf FAD-linked reductases, C-terminal domain	0	1	0.6	0.75
54593 d.32.1 sf Glyoxalase/Bleomycin resistance protein/Dihydroxybiphenyl dioxygenase	0	0.8	0.67	0.73
55347 d.81.1 sf Glyceraldehyde-3-phosphate dehydrogenase-like, C-terminal domain	0	0.5	0.33	0.4
55486 d.92.1 sf Metalloproteases (“zincins”), catalytic domain	0.01	0.5	0.5	0.5
55729 d.108.1 sf Acyl-CoA N-acyltransferases (Nat)	0	0.71	0.56	0.63
56672 e.8.1 sf DNA/RNA polymerases	0	0.8	0.8	0.8
57059 g.3.6 sf omega toxin-like	0	0.5	0.29	0.36
57095 g.3.7 sf Scorpion toxin-like	0.01	0.57	0.67	0.62
57196 g.3.11 sf EGF/Laminin	0	0.71	1	0.83
57667 g.37.1 sf C2H2 and C2HC zinc fingers	0	1	0.86	0.92
57716 g.39.1 sf Glucocorticoid receptor-like (DNA-binding domain)	0	0.71	0.71	0.71

Benchmarking

Performance of PSI-BLAST and SVMs using the non-enriched dataset was very variable, with the two methods often differing in performance for each superfamily. We found that 8 out of 24 superfamilies achieved a better F-measure with SVMs in the single domain analysis and 10 out of 49 obtained a greater F-measure in the multi domain analysis. F-measures were comparable for many other superfamilies, especially in the single domain study. SVMs outperformed PSI-BLAST for all 5 of the studied superfamilies from the small protein class (id 56992) as well as

performing better or comparably for superfamilies of the all alpha proteins class (id 46456).

Discussion

The SCOP database provides a gold standard structural resource with reliable comprehensive annotation meaning that domains should be accurately classified at the level of superfamily despite being diverse at the sequence level. It is desirable to be able to build machine learning models in order to be able to assign this functional annotation to domains where the structure is unknown and function is difficult to infer by traditional methods.

Machine learning methods benefit from having more training data. Our seed data sets, namely 24 and 49 large and sequence diverse (no two sequences sharing more than 20% sequence identity) superfamilies, provide a 'ground truth' since we know from SCOP (which uses structural and other considerations) that the proteins are in fact related. However, the datasets were somewhat limited in size and the question of how to extend them was not trivial: adding very weakly related sequences detected by PSI-BLAST might contaminate the superfamily by introducing proteins which in fact did not belong to the superfamily; but being very restrictive with the cut off would only add more examples of close homologs. We observed that the performance of the machine learning algorithms improved when the SCOP superfamily datasets were enriched and that the percentage of sequence similarity used as a cutoff in the enrichment process had an effect on the prediction performance. The performance at the sequence identity cutoff of 30% was better than the lower cutoff of 20% (figures 1 and 2). At the 20% level there existed the possibility of contamination and also the possibility of alignment errors which would affect the predicted secondary structure attributes. These effects may have led to a lower performance. We expect that the enrichment step could lead to improved accuracy in published studies that have focused on the prediction of SCOP protein fold [11, 32, 33, 16, 34, 35].

Attributes vary in their contributions to the predictions of superfamily membership in the machine learning models. Jensen et al. (2002) [21] previously concluded that secondary structure was the most important descriptor in their protein function predictions. In this study we also found that predicted secondary structure was important in predicting function with the composition, transition and distribution of secondary structure elements being the most important attributes in the single attribute analysis. Jenson *et al.* concluded that protein length was not a valuable attribute in their studies. However, we found that the length of the sequence was a valuable attribute in the single attribute analysis for the 24 superfamilies. Figure 3 shows many superfamilies display a clustering with regards to length in the non-enriched single domain resource. All 4 domains over 550 residues belong to the ARM repeat superfamily (48371 a.118.1 sf), with the Importin beta domain (d1qgra_ a.118.1.1) being the longest domain (877 residues). However, when combining all attributes for use with the 32 applied classifiers, the exclusion of domain length as an attribute had little affect on the overall performance of the model. This suggests that the classifiers were not dependant on domain length as an attribute and other sequence properties are important in accurately classifying superfamilies.

In analysis of the combined attributes in the single domain resource, the best performing classifier was LibSVM obtaining 66.3% correctly classified instances in an independent test set using a training dataset that was enriched at a level of 30% sequence identity. The success of the machine learning methods in predicting SCOP superfamily varied greatly depending on the

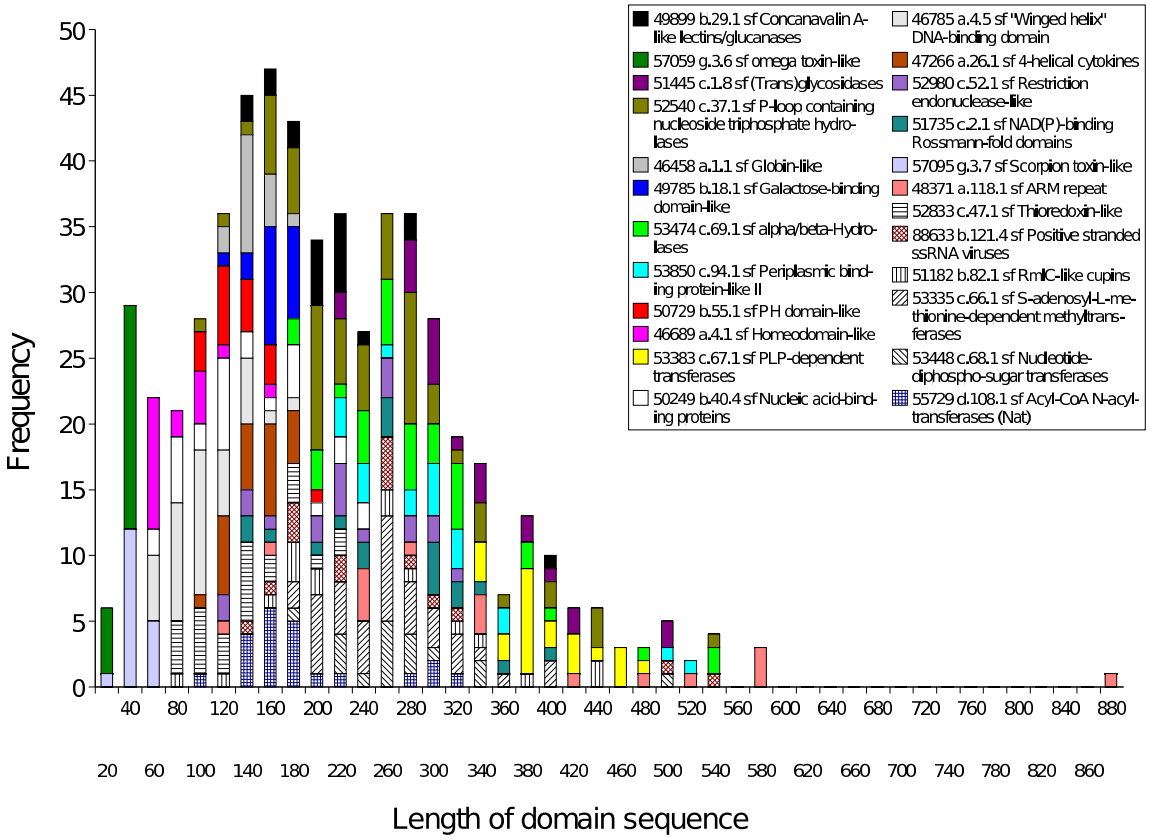


Figure 3: Sequence length of domains in superfamilies from Astral20 that contain >15 domains (excluding multi domain proteins). The length is grouped into bins of 20 amino acids. This figure highlights the clustering of superfamilies by the length of the domains within them. The Importin beta domain (d1qgra_ a.118.1.1) from the ARM repeat superfamily (48371 a.118.1 sf) is the longest domain (877 residues)

superfamily (table 4). The P-loop containing nucleoside triphosphate hydrolases (id 52540) and S-adenosyl-L-methionine-dependent methyltransferases (id 53335) had a large proportion of false positives. Thirty eight percent of instances belonging to the 53335 superfamily were classified as 52540 and 15% of 52540 instances were classified as 53335 suggesting that there is some similarity between these superfamilies or that the diversity of both groups means that classifying the two is difficult. Both superfamilies belong to the same alpha and beta proteins (a/b) (id 51349) class but are members of different folds within the SCOP classification. Figure 4 shows clearly (black bordered squares) that when the model misclassifies an instance, it usually classifies it correctly at the SCOP class level. This may reflect that the 'predicted secondary structure' attribute facilitated the correct class assignment (the SCOP class level represents the overall secondary structure composition of the protein). The alpha and beta proteins a/b (id 51349) class contains the largest number of superfamilies (10) in this study, resulting in some misclassifications among the superfamilies that it contains. The poorest performing superfamilies, nucleotide-diphospho-sugar transferases (id 53448) and restriction endonuclease-like (id 52980), both belong to this class. The 52980 superfamily also contains the smallest number of instances (15) in the training dataset enriched at 30%. The best performing superfamily, ARM repeat (id 48371), belongs to a class (all alpha proteins (id 46456)) containing only 5 superfamilies from this study and has 50 instances in the 30% enriched training dataset. It might be expected that there would be many misclassifications between the homeodomain-like su-

perfamily (id 46689) and the "winged helix" DNA-binding domain superfamily (id 46785) as both of these superfamilies belong to the same SCOP fold (DNA/RNA-binding 3-helical bundle). Whilst 33% of 46689 instances were misclassified as 46785, only 16% of 46785 instances were misclassified as 46689. This may be explained by the large number instances belonging to the 46785 superfamily, 114 at 30% enrichment, compared to 35% for the 46689 superfamily (table 2). The larger number of instances may have resulted in a better model being constructed. Therefore, it appears that the diversity of superfamilies at the class level as well as the number of instances available for training affect the performance of the classifiers.

The inclusion of multi domain proteins resulted in there being over twice the number of superfamilies available for study, with the alpha and beta proteins a+b (id 53931) class having a larger number of superfamilies available. The AdaBoostM1 classifier obtained 55.6% accuracy with a training dataset enriched at 30%. The classifiers still performed well despite the increase in the number of superfamilies resulting from the inclusion of domains from multi domain proteins. We would expect a classifier which assigned randomly to superfamilies to obtain 2% (1/49) correctly classified instances. Again, the success of the machine learning methods in predicting SCOP superfamily varied greatly depending on the superfamily (table 5). The restriction endonuclease-like (id 52980), nucleotidylyl transferase (id 52374) and cysteine proteinases (id 54001) superfamilies all performed poorly with F-measures of 0. The top performing superfamilies were the globin-like (id 46458) and C2H2 and C2HC zinc finger (id 57667) superfamilies. The globin-like (id 46458) superfamily was ranked 3rd in the single domain analysis whereas the C2H2 and C2HC zinc finger (id 57667) superfamily was absent. The globin-like (id 46458) superfamily was ranked 29th in terms of the fold increase in the number of instances after the enrichment step at 30% and was ranked 35th in terms of the total number of instances after the enrichment. The C2H2 and C2HC zinc finger (id 57667) superfamily was ranked 19th in terms of the fold increase in the number of instances after the enrichment step at 30% and was ranked 29th in terms of the total number of instances after the enrichment. It therefore seems unlikely that performance was biased towards these superfamilies due to imbalance in the dataset. Again, the P-loop containing nucleoside triphosphate hydrolases (id 52540) had a large proportion of false positives (64%). Additionally 23% and 30% of domains from the superfamily E set domains (id 81296) were misclassified as superfamilies Immunoglobulin (id 48726) and fibronectin type III (id 49265) respectively. Both superfamilies belong to Immunoglobulin-like beta-sandwich fold (id 48725) which is part of the all beta proteins class (id 48724) and were excluded from the single domain dataset. Generally, similar patterns were observed in the single and multi domain datasets with misclassifications at the superfamily level being correctly assigned at the fold or class level. Superfamilies that performed best in both the single and multi domain analysis belonged to either the all alpha protein (id 46456) or small protein classes (id 56992). Poorest performers belonged to the alpha and beta classes (a/b or a+b) (ids 51349, 53931).

For most superfamilies, PSI-BLAST did not detect unrelated domains with scores better than the threshold, although the program failed to detect all the possible correct matches (ie to related domains). For these superfamiles, the definition of a correct assignment, namely that the number of hits to domains from the true superfamily exceeded the number of hits from a false superfamily, meant that the precision was 1.0 leading to a boosted F-measure. A more exacting requirement for confident classification would be the identification of many (ideally all) related domains with scores better than the threshold. As an example, we describe the breakdown of PSI-BLAST results for 2 superfamilies. The globin-like superfamily (id 46458) performed well

within the PSI-BLAST results in the single domain analysis (2nd top F-measure). Fourteen out of 15 domains in this superfamily were assigned to the true superfamily. However, of these 14, 4 were classified based on single matches, that is PSI-BLAST only detected a match to one other protein in the same superfamily. The "Winged helix" DNA-binding domain superfamily (46785) produced relatively poor results with PSI-BLAST (F-measure 0.49). Of the 37 domains within this superfamily, only 12 were assigned to the true superfamily. Matches for 5 of these 12 were based on single hits and the maximum number of correctly returned domains for any query was 8. So, almost half of the assignments were not confident classifications.

The comparison with PSI-BLAST for the detection of these remotely related proteins shows that there are global sequence properties that can be used to successfully classify domains from superfamilies, with the performance in many cases depending on the class that the superfamily belongs to. The use of methods described in this study could be especially useful for many of the superfamilies that perform poorly in the PSI-BLAST comparison. In performing the sequence enrichment step we exploit BLAST to improve performance of the classifiers.

The protein universe does not contain only 24 or 49 superfamilies and we have not allowed for this possibility. The approach we describe does not allow for an extra category 'unknown superfamily'. One area of improvement would involve providing a method for identifying an instance that does not belong to any of the studied (24 or 49) superfamilies. This might evolve as a pre-process step. Additionally we would not expect the attributes we have used to be optimal for detecting close sequence relationships for which good solutions already exist.

Conclusions

Whilst the methods described here do not provide a complete solution for superfamily prediction they show that machine learning methods that consider simple sets of global sequences based attributes may be useful for suggesting superfamily membership and hence narrow down the potential functional space, especially for superfamilies belonging to all alpha (id 46456) and small protein classes (id 56992). We show that machine learning approaches to predicting SCOP categories can be improved by performing a sequence enrichment step that exploits unannotated sequences within genomic sequence databases. As such these approaches may complement profile methods for detecting distant relationships where function is difficult to infer.

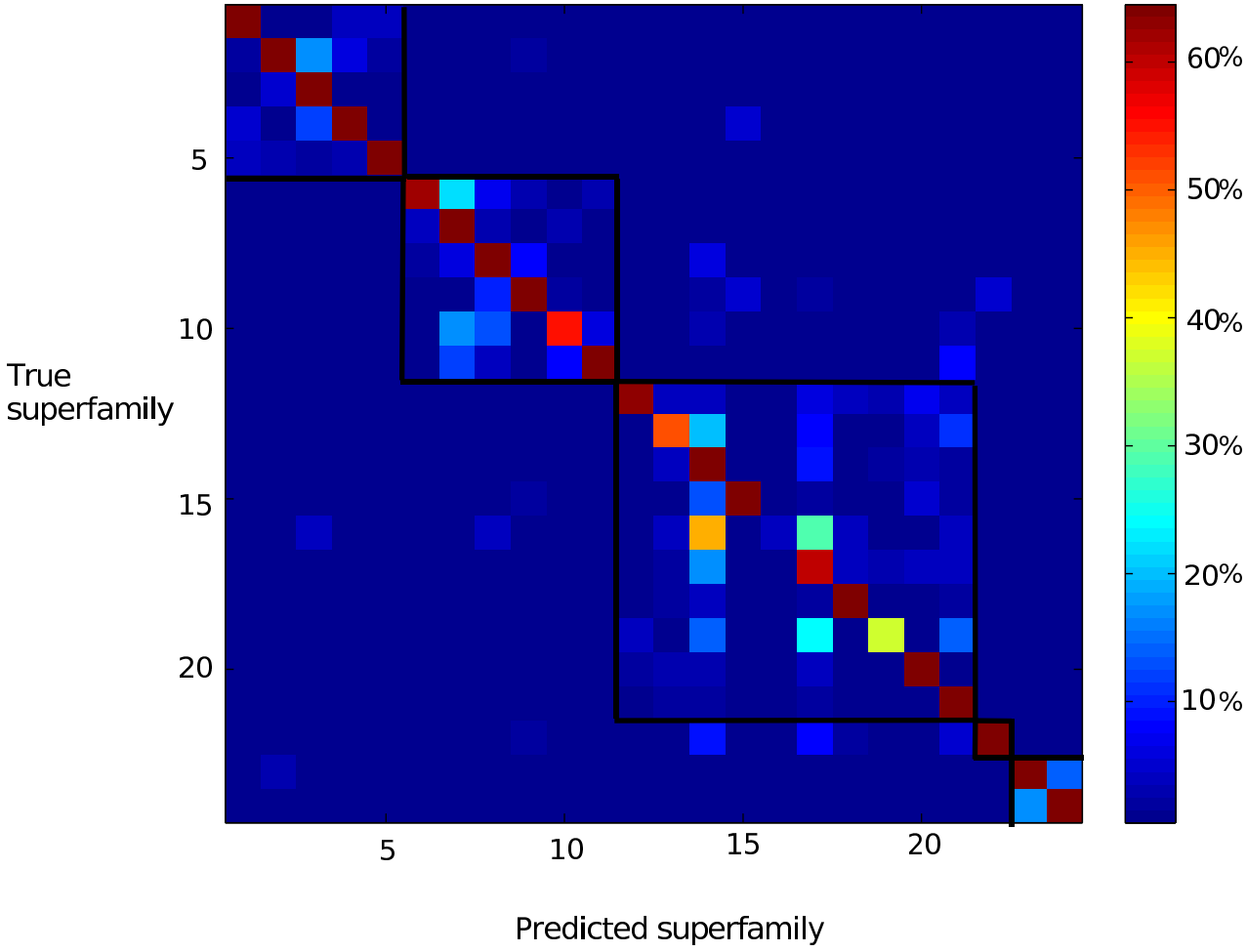
Acknowledgements

The authors would like to thank Alberto Paccanaro and Harri Saarikoski for their input regarding the machine learning methods. This work was funded by the MRC Programme Grant No. G9521010 (British Genetics of Hypertension [BRIGHT] study).

Disclosure Statement

No commercial associations reported by any authors. No competing financial interests exist.

Figure 4: Superfamily confusion matrix produced by the SVM model enriched at 30% sequence identity (excluding multi domain proteins). Each small square represents the percentage of domains belonging to the superfamily on the y axis (true superfamily) that were predicted to belong to the superfamily on the x axis (predicted superfamily). The colour of each square relates to the predicted percentage of total instances according to the colour ramp on the right of the matrix. Black bordered squares represent the 5 classes that the 24 superfamilies are grouped into. This figure highlights the fact that when a domain is miclassified at the superfamily level, it is usually correctly assigned at the class level within the SCOP hierarchy.



Axis labels:

[1] 46458 a.1.1 sf Globin-like, [2] 46689 a.4.1 sf Homeodomain-like, [3] 46785 a.4.5 sf "Winged helix" DNA-binding domain, [4] 47266 a.26.1 sf 4-helical cytokines, [5] 48371 a.118.1 sf ARM repeat, [6] 49785 b.18.1 sf Galactose-binding domain-like, [7] 49899 b.29.1 sf Concanavalin A-like lectins/glucanases, [8] 50249 b.40.4 sf Nucleic acid-binding proteins, [9] 50729 b.55.1 sf PH domain-like, [10] 51182 b.82.1 sf RmlC-like cupins, [11] 88633 b.121.4 sf Positive stranded ssRNA viruses, [12] 51445 c.1.8 sf (Trans)glycosidases, [13] 51735 c.2.1 sf NAD(P)-binding Rossmann-fold domains, [14] 52540 c.37.1 sf P-loop containing nucleoside triphosphate hydrolases, [15] 52833 c.47.1 sf Thioredoxin-like, [16] 52980 c.52.1 sf Restriction endonuclease-like, [17] 53335 c.66.1 sf S-adenosyl-L-methionine-dependent methyltransferase, [18] 53383 c.67.1 sf PLP-dependent transferases, [19] 53448 c.68.1 sf Nucleotide-diphospho-sugar transferases, [20] 53474 c.69.1 sf alpha/beta-Hydrolases, [21] 53850 c.94.1 sf Periplasmic binding protein-like II, [22] 55729 d.108.1 sf Acyl-CoA N-acyltransferases (Nat), [23] 57059 g.3.6 sf omega toxin-like, [24] 57095 g.3.7 sf Scorpion toxin-like

References

[1] R. Sadreyev and N. Grishin. COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. *J Mol Biol*, 326(1):317–36, 2003.

[2] C. L. Tang, L. Xie, I. Y. Koh, S. Posy, E. Alexov, and B. Honig. On the role of structural information in remote homology detection and sequence alignment: new methods using hybrid sequence profiles. *J Mol Biol*, 334(5):1043–62, 2003.

[3] G. Yona and M. Levitt. Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J Mol Biol*, 315(5):1257–75, 2002.

[4] J. Soding. Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21(7):951–60, 2005.

[5] C.A. Wilson, J. Kreychman, and M. Gerstein. Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. *J Mol Biol*, 297(1):233–49, 2000.

[6] M. N. Wass and M. J. Sternberg. ConFunc–functional annotation in the twilight zone. *Bioinformatics*, 24(6):798–806, 2008.

[7] IUBMB. *Enzyme nomenclature: Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes*. Academic Press, New York, 1992.

[8] M. Riley. Functions of the gene products of Escherichia coli. *Microbiol Rev*, 57(4):862–952, 1993.

[9] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–9, 2000.

[10] H. W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Guldener, G. Mannhaupt, M. Munsterkotter, P. Pagel, N. Strack, V. Stumpflen, J. Warfsmann, and A. Ruepp. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res*, 32(Database issue):D41–4, 2004.

[11] C.H. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–58, 2001.

[12] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247(4):536–40, 1995.

[13] C. Z. Cai, L. Y. Han, Z. L. Ji, X. Chen, and Y. Z. Chen. SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Res*, 31(13):3692–7, 2003.

[14] L.Y. Han, C.Z. Cai, Z.L. Ji, Z.W. Cao, J. Cui, and Y.Z. Chen. Predicting functional family of novel enzymes irrespective of sequence similarity: a statistical learning approach. *Nucleic Acids Res*, 32(21):6437–44, 2004.

[15] P. D. Dobson and A. J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *J Mol Biol*, 330(4):771–83, 2003.

[16] I. Melvin, E. Ie, R. Kuang, J. Weston, W. N. Stafford, and C. Leslie. SVM-Fold: a tool for discriminative multi-class protein fold and superfamily recognition. *BMC Bioinformatics*, 8 Suppl 4:S2, 2007.

[17] I. Melvin, J. Weston, C. S. Leslie, and W. S. Noble. Combining classifiers for improved classification of proteins from sequence or structure. *BMC Bioinformatics*, 9:389, 2008.

[18] D. Wieser and N. Mahesan. Remote homology detection using a kernel method that combines sequence and secondary-structure similarity scores. *In Silico Biology*, 31 March 2009.

[19] A. Clare and R. D. King. Predicting gene function in Saccharomyces cerevisiae. *Bioinformatics*, 19 Suppl 2:II42–II49, 2003.

[20] A. Clare, A. Karwath, H. Ougham, and R. D. King. Functional bioinformatics for Arabidopsis thaliana. *Bioinformatics*, 22(9):1130–6, 2006.

[21] L.J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, H.H. Staerfeldt, K. Rapacki, C. Workman, C.A. Andersen, S. Knudsen, A. Krogh, A. Valencia, and S. Brunak. Prediction of human protein function from post-translational modifications and localization features. *J Mol Biol*, 319(5):1257–65, 2002.

[22] I. Melvin, J. Weston, C. Leslie, and W. S. Noble. RANKPROP: a web server for protein remote homology detection. *Bioinformatics*, 25(1):121–2, 2009.

[23] J. E. Gewehr, V. Hintermair, and R. Zimmer. AutoSCOP: automated prediction of SCOP classifications using unique pattern-class mappings. *Bioinformatics*, 23(10):1203–10, 2007.

[24] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[25] S. E. Brenner, P. Koehl, and M. Levitt. The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Res*, 28(1):254–6, 2000.

[26] S F Altschul, T L Madden, A A Schaffer, J Zhang, Z Zhang, W Miller, and D J Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, Sep 1997.

[27] C. H. Wu, R. Apweiler, A. Bairoch, D. A. Natale, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, R. Mazumder, C. O'Donovan, N. Redaschi, and B. Suzek. The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res*, 34(Database issue):D187–91, 2006.

[28] I Dondoshansky. Blastclust (NCBI Software Development Toolkit), 6.1 edition. *NCBI, Bethesda, MD.*, 2002.

[29] L. J. McGuffin, K. Bryson, and D. T. Jones. The PSIPRED protein structure prediction server. *Bioinformatics*, 16(4):404–5, 2000.

[30] Lin Dong, Eibe Frank, and Stefan Kramer. Ensembles of balanced nested dichotomies for multi-class problems. pages 84–95. 2005.

[31] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[32] Chun Yuan Lin, Ken-Li Lin, Chuen-Der Huang, Hsiu-Ming Chang, Chiao Yun Yang, Chin-Teng Lin, Chuan Yi Tang, and D. Frank Hsu. Feature selection and combination criteria for improving predictive accuracy in protein structure classification. In *BIBE '05: Proceedings of the Fifth IEEE Symposium on Bioinformatics and Bioengineering*, pages 311–315, Washington, DC, USA, 2005. IEEE Computer Society.

[33] H. B. Shen and K. C. Chou. Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, 22(14):1717–22, 2006.

[34] M. T. Shamim, M. Anwaruddin, and H. A. Nagarajaram. Support Vector Machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. *Bioinformatics*, 23(24):3320–7, 2007.

[35] T. Damoulas and M. A. Girolami. Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics*, 24(10):1264–70, 2008.