doi:10.1016/j.jmb.2004.10.024 J. Mol. Biol. (2005) 345, 187–199

JMB

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Predicting Enzyme Class From Protein Structure Without Alignments

Paul D. Dobson and Andrew J. Doig*

Department of Biomolecular Sciences, UMIST, P.O. Box 88 Manchester M60 1QD, UK

Methods for predicting protein function from structure are becoming more important as the rate at which structures are solved increases more rapidly than experimental knowledge. As a result, protein structures now frequently lack functional annotations. The majority of methods for predicting protein function are reliant upon identifying a similar protein and transferring its annotations to the query protein. This method fails when a similar protein cannot be identified, or when any similar proteins identified also lack reliable annotations. Here, we describe a method that can assign function from structure without the use of algorithms reliant upon alignments. Using simple attributes that can be calculated from any crystal structure, such as secondary structure content, amino acid propensities, surface properties and ligands, we describe each enzyme in a non-redundant set. The set is split according to Enzyme Classification (EC) number. We combine the predictions of one-class *versus* one-class support vector machine models to make overall assignments of EC number to an accuracy of 35% with the top-ranked prediction, rising to 60% accuracy with the top two ranks. In doing so we demonstrate the utility of simple structural attributes in protein function prediction and shed light on the link between structure and function. We apply our methods to predict the function of every currently unclassified protein in the Protein Data Bank.

Keywords: protein function prediction; structure; EC number; machine learning; structural genomics

*Corresponding author

Introduction

In earlier work[1] we demonstrated that by representing proteins using simple attributes that are easily calculable from any crystal structure, it was possible to predict function as enzymatic or not. The method did not rely on detecting similarity to another protein and could be applied to any protein for which the attributes could be calculated. Here, we extend this idea down the functional hierarchy and address the problem of predicting the enzyme class of a protein that is known or predicted to be an enzyme.

Our capacity to solve protein structures is not being matched by our ability to assign function experimentally. Consequently, many new protein structures lack functional annotations. The number of structures with the annotation "Unknown Function" deposited in the Protein Data Bank (PDB)[2] shows that the frequency of this annotation has nearly trebled on average each year for the past four years, with six structures in 1999, growing to ten in 2000, then leaping to 52 in 2001, followed by a further 81 and 183 in the next two years, respectively. Given that Unknown Function is not the only annotation associated with an undetermined function, this must be considered an underestimate of the number of proteins lacking annotation. Structural genomics projects account for much of this increase, as many of the stages in the structure determination pathway are now automated, so allowing high-throughput pipelines to be constructed.[3] Methods to predict function from sequence and structure are important to fill the gap between the number of structures known and those that have functional annotations,[4] whilst also enabling us to explore links between structure and function.

Function prediction mostly relies on detecting

Abbreviations used: PDB, Protein Data Bank; EC, Enzyme Classification; SVM, support vector machines.

E-mail address of the corresponding author: andrew.doig@umist.ac.uk

similarity between a functionally annotated protein and the query protein, then transferring the annotations across. The method by which the alignment is made can take different forms. Sequence similarity can be detected using such tools as BLAST,[5] FASTA[6,7] and PSI-BLAST.[8] Alternatively, smaller-scale sequence motifs can be searched for, such as those in the PRINTS,[9] BLOCKS,[10] PROSITE[11] and InterPro[12] databases. These motifs are short, conserved sequences that can be indicative of protein function and so are of high utility in the problem of function prediction.

These sequence-based techniques have their structural counterparts. Methods such as Combinatorial Extension[13] and VAST[14] can be used to detect similar structures and folds. The structural counterpart of the sequence motif can be found in databases such as ProCat[15,16] and SPASM[17] that use the spatial arrangement of atoms in protein functional sites to create a template.

Defining protein function is not a simple task and the problem has generated much discussion.[18,19] Here, we adopt definitions from the Enzyme Classification (EC). This scheme dates back to 1956, when the IUBMB began to regularise the naming and categorisation of enzymes. A hierarchical structure has developed with six classes of enzyme at the top level. They are oxidoreductases, transferases, hydrolases, lyases, isomerases and ligases, determined by the general reaction catalysed (Table 1).

Note that this system gives a description of function that is independent of the protein structure and reaction mechanism. As a consequence, enzymes that catalyse the same reaction through different mechanisms are given the same classification, which can lead to structurally and mechanistically different proteins being considered as functionally identical. This lack of context has been the basis for criticisms of the EC system[18] and is a potential source of difficulty in this work. In order to predict enzyme class from structure without creating alignments it is necessary to use structural attributes that capture information pertinent to functional differences. As proteins may be assigned the same class despite being structurally and mechanistically different, then there is a higher level of structural diversity within a class than if definitions of function also incorporated structural context, and so the task of discrimination of function from structure is more difficult.

To return to existing methods for function prediction, when the level of sequence and structural similarity between query and matched protein is very high, the confidence of the predictions is typically also high, but as similarity lessens so confidence diminishes. For pair-wise sequence alignments above 50%, less than 30% share exact (four-digit) EC numbers.[20] Certain proteins lie in such remote regions of fold space that they show no obvious similarity to other proteins. The extent of this has been estimated[21] to be that 5–10% of proteins are orphans (no homologues), 10% have only one homologue, and 30% belong to families with less than ten members. This implies that the probability of detecting similar proteins is low. When coupled to the fact that many of the identified similar proteins also lack function annotations and the previous point regarding the level of function conservation, it becomes apparent that there is a

Table 1. The top level of the Enzyme Classification

Code	Class	Description of Reaction
1	Oxidoreductase	Oxidation or reduction
2	Transferase	Transfer of a chemical group, such as methyl, glycosyl, etc., from substrate to product,
3	Hydrolase	Bond cleavage by hydrolysis (carbon-carbon, carbon-nitrogen, carbon-oxygen, and certain others)
4	Lyase	Elimination of double bonds or ring structures, not by hydrolysis or oxidation
5	Isomerase	Isomeric geometrical or structural changes
6	Ligase	Ligation ('joining together'), typically large of molecules in a reaction coupled to hydrolysis of a pyrophosphate bond in ATP or similar, usually leading to high energy bonds

Summary of the classes at the top level of the Enzyme Classification scheme and the general reactions they mediate.

pressing need for function prediction methods that are independent of similarity.

Similarity-independent prediction methods are found in many diverse and innovative forms.[22] Phylogenetic profiling[23] links the function of proteins with similar expression profiles across different organisms. Gene neighbour methods work by finding proteins that are co-located on a chromosome, as it has been observed that functionally similar proteins often cluster.[24,25] These methods are not reliant on alignments in the usual sense, but do still depend on identified proteins being annotated. There are methods that can make predictions even without annotations. Text data mining applies Natural Language Processing techniques to scientific literature in an attempt to gather functional information.[26] Amino acid composition alone contains a surprising amount of relevant information that has been utilised for function prediction using machine-learning techniques.[27] A whole battery of neural network-based predictions of post-translational modification states, subcellular localisations, isoelectric points, etc. have been successfully combined to predict function.[28] All of these methods work by detecting similarity to other proteins, but in a manner that is fundamentally different to the alignment-based techniques. Instead, similarity is between the query protein and the generalised properties of a functional class of proteins, rather than to a specific protein. The advantage of this approach is that a prediction is possible even if an alignment to an annotated protein cannot be made. Our method falls into this category as it characterises the simple structural properties of each class of enzymes at the top level of the EC scheme. Our initial work was based on the observations made by Stawiski and co-workers,[29] who noted broad structural differences between proteases and non-proteases. These included such properties as high C^{α} density and lower than average surface areas in proteases when compared to non-proteases. By combining multiple weakly discriminating attributes using a neural network-based classifier the function of a protein could be predicted. The extent to which homology played a role in this method is unclear, but we have also shown that grouping proteins by similar function (in our case, two groups; enzymes and non-enzymes) allows us to identify simple structural attributes that differ between groups, and that these differences can be used to predict function. We found that when predicting function as enzymatic or not, attributes such as secondary structure content, cofactor presence and residue fractions (particularly at the surface) were useful. Using a non-redundant subset of the PDB[2] we show that it is possible to predict enzyme class from structure using support vector machines[30,31] with attribute subset selection.[32] The problem is posed as 15 one-class *versus* one-class problems. Using this approach most model accuracies are initially imbalanced and/or incapable of highly accurate prediction due to class size differences, which affect how errors are distributed between classes. To enhance performance each of the 15 models is optimised to give more accurate and balanced results by removing uninformative attributes, so making each problem less complex and the models more robust.

The predictions of the 15 sub-problems were combined using a one-*versus*-rest support vector classification approach (in which a model is built for one class against all enzymes that are not of that class). This approach failed when using the total set of descriptors due to the complicating presence of irrelevant and noisy attributes. Predictions are 35% accurate with the top prediction, and correct to an accuracy of 60% with the top two predictions. This demonstrates that protein function, as far as the first level of the Enzyme Classification, can be predicted from structure without using alignment-based measures. The method is implemented on the www†.

Results

Functional class sizes differ greatly (Table 2). This poses a problem when training classification models. For example, the most imbalanced problem of hydrolases (160 examples in the data set) *versus* ligases (20 examples) achieves an accuracy of 89% simply by predicting hydrolase each time. Class accuracies would then be 100% for hydrolases and 0% for ligases, which is highly imbalanced. This model is based purely upon class sizes and so tells us nothing about how to distinguish between hydrolase and ligase structures. The support vector machine operates by finding a hyperplane that separates two classes in a training set with minimal error (with components to ensure generalisation ability is maintained). The error penalty for each class can be adjusted to reflect the relative size of each class, so that each binary classifier is altered to predict in a balanced manner and so the "predict largest class" model can be avoided. For example, for the hydrolase *versus* ligase problem described earlier, incorrectly predicting what is really a hydrolase adds $20/160 = 0.125$ to the error function,

Table 2. Functional classes and sizes

Functional Class (#)	Class Size
Oxidoreductase (1)	79
Transferase (2)	128
Hydrolase (3)	160
Lyase (4)	60
Isomerase (5)	51
Ligase (6)	20

The set is culled from Astral 1.63.[40,41]

† http://wolf.bi.umist.ac.uk/~mjfikpd2/predict/enon.html

Table 3. 55 Attributes used to describe each protein

Residue Fractions	Surface Fractions	General Properties
Ala	Ala	Number of Residues
Arg	Arg	Total Surface
Asn	Asn	Surface Fractal Dimension
Asp	Asp	Disulphide Bonds
Cys	Cys	Surface Area:Volume
Gln	Gln	**Secondary Structure Content**
Glu	Glu	Helix
Gly	Gly	Sheet
His	His	Turn
Ile	Ile	**Cofactors**
Leu	Leu	ATP
Lys	Lys	NAD
Met	Met	FAD
Phe	Phe	**Metals**
Pro	Pro	Iron
Ser	Ser	Magnesium
Thr	Thr	Copper
Trp	Trp	Calcium
Tyr	Tyr	
Val	Val	

Each attribute is simple to calculate from a PDB file. For more on how each attribute is calculated see Methods.

whereas an error when predicting a ligase costs 160/20=8. Errors in the ligase class cost more than errors in the hydrolase class, so that the best way to minimise the error function is no longer to predict everything as hydrolase.

Each protein is described using the attributes listed in Table 3. All are simple to calculate from any high-quality protein structure.

The multi-class prediction problem is broken down into 15 sub-problems of each EC class against each other. Using all attributes, a complex and reasonably accurate model can be constructed in most cases. Subsets of attributes can simplify each problem, however. Backwards elimination search techniques generate more accurate and simpler solutions (Table 4). For instance, in the first row the problem is oxidoreductase *versus* transferase. After subset selection only 11 of 55 attributes remained, yet they were sufficient to build a model with a total accuracy of 66.2%. Class accuracies are well balanced; with oxidoreductases being predicted correctly 68.4% of the time and the transferase class 64.8% of the time. This is not true in all cases, particularly for the problems of transferase *versus* lyase and hydrolase *versus* lyase. This is due to the scoring function benefiting more from achieving high accuracy with poor balance, than attaining good balance at the expense of accuracy. What this says about the underlying distributions and biological relationships is unclear, though it may suggest that for these problems the set of attributes did not contain many strong classifiers.

The final subsets of attributes for each sub-problem are shown in Tables 5–10. Grey squares are those attributes in the subset; white are those that have been excluded. Noteworthy properties of the subsets include the high utility of iron for problems involving oxidoreductases, the high usage of surface composition data, and the very large number of attributes required to construct models for ligases. This could be due to the difficulty of learning with a very small set of data, though it is also possible that it reflects the high complexity of ligase function, in which the ligation of two (typically large) molecules occurs whilst ATP is hydrolysed.

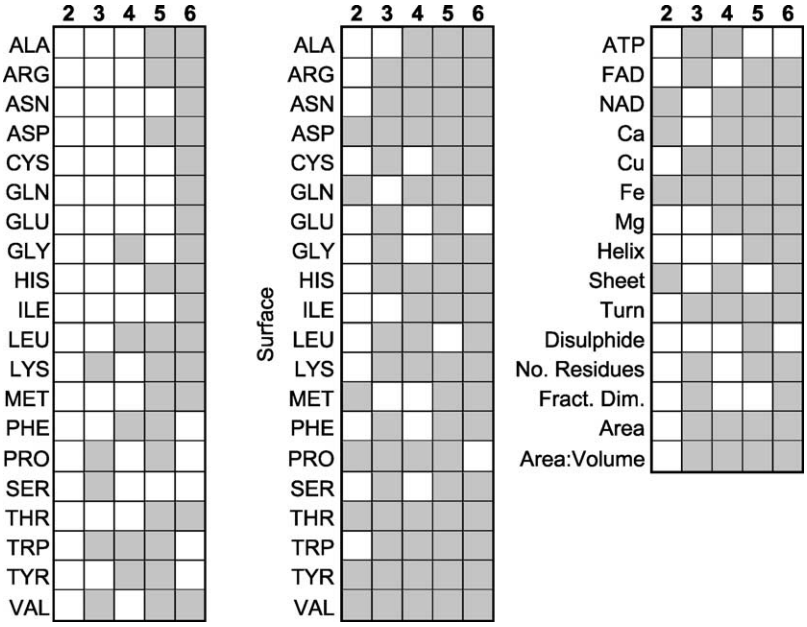
The percentage of correct predictions in each rank is shown in Table 11. As a new version of Astral became available during this work the function of new structures (not already present in the database) was predicted. Though this extra set was very small (117), comparable performance was achieved (values in parentheses in Table 11). This emphasises

Table 4. Sub-problem performance and number of attributes

Sub-problem		Accuracy (%)			
A	B	A	B	Total	Attributes
1	2	68.4	64.8	66.2	11
1	3	79.7	61.3	67.4	30
1	4	75.9	75.0	75.5	29
1	5	73.4	74.5	73.8	44
1	6	81.0	75.0	79.8	46
2	3	58.6	58.8	58.7	15
2	4	35.9	75.0	48.4	25
2	5	53.9	66.7	57.5	19
2	6	59.4	75.0	61.4	54
3	4	46.2	78.3	55.0	13
3	5	58.8	68.6	61.1	12
3	6	49.4	70.0	51.7	31
4	5	50.0	68.6	58.6	26
4	6	50.0	60.0	52.5	37
5	6	62.7	70.0	64.8	43

The performance of each one-class *versus* one-class sub-problem, with the number of attributes in the optimal model after attribute selection.

Table 5. Attribute selection map for oxidoreductases



Grey squares are attributes within the subset; white are excluded attributes.

the generalising ability of the method, showing that over-fitting was avoided.

Further to this, we applied our method to predict function for the 222 proteins in the PDB with the annotation Unknown Function. First function is predicted as enzymatic or not, and then enzyme class is predicted (even for those predicted as non-enzyme). Class numbers are shown in rank order. For this set, we predict that 164 are enzymes and 58 are non-enzymes. Within the enzymes, we predict 11 oxidoreductases, 21 transferases, 120 hydrolyases, five lyases, five isomerases and two ligases. These predictions are available in the Supplementary Material. A literature search was carried out to try to ascertain broad functions for these proteins, either from more recent literature than the PDB annotations, or by inference from indirect sources. Of 222 proteins, 150 have no obvious known function. The majority of these are the products of structural genomics projects, demonstrating how the assignment of function is falling behind the determination of structure. Of the remaining 72, 61 are probably enzymes (of which we predict 54 as such), and 11 are non-enzymes (we predict six as

Table 6. Attribute selection map for transferases

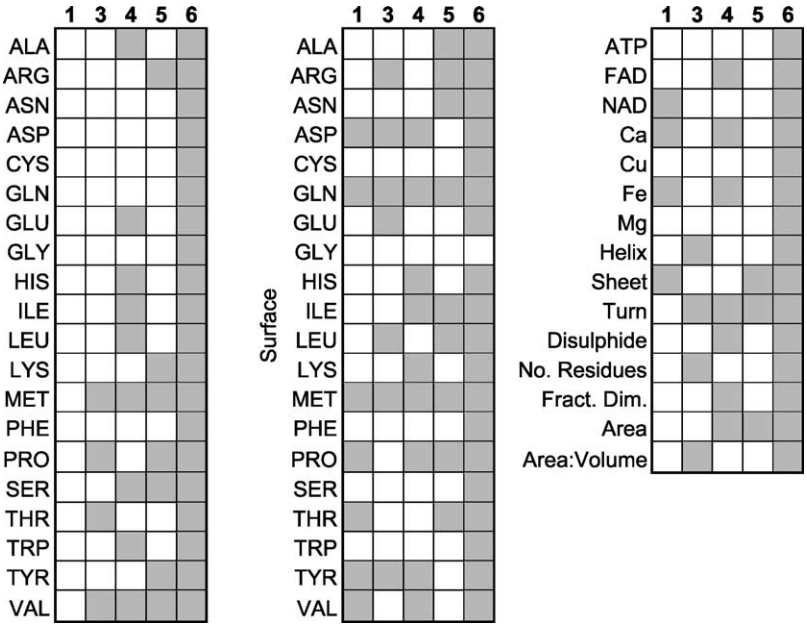
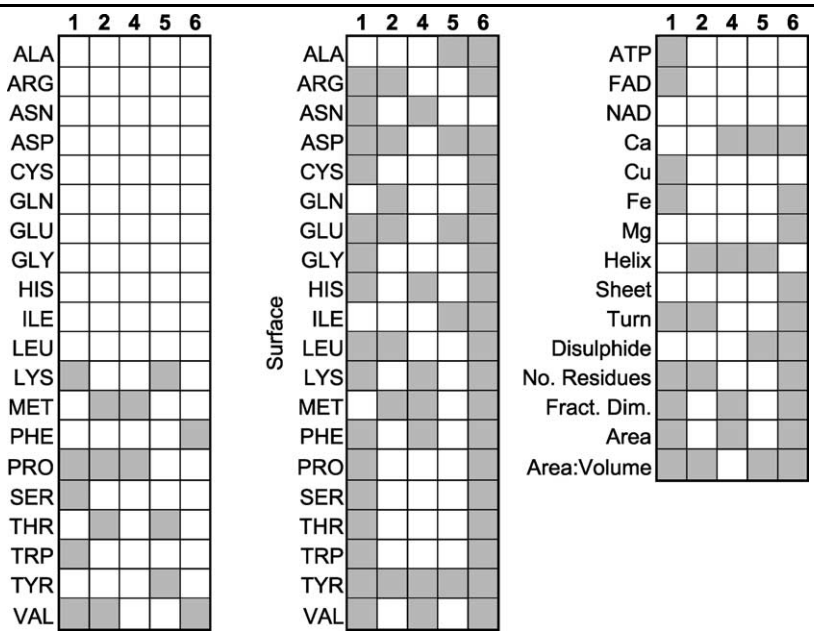


Table 7. Attribute selection map for hydrolases



such). Probable enzyme class assignments were possible for 56 enzymes, 16 of which we predict correctly with the top-ranked prediction, corresponding to 29% accuracy. Including the next ranked prediction increases accuracy to 61% (34 correct), followed by 82% (46 correct) with the third ranked prediction.

The accuracy of each rank can be broken down further by class (Figure 1). It can be seen that despite individual models being highly accurate and well balanced, combination by the multi-class support vector machine still leads to larger classes dominating the top ranked predictions. Ligases in particular suffer here, though successes are all in the first rank.

Discussion

We have developed a system for predicting the function of a protein from its structure even when an alignment to an annotated protein cannot be made. In doing so we have demonstrated the utility of simple attributes of protein structure in protein function prediction. It is not our intention to

Table 8. Attribute selection map for lyases

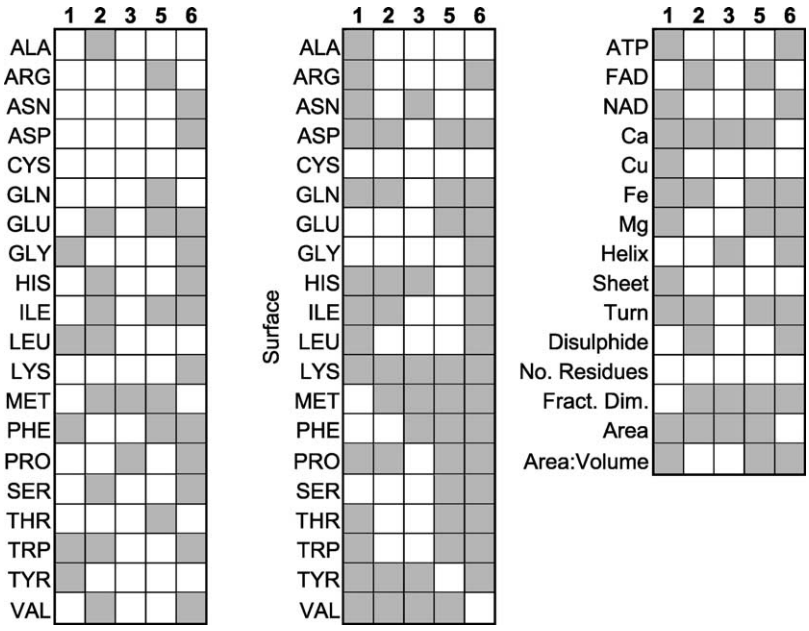
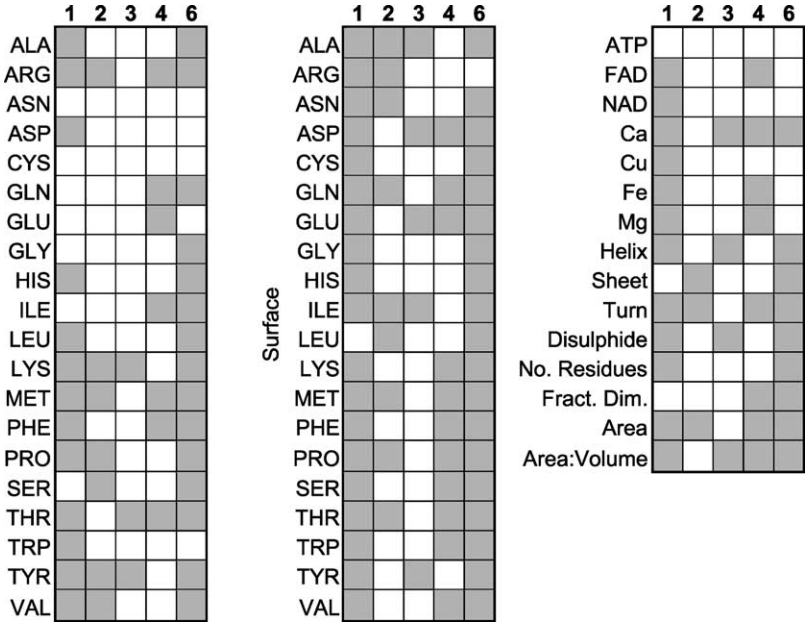


Table 9. Attribute selection map for isomerases



compete with alignment-based methods or match the performance of alignment-independent sequence-based classifiers. Our goal is simply to show that simple structural attributes can be used to predict function and so provide an alternative when these approaches fail.

It is important to be able to predict protein function from structure to fully exploit the information being generated by structural genomics projects. Experimentally determining function can be difficult and often expensive, so target selection by function prediction is important. Here, we provide a tool for aiding this even in the very remote regions of fold space, where there are currently no predictors that incorporate structural information. Further to this, the approach lets us explore the relationship between structure and more broad definitions of protein function.

The method works by generalising the structural features of proteins that share EC numbers. Each protein in a non-redundant training set is described using simple attributes, such as residue fractions, surface properties, secondary structure fractions and ligands. With this as input we use a supervised

Table 10. Attribute selection map for ligases

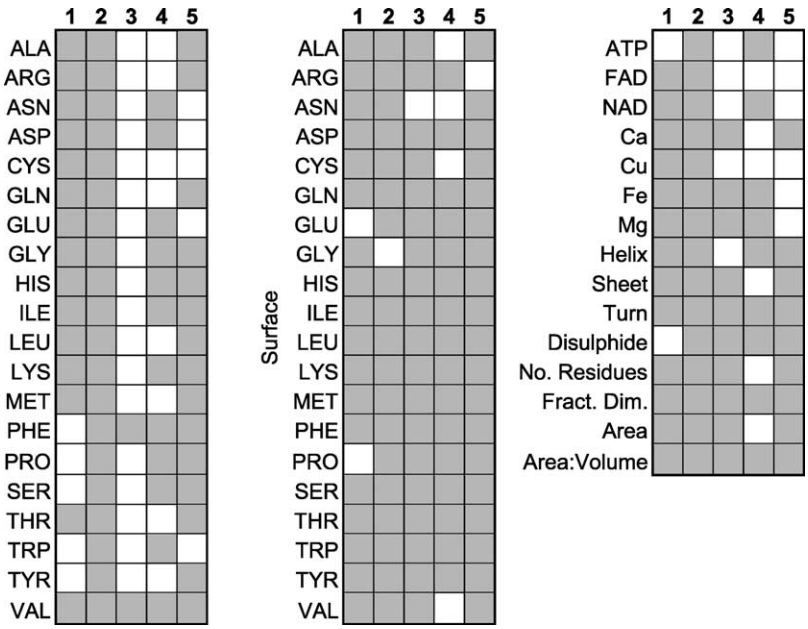


Table 11. Rank accuracies

Rank	Correct	Accuracy (%)	Cumulative Accuracy (%)
1	176(33)	35(28)	35(28)
2	126(30)	25(26)	60(54)
3	81(21)	16(18)	77(72)
4	45(15)	9(13)	86(85)
5	50(13)	10(11)	96(96)
6	20(5)	4(4)	100(100)

For each query protein the potential classes are ranked. Accuracies in parentheses are for the new set of proteins culled from Astral 1.65, but not present in Astral 1.63. The accuracy of a rank is the number of times the prediction is correct for that rank. For the top rank 35% of predictions are correct. Second rank predictions are 25% accurate. With the top two ranks the cumulative accuracy is 60%.

learning technique, the support vector machine, to build binary classifiers that discriminate between enzyme classes in a one-class *versus* one-class manner. Though this approach requires more models to be constructed than the alternative one-class *versus* other-classes adaptation of binary classifiers to multi-class problems, each individual model is, in theory at least, more simplistic and has less difficulty with imbalance between class sizes.

To optimise each sub-problem attribute selection by backwards elimination was used. This gave simpler, better-balanced and more accurate results. That the removal of attributes can lead to improved results may at first seem counter-intuitive. It must be considered that certain attributes may not be pertinent to a problem and only serve to complicate each model. When removed the problem can be freer from noise, as it contains only useful data. This increased simplicity permits better separation and generalisation. To explore useful combinations of attributes it is possible to adopt two different approaches. The "pre-filter" methods rely on identifying potentially discriminating attributes prior to constructing the machine learning classifier, typically by using some statistical measure of the information content of attributes, such as their correlation coefficients. In theory, two well-correlated attributes contain much the same information and so the presence of both is not necessary. Our preference is for on-line, "wrapper" methods, which employ the support vector machine within the backwards elimination algorithm. This allows us to more subtly explore the utility of attributes as they interact in the context of the classification algorithm, which permits a more relevant selection of attributes.

We endeavour to ensure that our methods do not sacrifice generalisation in order to achieve accuracy. By running a validation set in parallel during attribute subset selection we can guarantee that models are capable of predicting unseen data.

The imbalance caused by class size differences is handled by weighting the error penalty during model training according to class ratios, and by incorporating the same ratios into the backwards elimination scoring function. Despite this, class imbalance remains a problem and a valid goal for structural genomics projects must include structures for lyases, isomerase and particularly ligases, to gain more insight into the function of these less abundant, but functionally significant proteins.

The manner in which the set was culled from protein structure databases introduces a level of difficulty greater still than that in our previous

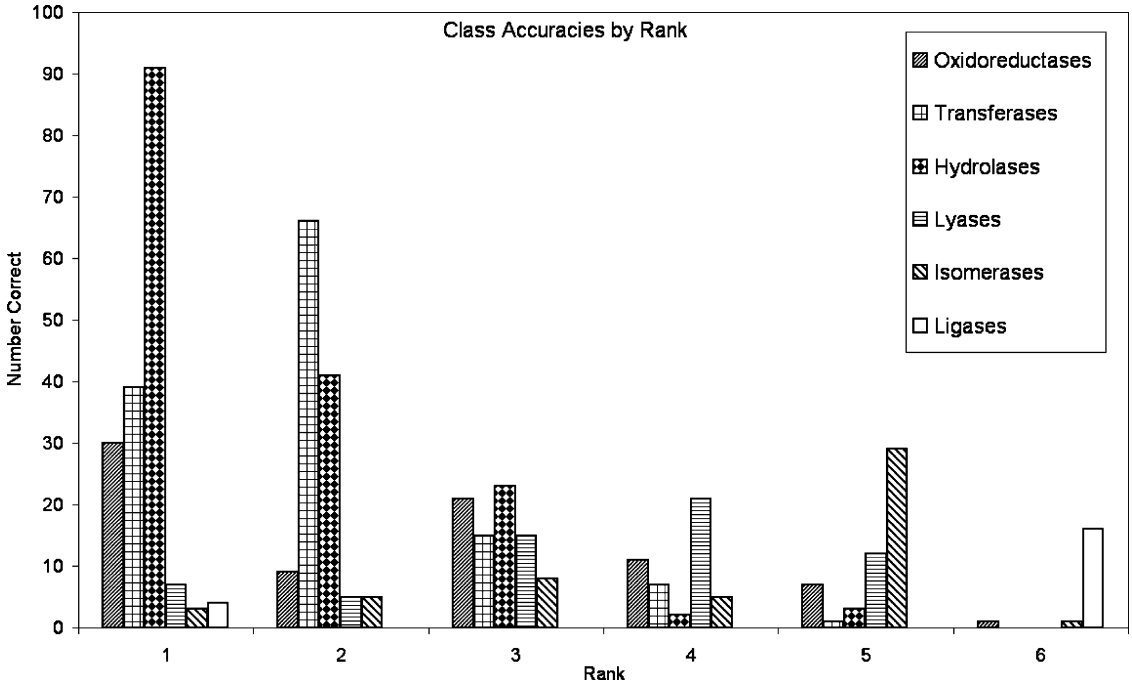


Figure 1.

work. In earlier investigations we adopted the standard approach of culling a non-redundant set of proteins and then separating into enzymes and non-enzymes. This represents a random subset of data in which homology cannot play a role. Here, we deliberately incorporate structural similarity outside of the functional class in order to maximise dataset size and better represent the full range of structures in each functional class. This we achieve by first splitting the data and then removing redundancy, rather than removing redundancy and then splitting into classes. Given two structurally similar proteins with different functions, the first approach only allows one of the proteins to be present in one functional class or the other. The second approach allows the presence of both proteins, representing both functional classes more fully. This means that each model must discriminate between functional classes even if they contain some broad structural similarity. Better coverage of the range of possible structures makes the problem significantly more difficult, but the results more informative.

While the main goal of this project is to develop methods to predict protein function, it should be considered that to demonstrate the utility of structural information we have restricted ourselves to attributes available from structures (and only if these are rapidly calculable for the purposes of making the approach practical). Attributes calculable from sequence have been deliberately excluded as this information has already been shown to be of great use.[28] If the results of this work are combined with many sequence-based attributes, then the resulting classifier should be of greater strength.

The complexity of generalising the properties of a non-redundant set of proteins that share a function is such that it is advisable to first seek to use methods reliant on high-level similarity, such as sequence and structure alignments. Even though recent papers have advised that this is not always a secure method,[20] when similarity is high the confidence and precision of the predicted function is often greater.

It is apparent that discovering gross-structural attributes to discriminate between functional classes of proteins is difficult and only through the interaction of many weakly differentiating attributes can stronger classifiers be constructed. Consequently it is challenging to deconstruct models and rationalise how they operate. Certain explanations are possible; such as the presence of iron is a very strong indicator that the enzyme is an oxidoreductase. All models involving the oxidoreductase class use iron and when the raw data are examined it is clear why. Of the 33 enzymes containing iron, 27 are oxidoreductases. Iron can have a role in redox chemistry, which may explain the attribute's usefulness.

Another example would be the high utility of the fraction of surface tyrosine in models involving hydrolases. On average, surface tyrosine is marginally higher in hydrolases than in other classes, though perhaps more tellingly, a subset of hydrolases contains particularly high fractions of surface tyrosine. Taking a cut-off of 5% of the surface being attributable to tyrosine, which corresponds to approximately one standard deviation from the 3% mean, then 30% of the proteins (normalised by class size) above this are hydrolases, which is higher than for other classes. Therefore, for a subset of hydrolases, surface tyrosine fractions are strong classifiers.

However, these sorts of deconstructions are the exception rather than the rule. For most attributes it is not clear how they are contributing to discrimination, though it is possible to speculate on how they might operate. For example, generally surface residue fractions are more frequently used than overall residue fractions. This could reflect a number of properties of the protein surface, such as localisation (many glycosylation sites for extracellular proteins, hydrophobic residues for membrane proteins, etc.), or the interaction preferences of various types of protein (those involved in binding nucleic acids might require large charged patches on the protein surface). There may also be more specific information from the protein surface contributing to discrimination, particularly for low abundance residues. As an example of this, the protein with the highest proportion of surface tryptophan, PDB 1H8G, EC 3.5.1.28, has 10% of its surface accounted for by this residue type (compared with an average of 1.2% across the whole set). These tryptophan residues are parts of hydrophobic pockets that cover the protein in order to bind choline,[33] demonstrating how rare residues can be tolerated if they are in a functional role.

The set of attributes represents various aspects of the whole of the protein structure, yet the most important region of an enzyme as regards function should be the reaction centre. At present active site identification is, for the most part, only possible using methods such as ProCat[15,16] and SPASM.[17] These approaches are based on motif-identification and so do not belong in the category of alignment-independent prediction algorithms. Many alternative approaches to active site identification are being explored[34–39] that, if sufficiently accurate, will permit a whole new set of attributes to describe functional site differences between classes of enzyme.

Two related approaches to function prediction have been developed that operate on sequence-based attributes. We have already mentioned the excellent work by Jensen *et al.*,[28] in which a neural network approach combines predictions of post-translational modification states, subcellular localisations and other such information into function predictions. Similar to this is the work by Cai *et al.*,[40] in which function is predicted using support vector machines. Quoted performances are highly accurate, with predictions being made to the second level of the EC hierarchy. However, the authors note the influence of sequence similarity ("our study seems to suggest that sequence distance has certain

(*sic*) level of influence on the accuracy of SVM classification", page 71). This approach seemingly gains much of its performance accuracy from redundancy within the data set and so does not entirely fall into the category of alignment-independent predictive methods. For this to be the case performance must be demonstrated on a less redundant data set.

A brief précis of our prediction method is as follows: we use gross structural attributes to describe proteins in a non-redundant subset and generate accurate, balanced models for each binary sub-problem. We combine these results using a multi-class SVM into a function prediction method that performs significantly better than random (22.1%, see Methods).

It is difficult to estimate the performance of the multi-class support vector machine to give a total accuracy to the approach. Internal SVM parameters require tuning to optimise performance, but the figure often quoted in the literature as the overall performance accuracy for multi-class problems is the result of optimising to maximise total accuracy. This is a rather unprincipled approach as it can result in models that are biased and not well balanced. Here, this approach would select models that are 45% accurate, yet closer analysis of the results reveals that all accuracies are concentrated in the three largest classes. To optimise the multi-class model we adopt the strategy of penalising a scoring function to an extent related to the relative class size, much as in the backwards elimination and sub-problem parameter tuning. In this case total prediction accuracy is 35%. This increases to 60% in the top two ranks, and 77% with the top three. There is some remaining bias, with the top rank prediction gaining most of its performance accuracy from the larger classes, and with the isomerase class only being 6% accurate. Better multi-class models for predicting isomerases can be achieved by altering parameters, but at the expense of other classes.

The "correct" accuracy for a multi-class problem is really dependent upon the system for combining predictions. Using our approach breaks the problem down into one-class *versus* one-class models, optimises subsets and parameters, and then makes predictions for all proteins. By doing this we can generate accurate and robust predictors of protein function from structure.

Methods

Data set

The dataset is constructed using function definitions obtained from DBGet[41] PDB Enzyme[42] cross-links and structural relations from the Astral SCOP 1.63 superfamily level dataset.[43,44] In each functional class no structure contains a domain from the same superfamily as any other structure. Within a functional class there is therefore no similarity greater than or equal to the superfamily level. Across classes, domains from certain superfamilies can be present more than once. This is in order to represent the full range of protein structures within a functional class and also force the method to address the problem of predicting protein function correctly, even for broadly similar structures. The Astral lists were culled so that only whole protein structures with a SPACI score of 0.3 or greater could be selected for each functional class.[43] The SPACI score combines various measures of structural quality into one value. Structures with low SPACI scores are excluded to maintain accuracy within the set. In an attempt to preserve biologically significant attributes the PDB "biological units" were used.[2] This is a system whereby the PDB files have been converted into what is considered to be their functional form (e.g. monomer, dimer, etc.).

The clustering strategy for building the set was as follows: for each functional class find all proteins with a SPACI score[43] 0.3 or above. For one functional class a protein is chosen from the set and all the superfamily domains it contains are identified. Each of these superfamilies is now represented in the class, so all proteins that contain domains from the same superfamilies are now eliminated from future selections. When the next protein is chosen it is from a reduced set that does not contain domains from any of the superfamilies previously selected. This process is repeated until there are no further selections to make. In this way we can choose a set of proteins that covers the range of superfamilies within a functional class. This is done for each functional class. It is not possible for domains from the same superfamily to be present more than once within a functional class, but it is possible for domains from the same superfamily to be present in more than one functional class.

During this project Astral upgraded to version 1.65. From the new entries a further validation set was culled in a similar manner.

Attributes for model building

Attributes used for describing each protein are deliberately simple and rapid to calculate. Size is the number of amino acid residues in the protein. Residue preference is simply the number of each residue type in a protein divided by the total number of residues.

Surface residue preference is the total surface area attributable to each residue type divided by the total surface area (also an attribute), as calculated by NACCESS.† Another surface-based attribute is the fractal dimension. This is calculated as by Stawiski *et al.*,[29] by calculating the gradient of the log–log plot of probe radius against molecular surface (calculated by MSMS[45]). Fractal dimension might be thought of as representing the "crinkliness" of the protein's surface. This attribute is included to attempt to capture mid-to-small scale variations in the protein surface topology. Larger scale variations, such as deep surface pockets, should be captured by the surface area to volume ratio. This goes some way to incorporating information on large invaginations on the protein surface, but also may reflect unusually shaped proteins.

Secondary structure contents are derived from the Stride[46] assignments of helix (α, 3_{10} or π), sheet and turn. Heterogen and metal data are taken from the PDB file HETNAM records and are presented in binary form (1 for present, 0 for not present). As cofactor analogues are often used in crystallography we accept the following codes to

† http://wolf.bms.umist.ac.uk/naccess

represent ATP: AMP, A. ADP and ATP. FAD is represented by the codes FAD or FDA. We take NAD, NAH, NAP and NDP to represent the presence of NAD. It is evident that this system could miss many analogues, but in earlier work this was found to be the best way to incorporate heterogen information without introducing excess noise (many analogues that are equivalent in the databases of heterogens are in fact not equal and so introduce considerable error).

Support vector machines

The support vector machine algorithm[30,47] is essentially a binary classifier, though it can be extended to handle multiple classes (see Combining classifiers with multi-class SVMs, later). It separates two classes described by *n*-attributes (in *n*-dimensional space) in a way that minimises error without over-fitting to the data.[31] This is important to ensure high performance on unseen data. The separation is achieved by the segregation of the space into half-spaces with a linear hyperplane so that each half-space corresponds to a class label. Query sample orientation relative to this hyperplane gives the predicted class. Most real problems do not have simple, linear solutions. To address this a kernel function can be used to map the data into a higher dimensional space where a linear separation is feasible. Support vector machines frequently out-perform other machine-learning methods of choice and are particularly useful for noisy data.

The implementation of the algorithm used here is LIBSVM.† The C-SVC type machine was used with a radial basis function kernel. Two internal parameters, "cost" and "γ" (error penalty and kernel function variables), are optimised by grid searching. This optimisation is performed prior to attribute subset selection, as it is too computationally expensive to perform for each subset generated during selection. Classes are weighted according to their relative sizes. All other parameters are run on default settings. All non-binary attributes are normalised to be in the range 0 to 1.

Validation is performed using leave-one-out testing. Each protein takes a turn being the query protein, with the model being built on the remaining proteins in the set. The query protein then has its class predicted. The sum of correct predictions divided by total number of queries estimates the whole model accuracy.[48]

Backwards elimination for attribute subset selection

The full set of attributes used to describe each protein may not be optimal. Certain attributes contain little or no information relevant to the classification task. Noisy data make a problem more complex and often its removal can lead to more simple, easy to interpret and better performing models.[32] Choosing which attributes to keep and which to discard is not a trivial problem. As the number of attributes grows the problem rapidly becomes highly complex. For N attributes we have 2^N-1 possible subsets of attributes (ignoring the all-absent option). Here, we use a backwards elimination approach to select better performing subsets. The basic idea of backwards elimination is to assess performance using a set of attributes, and then eliminate each attribute from the set one at a time. The attribute that gives the greatest performance gain upon its elimination is permanently removed from the set. This process is repeated until no further enhancement occurs.

The most simplistic measure of performance is to use the total accuracy. For problems with unequal class sizes this can lead to bias within the model. To avoid this a scoring function can be used that takes into account class sizes. The same error penalties that are used in weighting the support vector machine (the class size ratios) can be used to construct a score for each model. For classes A and B, where the size of class A is twice that of B, an error in B adds double an error in A to the final score. Choosing the lowest score is then equivalent to selecting that model that gives most optimal balanced performance. This same scoring function is used for tuning C and γ.

During selection it is necessary to partition the training set further, so that a model can be built and its performance assessed. The key idea behind machine-learning algorithms of this type is that the data used to build the model reflects the real underlying distribution. The model should be able to predict any sample drawn from the same distribution. For small data sets the partition can lead to different approximations of the real distribution (as it is difficult to approximate a distribution from only a few samples). The average accuracy of multiple random partitions of the same training data gives a performance estimate that better reflects the utility of the attributes, and which is not so dependent upon the partition, than if the same partition is used throughout selection process.

It is possible that the subset of attributes can begin to describe only the data in the training set. To avoid this we partition the data into thirds and use two thirds, the training set, to optimise the algorithm. The remaining third, the validation set, we run in parallel to score the top subset at the end of each round of elimination. The performance on the validation set in no way influences the selection of the attribute to be eliminated. If over-fitting to the training set occurs the performance on the validation set begins to degrade. By tracking the validation set performance we can choose the stopping point of the elimination and so pick the most optimal and best generalising model. Taking such measures is an essential stage in the construction of robust models that avoid over-fitting and it is often neglected, with the consequence that many performance accuracies quoted in the literature are only applicable to the training set and do not truly reflect predictive accuracy.

Combining classifiers with multi-class SVMs

Support vector machines have been adapted to handle multi-class data in a number of ways.[49] Most methods involve reducing the multi-class problem down to binary problems. The one-*versus*-all approach involves constructing models for one class against all others. To predict enzyme function one might build the model "oxidoreductases" *versus* "not-oxidoreductases". Initial trials using this approach to predict enzyme class from the attributes failed as a result of the large difference in class sizes (despite weightings) and the complexity of the problem. To make each sub-problem simpler it is possible to construct models for one class against each other class (e.g. oxidoreductases *versus* transferases, hydrolases *versus* ligases, etc.). For a six-class problem 15 models are required. Performance on all attributes using the multi-class LIBSVM has a maximum performance accuracy of 34%, though this model is simplistic in that it concentrates on the largest classes only, with almost all the correct predictions being found in the hydrolase class.

† http://www.csie.ntu.edu.tw/~cjlin/libsvm

This is not markedly different from the model Always Predict Hydrolase, which is 32% accurate simply because of the large number of hydrolases. For this reason it was necessary to break down the enzyme class problem, tune each model and recombine results. By outputting the predictions of each of the 15 models it is possible to represent their results as 15-component vector. The standard multi-class SVM approach built into LIBSVM can then handle this much cleaner and better-optimised problem. However, there are still issues concerning parameter tuning that can lead to great variability between classes in terms of accuracy. Often figures quoted as total accuracies in the literature are misleading, as they are not explicit about how errors are distributed between classes. Selection of support vector machine parameters can greatly alter this, with a grid search of C and γ on the multi-class problem here able to swing from high total accuracy by concentrating on larger classes, to low total accuracy but with improved performance on smaller classes. Here, we adapt the scoring function described earlier to the multi-class problem, so that the error penalty reflects relative class sizes.

Our approach allows us to rank the six possible classes in likelihood order. The initial multi-class SVM uses all sub-problem results to predict a class. This predicted class is then excluded from training of the subsequent SVM that predicts the next class. The process is repeated until all options are exhausted.

Random performance we take as the sum of the squares of class size divided by the total set size. For the problem of predicting enzyme class when it is known that the protein is an enzyme, random is 22.1%.

Acknowledgements

This work was funded by a BBSRC Engineering and Biological Systems Committee studentship.

Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.jmb.2004.10.024

References

1. Dobson, P. D. & Doig, A. J. (2003). Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.* **330**, 771–783.
2. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H. *et al*. (2000). The Protein Data Bank. *Nucl. Acids Res.* **28**, 235–242.
3. Brenner, S. E. (2001). A tour of structural genomics. *Nature Genet.* **2**, 801.
4. Rost, B., Liu, J., Nair, R., Wrzeszczynski, K. O. & Ofran, Y. (2003). Automatic prediction of protein function. *Cell. Mol. Life Sci.* **60**, 2637–2650.
5. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410.
6. Pearson, W. R. (1990). Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol.* **183**, 63–98.
7. Pearson, W. R. & Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.
8. Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.* **25**, 3389–3402.
9. Attwood, T. K. (2002). The PRINTS database: a resource for identification of protein families. *Briefings Bioinformatics*, **3**, 252–263.
10. Henikoff, S., Henikoff, J. G. & Pietrokovski, S. (1999). Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics*, **15**, 471–479.
11. Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C. J., Hofmann, K. & Bairoch, A. (2002). The PROSITE database, its status in 2002. *Nucl. Acids Res.* **30**, 235–238.
12. Mulder, N. J., Apweiler, R., Attwood, T. K., Bairoch, A., Barrell, D., Bateman, A. *et al*. (2003). The InterPro Database, 2003 brings increased coverage and new features. *Nucl. Acids Res.* **31**, 315–318.
13. Shindyalov, I. N. & Bourne, P. E. (1998). Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* **11**, 739–747.
14. Gibrat, J.-F., Madej, T. & Bryant, S. H. (1996). Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.* **6**, 377–385.
15. Wallace, A. C., Laskowski, R. & Thornton, J. M. (1996). Derivation of 3D coordinate templates for searching structural databases: application to the Ser-His-Asp catalytic triads of the serine proteinases and lipases. *Protein Sci.* **5**, 1001–1013.
16. Wallace, A. C., Borkakoti, N. & Thornton, J. M. (1997). TESS: a geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases: application to enzyme active sites. *Protein Sci.* **6**, 2308–2323.
17. Kleywegt, G. J. (1997). Validation of protein models from Cα coordinates alone. *J. Mol. Biol.* **273**, 371–376.
18. Babbitt, P. C. (2003). Definitions of enzyme function for the structural genomics era. *Curr. Opin. Chem. Biol.* **7**, 230–237.
19. Shrager, J. (2003). The fiction of function. *Bioinformatics*, **19**, 1934–1936.
20. Rost, B. (2002). Enzyme function less conserved than anticipated. *J. Mol. Biol.* **318**, 595–608.
21. Liu, J. & Rost, B. (2001). Comparing function and structure between proteomes. *Protein Sci.* **10**, 1970–1979.
22. Dobson, P. D., Cai, Y., Stapley, B. J. & Doig, A. J. (2004). Prediction of protein function in the absence of significant sequence similarity. *Curr. Med. Chem.* **11**, 2135–2142.
23. Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D. & Yeates, T. O. (1999). Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl Acad. Sci. USA*, **96**, 4285–4288.
24. Dandekar, T., Snel, B., Huynen, M. & Bork, P. (1999). Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.* **23**, 324–328.
25. Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D. & Maltsev, N. (1999). The use of gene clusters to infer functional coupling. *Proc. Natl Acad. Sci. USA*, **96**, 2896–2901.

26. Stapley, B. J., Kelley, L. A. & Sternberg, M. J. E. (2002). Predicting the sub-cellular location of proteins from text using support vector machines. *Pac. Symp. Biocomput.* **7**, 374–385.
27. Cai, Y., Liu, X. & Chou, K. (2002). Artificial neural network model for predicting protein subcellular location. *Comput. Chem.* **26**, 179–182.
28. Jensen, L. J., Gupta, R., Blom, N., Devos, D., Tamames, J., Kesmir, C. *et al*. (2002). Prediction of human protein function from post-translational modifications and localization features. *J. Mol. Biol.* **319**, 1257–1265.
29. Stawiski, E. W., Baucom, A. E., Lohr, S. C. & Gregoret, L. M. (2000). Predicting protein function from structure: unique structural features of proteases. *Proc. Natl Acad. Sci. USA*, **97**, 3954–3958.
30. Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK.
31. Vapnik, V. (1999). *The Nature of Statistical Learning Theory*, Springer, New York.
32. John, G., Kohavi, R. & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning*: *Proceedings of the Eleventh International Conference*, pp. 121–129, AAAI, Morgan Kaufmann San Francisco, CA.
33. Fernandez-Tornero, C., Lopez, R., Garcia, E., Gimenez-Gallego, G. & Romero, A. (2001). A novel solenoid fold in the cell wall anchoring domain of the pneumococcal virulence factor LytA. *Nature Struct. Biol.* **8**, 1020–1024.
34. Ondrechen, M. J., Clifton, J. G. & Ringe, D. (2001). THEMATICS: a simple computational predictor of enzyme function from structure. *Proc. Natl Acad. Sci. USA*, **98**, 12473–12478.
35. Gutteridge, A., Bartlett, G. J. & Thornton, J. M. (2003). Using a neural network and spatial clustering to predict the location of active sites in enzymes. *J. Mol. Biol.* **330**, 719–734.
36. Bartlett, G. J., Porter, C. T., Borkakoti, N. & Thornton, J. M. (2002). Analysis of catalytic residues in enzyme active sites. *J. Mol. Biol.* **324**, 105–121.
37. Porter, C. T., Bartlett, G. J. & Thornton, J. M. (2004). The catalytic site atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucl. Acids Res.* **32**, D129–D133.
38. Jones, S. & Thornton, J. M. (2004). Searching for functional sites in protein structures. *Curr. Opin. Chem. Biol.* **8**, 3–7.
39. Bate, P. & Warwicker, J. (2004). Enzyme/non-enzyme discrimination and prediction of enzyme active site location using charge-based methods. *J. Mol. Biol.* In the press.
40. Cai, C. Z., Han, L. Y., Ji, Z. L. & Chen, Y. Z. (2004). Enzyme family classification by support vector machines. *Proteins: Struct. Funct. Bioinformatics*, **55**, 66–76.
41. Fujibuchi, W., Goto, S., Migimatsu, H., Uchiyama, I., Ogiwara, A., Akiyama, Y. & Kanehisa, M. (1997). DBGET/LinkDB: an integrated database retrieval system. *Pac. Symp. Biocomput.* **3**, 681–692.
42. Bairoch, A. (2000). The ENZYME database in 2000. *Nucl. Acids Res.* **28**, 304–305.
43. Brenner, S. E., Koehl, P. & Levitt, M. (2000). The ASTRAL compendium for sequence and structure analysis. *Nucl. Acids Res.* **28**, 254–256.
44. Chandonia, J. M., Walker, N. S., Conte, L. L., Koehl, P., Levitt, M. & Brenner, S. E. (2002). ASTRAL compendium enhancements. *Nucl. Acids Res.* **30**, 260–263.
45. Sanner, M. F., Spehner, J.-C. & Olson, A. J. (1996). Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, **38**, 305–320.
46. Frishman, D. & Argos, P. (1995). Stride: knowledge-based protein secondary structure assignment. *Proteins: Struct. Funct. Genet.* **23**, 566–579.
47. Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Min. Knowledge Discov.* **2**, 121–167.
48. Bishop, C. M. (1995). Section 9.8.1. Cross validation. In *Neural Networks for Pattern Recognition*, pp. 372–375, OUP, Oxford.
49. Wu, T.-F., Lin, C.-J. & Weng, R. C. (2003). Probability estimates for multi-class classification by pairwise coupling. In *Advances in Neural Information Processing Systems* (Thrun, S., Saul, L. & Scholkopf, B., eds), MIT Press, Cambridge, MA.

Edited by J. Thornton

(Received 30 June 2004; received in revised form 25 August 2004; accepted 12 October 2004)