Bioinformatics Advance Access published February 19, 2004

Prediction of *Saccharomyces Cerevisiae* Protein Functional Class from Functional Domain Composition

Yudong Cai & Andrew J. Doig*

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

* To whom correspondence should be addressed: Phone: +44-161-2004224, Fax: +44-161-2360409, e-mail: Andrew.Doig@umist.ac.uk

Running title: Protein Function Prediction by Domain Composition

Key Words: yeast, functional genomics, Support Vector Machines, InterPro, nearest neighbour; InterPro

Abbreviations:NNA, Nearest Neighbour Algorithm; SVM, Support Vector Machine; ORF, Open Reading Frame

ABSTRACT

Motivation: A key goal of genomics is to assign function to genes, especially for orphan sequences.

Results: We compared the clustered functional domains in the SBASE database to each protein sequence using BLASTP. This representation for a protein is a vector, where each of the non-zero entries in the vector indicates a significant match between the sequence of interest and the SBASE domain. The machine learning methods Nearest Neighbour Algorithm and Support Vector Machines are used for predicting protein functional classes from this information. We find that the best results are found using the SBASE-A database and the Nearest Neighbour Algorithm, namely 72% accuracy for 79% coverage. We tested assigning function based on searching for InterPro sequence motifs and by taking the most significant BLAST match within the dataset. We applied the functional domain composition method to predict the functional class of 2018 currently unclassified yeast ORFs.

Availability: A program for the prediction method, called FCPFD (Functional Class Prediction based on Functional Domains), is available by contacting Y.D. Cai at y.cai@umist.ac.uk.

INTRODUCTION

The most widely used methods for predicting the function of a new protein sequence involve sequence alignment, either global (over the entire sequence) or local (as in BLAST or FASTA) (Altschul, *et al.*, 1997, Pearson, 1996). Advanced sequence comparison methods (e.g. PSI-Blast (Altschul, *et al.*, 1997)) have pushed the level of sequence identity required to infer homology down to below 20%. The identification of sequence motifs or pattern matching tools is another powerful method to predict function. For example, the PRINTS (Attwood, *et al.*, 2000), BLOCKS (Henikoff, *et al.*, 1999), Pfam (Bateman, *et al.*, 2000), ProDom (Corpet, *et al.*, 1999), SMART (Ponting, *et al.*, 1999), Domo (Gracy and Argos, 1998), Identify (Nevill-Manning, *et al.*, 1998), PROF-PAT (Bachinsky, *et al.*, 2000) and PROSITE (Hofmann, *et al.*, 1999) databases can be used to search an unknown sequence for hundreds of known motifs. InterPro integrates numerous resources for protein families, domains and functional sites (Apweiler, *et al.*, 2000). Correlations between short signals and functional annotations in a protein database can be used (Perez, *et al.*, 2002). Alternatively, structure can first be predicted and then used to infer function (Skolnick and Fetrow, 2000), such as by searching for three-dimensional structural templates of sequence motifs (Kasuya and Thornton, 1999). Even a low level of structure prediction, such as an assignment into a broad fold class, is useful in function prediction, since fold classes show trends for particular functions (Hegyi and Gerstein, 1999).

More recent methods go beyond sequence matching (Eisenberg, *et al.*, 2000, Marcotte, 2000, Pellegrini, 2001). A phylogenetic profile shows the pattern of presence or absence of a protein between genomes. Proteins that have the same phylogenetic profiles can be inferred to be functionally related (Pellegrini, *et al.*, 1999). Protein-protein interactions can be predicted by searching for interacting pairs of proteins that are fused to a single protein chain in another organism (Marcotte, *et al.*, 1999), as such pairs are often functionally related (the Rosetta Stone method). The gene neighbour method uses the observation that if the genes that encode two proteins are close on a chromosome, the proteins tend to be functionally related (Dandekar, *et al.*, 1999, Overbeek, *et al.*, 1999). While such methods can be powerful (Marcotte, *et al.*, 1999), they are limited to cases where their presence varies between genomes, gene fusion has occurred or a set of gene neighbours are observed, and so are not applicable to all cases. More general methods are therefore often needed. Predictions of human protein functional classes and enzyme categories were made with neural networks using sequence properties such as phosphorylation sequences, number of charged residues, predicted secondary structure and average hydrophobicity (Jensen, *et al.*, 2002). Functional class can also be predicted using amino acid and amino acid pair composition (Morrison, *et al.*, 2003). Data mining prediction methods devise rules in the form of IF… THEN statements that make predictions of function using sequence based attributes, predicted secondary structure and sequence similarity. These have been shown to give accurate predictions for a limited number of sequences in the *E. coli* and *M. tuberculosis* genomes, sometimes in the absence of homology (King, *et al.*, 2000, King, *et al.*, 2001).

The major difficulty in this field is finding any information for the large number of genes that are so distant from any known sequence that alignment methods cannot offer any prediction. Traditional sequence alignment and pattern matching approaches are generally not capable of detecting functional and structural homologues when sequence identity falls below about 20%. There is therefore a great need for methodologies capable of predicting protein function within this twilight zone and beyond. This situation is not uncommon: for example, there are no apparent orthologues for 1/3 of the yeast genome (≈2000 proteins) (http://mips.gsf.de/proj/yeast/catalogues/) and ≈15,000 proteins of totally unknown function within the human genome (http://www.genome.ad.jp/dbget-bin/get_htext?H.sapiens.kegg). Here we use functional domain composition to predict protein function. These approaches are not intended to replace existing profile or homology-based search protocols, rather to use alternative data/patterns in the sequence data to go beyond traditional approaches.

The yeast *Saccharomyces cerevisiae* was the first eukaryotic organism to have its entire genome sequenced (Goffeau, *et al.*, 1996, Mewes, *et al.*, 1997, Zagulski, *et al.*, 1998).

The challenge now is identification of the Open Reading Frames (ORFs) and complete functional assignment of these coding regions. Experimental assignment of these ORFs, such as the EUROFAN project (Dujon, 1998), usually takes the form of deletion studies. The ORF of interest is deleted or its expression suppressed. Its function is then inferred from the resulting phenotype under different environmental conditions, for example by withdrawing certain nutrients from the growth media. Alternatively, analysis of cellular mRNA levels can reveal co-expression of certain ORFs under certain environmental conditions. Co-expression of two or more ORFs implies that they function as part of the same cellular pathway. These methods are both time-consuming and expensive.

Here, we present a method for predicting the functions of yeast ORF functions from functional domain composition. We believe this will be a key tool for increasing the throughput of experimental studies. The results of the method will act as a guide to the possible functions an ORF can adopt, whilst virtually eliminating some functions altogether. Experimentalists may therefore approach functional assignment experiments already knowing the most likely and the least likely functions of all ORFs in the yeast genome.

METHODS

Dataset

The MIPS database contains 6294 ORFs of *Saccharomyces cerevisiae*. The aim of this project is to assign a protein into one of 13 functional classes (Metabolism; Energy; Cell Growth; Transcription; Protein Syntheses; Protein Destination; Transport Facilitation; Intracellular Transport; Cellular Biosynthesis; Signal Transduction; Cell Rescue; Cellular Homeostasis and Cellular Organisation), as defined using the MIPS classification system *(Mewes, et al., 1999)*. MIPS assignments were taken from the May 2001 release. A non-homologous data set was generated using CLUSTAL-W (Thompson, et al., 1994) to leave no two proteins having greater than 20% sequence identity. 3484 sequences of known function were reduced to 3010 sequences in the training set (listed in Supplementary Information, Table 1).

Functional Domain Composition

We use SBASE, a collection of around 300,000 annotated structural, functional, ligand-binding and topogenic segments of proteins collected from the literature, protein sequence databases and genomic databases (Vlahovicek, *et al.*, 2002). The protein domains are defined by their sequence boundaries given by the publishing authors or in one of the primary sequence databases (Swiss-Prot, PIR, TREMBL etc.). Domain groups are included if they have well defined sequence boundaries, and if they can be distinguished from other sequences using a similarity search technique. The SBASE database uses a set theoretical approach for representing similarities. Sequences are considered similar if they are members of a similarity group in which all or most sequences are similar to each other and less similar to other members of the database. SBASE-A and SBASE-B Release 9.0 were used (ftp://ftp.icgeb.trieste.it/pub/SBASE (Vlahovicek*, et al.*, 2002)). There are 2425 domain types in SBASE-A (The consolidated domain collection. Release 9.0) and 739 domain types in SBASE-B (Miscellaneous experimental domain groups. Release 9.0).

We use a set of discrete numbers as a vector to define the native functional domains within a protein sequence. With each of the 2425 or 739 domains as a vector-base, a protein can be defined as a 2425-D or 739-D (dimensional) vector according to the following procedure:

(1) Use BLASTP to compare a protein sequence with each of the 300000 sequences grouped in 2425 domain sequences in SBASE-A or 739 domain sequences in SBASE-B to find the high-scoring segment pairs (HSPs) and the smallest sum probability (P).

(2) If the HSP score >> 70 and P < 0.8 for SBASE-A, or the HSP score >> 30 and P < 1.0 for SBASE-B in comparing the protein sequence with the i-th domain sequence, then the i-th

component of the protein in the 2425-D or 739-D space is assigned the HSP score; otherwise, it is assigned a value of zero.

(3) The vector for each protein sequence can thus be explicitly formulated as

$$X = (X1, X2\dots X2425)$$

where

$$X1 = \begin{cases} \text{HSP score, when HSP score} >> 75\ (30) \text{ and } P < 0.8\ (1.0) \\ 0, \text{otherwise} \end{cases}$$

Nearest Neighbour Algorithm (NNA)

The Nearest Neighbour Algorithm (NNA) (Cover and Hart, 1967, Friedman, *et al.*, 1975) can be used particularly in the situations when the distributions of the patterns and the categories of the patterns are unknown. NNA classifies the new patterns into their class membership by comparing the features of the unknown new patterns with the features of the patterns that have already been classified. The approach will weight heavily the evidence derived from the nearby patterns. It is attractive because it is simple to implement and has a low probability of error.

Consider a set of patterns $x_1, x_2\dots x_n$ which have been classified into categories χ_1, $\chi_2\dots \chi_n$, from which an unknown sample x can be classified into those categories using the NNA. Firstly, the nearest neighbour of x can be defined as

$$nn(x) = x_i$$

$$\text{where } d(x, x_i) = \min_{k=1}^{n} d(x, x_k)$$

The nearest neighbour algorithm chooses to classify x to the category χ_j $\{\chi_1, \chi_2\dots \chi_m\}$ if its nearest neighbour also belongs to the category χ_j $\{\chi_1, \chi_2\dots \chi_m\}$. It can be expressed as:

If $nn(x) = x_i$ and $x_i \quad \chi_j \quad \{\chi_1, \chi_2 \ldots \chi_m\}$

Then $x \quad \chi_j$

The score is defined as:

$d(x, x_i) = 1 - (x \bullet x_i)/(\|x\|\|x_i\|)$.

We rank the prediction categories of each protein into $\{\chi_1, \chi_2 \ldots \chi_m\}$ if and only if

$$\min(d(x,x_i)|(x_i \quad \chi_1)) \leq \min(d(x,x_i)|(x_i \quad \chi_2)) \leq \ldots \leq \min(d(x,x_i)|(x_i \quad \chi_m))$$

Here we use "≤" because some proteins belong to more than one category (they are multifunctional) or some proteins whose functions are different share the same domains.

For each protein I, in the dataset of the N function-known proteins, we create a vector $X(I, J)$ (I=1…N, J=1…M) by searching the domain database (M domain types). For a function-unknown protein N+1, we determine its vector $X(N+1, J)$ (J=1…M) by the domain database search. We then calculate the similarity score between protein I and each function-known protein:

$$Score(N+1, I) = 1 - (X(I,J).X(N+1,J))/(|X(I,J)||X(N+1,J)|)$$

I=1…N, J=1…M

We rank the prediction classes of protein I into $\{Class(1), Class(2) \ldots Class(m)\}$ (m is the number of functional classes) if and only if

$$\min(Score(N+1, I_i)|I_i \quad Class(1) \leq \min(Score(N+1, I_i)|I_i \quad Class(2) \leq \ldots$$

$$\leq Score(N+1, I_i)|I_i \quad Class(m)))$$

Two examples of the results are found in Tables 1 and 2.

Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are a class of learning machines based on statistical learning theory. The basic idea of applying SVM to pattern classification can be stated as follows: First, map the input vectors into one feature space (possible with a higher dimension), either linearly or non-linearly, which is relevant to the selection of the kernel function. Then, within the feature space from the first step, seek an optimized linear division.

i.e. construct a hyperplane which separates two classes. (This can be extended to more than two classes). SVM training always seeks a global optimized solution and avoids over-fitting, so it has the ability to deal with a large number of features. A complete description of the theory of SVMs for pattern recognition is in Vapnik's book (Vapnik, 1998). SVMs have been used in a range of problems including drug design (Burbridge, *et al.*, 2000), image recognition and text classification (Joachims, 1998).

In this paper, we apply Vapnik's Support Vector Machine (Vapnik, 1995) for predicting the protein functional class. We downloaded the SVMlight program which is an implementation (in C Language) of SVM for the problem of pattern recognition. The optimization algorithm used in SVMlight has been described (Joachims, 1999, Joachims, 1999). The code has been used in text classification and image recognition (Joachims, 1998).

Suppose we are given a set of samples, i.e., a series of input vectors

$X_i \quad R^d(i = 1,...,N)$

with corresponding labels $y_i \quad \{+1,-1\}(i = 1,...,N)$.

where –1 and +1 are used to stand respectively for the two classes. The goal here is to construct one binary classifier or derive one decision function which has a small probability of misclassifying a future sample (from the available samples). Both the basic linear separable case and the most useful linear non-separable case for most real life problems are considered here:

The linearly separable case

In this case, there exists a separating hyper plane whose function is $\vec{W} \quad \vec{X} + b = 0$, which implies:

$$y_i(\vec{W} \cdot \vec{x}_i + b) \geq 1, i = 1,...,N$$

By minimizing $\frac{1}{2}\|\vec{W}\|^2$ subject to this constraint, the SVM approach tries to find a unique separating hyperplane. Here $\|\vec{w}\|^2$ is the Euclidean norm of $\vec{w}$, which maximizes the distance between the hyper plane (Optimal Separating Hyperplane or OSH (Cortes and Vapnik, 1995) and the nearest data points of each class. The classifier is called the largest margin classifier. By introducing Lagrange multipliers α_i, using the Karush-Kuhn-Tucker conditions and the Wolfe dual theorem of optimization theory, the SVM training procedure amounts to solving the following convex QP problem:

$$Max: \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \; y_i y_j \; \vec{X}_i \cdot \vec{X}_j$$

subject to the following two conditions:

$$\alpha_i \geq 0$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0, i = 1,\ldots,N$$

The solution is a unique globally optimized result having the following expansion:

$$\vec{W} = \sum_{i=1}^{N} y_i \alpha_i \; \vec{x}_i$$

Only if the corresponding $\alpha_i > 0$, are these $\vec{x}_i$ called Support Vectors. When a SVM is trained, the decision function can be written as:

$$f(\vec{x}) = \mathrm{sgn}(\sum_{i=1}^{N} y_i \alpha_i \; \vec{x} \cdot \vec{x}_i + b)$$

where $\mathrm{sgn}(\,)$ in the above formula is the given sign function.

The linearly non-separable case

Two important techniques needed for this case are given respectively as below:

(i) “soft margin” technique.

In order to allow for training errors, Cortes and Vapnik (Cortes and Vapnik, 1995) introduced slack variables:

$$\xi_i > 0, i = 1,\ldots, N$$

The relaxed separation constraint is given as:

$$y_i(\overline{w}\cdot \vec{x}_i + b) \geq 1 - \xi_i \ ,(i = 1,..., N)$$

The OSH can be found by minimizing

$$\frac{1}{2}\|\overline{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

instead of $\frac{1}{2}\|\overline{w}\|^2$ for the above two constraints.

Where C is a regularization parameter used to decide a trade- off between the training error and the margin.

(ii) “kernel substitution” technique

SVM performs a nonlinear mapping of the input vector $\overline{x}$ from the input space R^d into a higher dimensional Hilbert space, where the mapping is determined by the kernel function. Then as in the linearly separable case, it finds the OSH in the space H corresponding to a non-linear boundary in the input space.

Two typical kernel functions are listed below:

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d$$

$$K(\vec{x}_i, \vec{x}_j) = \exp(-r\left\|\vec{x}_i - \vec{x}_j\right\|^2)$$

Where the first one is called the *polynomial kernel function of degree d* which will eventually revert to the linear function when d=1, the latter one is called the RBF (Radial Basic Function) kernel. Finally, for the selected kernel function, the learning task amounts to solving the following QP problem,

$$Max: \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j \ y_i y_j \ K(\vec{X}_i \cdot \vec{X}_j)$$

subject to:

$$0 \leq a_i \leq C$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0, i = 1,...,N$$

The form of the decision function is

$$f(\vec{x}) = \mathrm{sgn}(\sum_{i=1}^{N} y_i \alpha_i \; K\left(\vec{x}, \vec{x}_i\right) + b)$$

For a given data set, only the kernel function and the regularity parameter C must be selected to specify one SVM. In this work we used the radial basis function and set C to 1000.

InterPro

An alternative to SBASE would be to use InterPro (Apweiler, *et al.*, 2000) to search the entire training set for hits to motifs in PROSITE, PRINTS, ProDom, Pfam, SMART, TIGRFAMs, SWISS-PROT and TrEMBL. We therefore submitted the entire set of yeast ORFs to InterPro. We found that 1662 ORFs in the unknown protein MIPS class had hits from InterPro, while there were 2018 hits in this class for SBASE-A. Hence SBASE was more useful in this case.

We used InterPro to predict functional class as follows: We used InterPro release 5.2 (Mulder, *et al.*, 2003), built from Pfam 7.3, PRINTS 33.0, PROSITE 17.5, ProDom 2001.3, SMART 3.1, TIGRFAMs 1.2 and the current SWISS-PROT and TrEMBL data. This release of InterPro contains 5875 entries, representing 1272 domains, 4491 families, 97 repeats and 15 post-translational modification sites. Each ORF is coded as a 5875 dimensional vector with an entry of 1 if there is a hit with a domain, otherwise, 0. This is the input vector for the NNA, implemented as above.

BLAST

We used BLAST to predict functional class as follows: A BLAST search for the 3010 ORFs in the training set was performed by searching each sequence against the remaining

3009 ORFs. The functional class of the best match was taken as the prediction, regardless of any HSP score cut-off.

RESULTS

Jackknife test using Nearest Neighbour Algorithm

We examined the prediction quality by the jackknife test, a leave-one-out cross validation. During the process of the jackknife test, the training and testing datasets are open, and a protein will in turn move from each to the other. The function domain composition of each protein is input to the NNA. Examples of the results are shown in Tables 1 and 2. From Table 2, we can see this method can give some false positives (i.e. YAL020C scores equally highly for classes 3, 4, 8 and 13, though only 13 is correct). This is because some different functional classes share the domains.

Insert Tables 1 and 2

For the total number of 3010 ORFs, the results are shown in Tables 3, 4, 5 and 6. The complete set of results for SBASE-A are in Supplementary Information, Table 2. For a significant proportion of the ORFs (21% for SBASE-A and 8% for SBASE-B) no prediction can be made. These are 435 sequences that have no hits to any motif in SBASE-A and 194 orphan proteins that have no common domains with any other sequence, leaving 2381 sequences for which predictions can be made. If a prediction can be made it is usually correct. Table 4 subdivides the results by functional class, showing considerable variation, depending on the frequency of SBASE motifs in each class. Transport facilitation is most accurately predicted at 90% accuracy using SBASE-A, while Cellular Biogenesis much worse than any other functional class at 15% using SBASE-A. Our overall success rate of 72% accuracy for 79% coverage compares very well to a random guess (19%, given that the mean number of class assignments is 2.4), especially given that homologous sequences have been removed from the training set. According to the yeast genome database (http://mips.gsf.de/proj/yeast/CYGD/db/index.html) of December 2002, 3289/6317 (52%) of proteins are of known function, while 1046/6317 (17%) can be assigned a function based on a FASTA similarity to a known protein. This leaves 1982/6317 (32%) of unknown function. Our success rates therefore compare favourably to existing methods.

Insert Tables 3, 4, 5 and 6

Jackknife test using Support Vector Machines

For the SVMs, we chose a Gaussian Radial Basis Function as the kernel function. The width of the Gaussian RBFs (in this paper, we use the default value in SVMlight) is selected as that which minimized an estimate of the VC-dimension. The parameter C that controls the error-margin tradeoff is set at 1000. The function domain composition of each protein is input to the SVMs. After being trained, the hyperplane output by the SVMs was obtained. This indicates that the trained model, i.e. hyperplane output which includes the important information, is able to identify the protein functional classes. The SVM method applies to two-class problems. In this paper, for the 13-class problems, we use a simple and effective method: “all-versus-all” method (Ding and Dubchak, 2001) to transfer it into a two-class problem. We used leave-one-out cross-validation (jackknife test). 435 sequences have no matches to any SBASE-A sequences so 435/3010 = 14.5% of sequences have no prediction. For the remaining 2575 the success rates are given in Table 7, namely 36% for 86% coverage.

Insert Table 7

The results in Tables 3, 5 and 7 show that the nearest neighbour algorithm is better than the SVMs, despite being computationally much simpler. This may be because the nature of the large and complicated training set is a particular problem for an SVM. The SBASE-A search is better than SBASE-B search. This is probably because the 2425 domains in SBASE-A are long enough to avoid false positive hits when using a BLAST search. As the 739 domains in SBASE-B are quite short, false positive hits from a BLAST search are likely. The false positives are noise for the prediction. Our previous work used amino acid composition and amino acid pair composition to assign MIPS functional class using the

SIMCA algorithm (Morrison, *et al.*, 2003). The accuracy of that work was considerably lower (32%), though the coverage was higher (98%).

Functional Class Assignment using InterPro

We ran the 3010 protein sequences in the training set through InterPro, searching for sequence motifs. 2423 sequences had InterPro hits, leaving 587 that could not be assigned with this method. Table 8 shows the accuracy of the InterPro assignments. The overall success rate of 71% and the coverage (81%) are comparable to the SBASE-A NNA method (72% accuracy and 79% coverage). The pattern of success rate by class is similar with the two methods as class 9, cellular biogenesis, remained particularly problematic.

Functional Class Assignment using BLAST

We compared all of the 3010 protein sequences in the training set against each other, using no cut-off and taking the class of highest scoring sequence as the prediction, no matter how poor the alignment, giving the results shown in Table 9. Despite the training set containing only non-homologous sequences, the resulting poor alignments were still useful, giving an overall accuracy of 70%. This is still a little poorer than the SBASE-A NNA or the InterPro methods. However, the coverage is now 100%. The success rate by class is again similar to the previous methods.

Application to Unclassified Genes

The January 2002 release of the MIPS functional classification scheme lists 115 ORFs as Classification Not Yet Clear Cut and 2399 as Unclassified Proteins. We applied our functional domain composition method using the SBASE-A database and the NNA algorithm to predict the functions of these proteins. From these 2845, 827 cannot be predicted with this method as they have no matches to any functional domain. Predictions for the remaining 2018 ORFs are in the Table 3 in the Supplementary Information. While these sequences have no significant sequence identity to any ORFs of known function in the remainder of the genome, this is also true for our training set. We therefore expect that these predictions will also have an accuracy of approximately 70%.

DISCUSSION

Our results show that the functional class of a protein in the *s. cerevisiae* genome is predictable with high accuracy and coverage. The development in statistical prediction of protein attributes generally consists of two aspects: one is to construct a training dataset and the other is to formulate a prediction algorithm. The latter can be further separated into two sub-sections: one is how to give a mathematical expression to effectively represent a protein sequence and the other is how to find an algorithm to accurately perform the prediction. Here, our training dataset contains 3010 non-homology ORFs which are refined from the whole yeast genome. The functional domain composition incorporates both the sequence-order information and the functional type information and should complement other methods for protein functional class prediction. Success rates and coverage for different methods are summarised in Table 10. The Nearest Neighbour algorithm gave a much better performance than Support Vector Machines and is much faster to run. The results using the SBASE-A domain database were slightly more accurate to InterPro and BLAST.

Insert Table 10

Our predictions for the previously unclassified proteins should be of great value to programs determining experimentally the function of yeast genes, notably the EUROFAN project (Dujon, 1998). The methodology we present should also be generally applicable to all genomes with functional classification schemes that can be used for training.

The program of the prediction method which use NNA, called FCPFD (Functional Class Prediction based on Functional Domains), is available by contacting Y.D. Cai at y.cai@umist.ac.uk.

Acknowledgements

This work was funded by the BBSRC (grant number 36/BIO14432).

REFERENCES

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nuc. Acids Res.*, **25**, 3389-402.

Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M., Bucher, P., Cerutti, L., Corpet, F., Croning, M. D. R., Durbin, R., Falquet, L., Fleischmann, W., Gouzy, J., Hermjakob, H., Hulo, N., Jonassen, I., Kahn, D., Kanapin, A., Karavidopoulou, Y., Lopez, R., Marx, B., Mulder, N. J., Oinn, T. M., Pagni, M., Servant, F., Sigrist, C. J. A. and Zdobnov, E. M. (2000) InterPro—an integrated documentation resource for protein families, domains and functional sites. *Bioinformatics*, **16**, 1145-1150.

Attwood, T. K., Croning, M. D. R., Flower, D. R., Lewis, A. P., Mabey, J. E., Scordis, P., Selley, J. and Wright, W. (2000) PRINT-S: the database formerly known as PRINTS. *Nuc. Acids Res.*, **28**, 225-227.

Bachinsky, A. G., Frolov, A. S., Naumochkin, A. N., Nizolenko, L. P. and Yarigin, A. A. (2000) PROF-PAT 1.3: updated database of patterns used to detect local similarities. *Bioinformatics*, **16**, 358-366.

Bateman, A., Birney, E., Durbin, R., Eddy, S. R., Howe, K. L. and Sonnhammer, E. L. L. (2000) The Pfam protein families database. *Nuc. Acids Res.*, **28**, 263-266.

Burbridge, R., Trotter, M., Holden, S. and Buxton, B. (2000) Drug Design by Machine Learning: Support Vector Machine for Pharmaceutical Data Analysis. *Proceedings of the AISB'00 Symposium on Artificial Intelligence in Bioinformatics*, 1-4.

Corpet, F., Gouzy, J. and Kahn, D. (1999) Recent improvements of the ProDom database of protein domain families. *Nuc. Acids Res.*, **27**, 263-7.

Cortes, C. and Vapnik, V. (1995) Support vector networks. *Machine Learning*, **20**, 273-293.

Cover, T. M. and Hart, P. E. (1967) Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, **13**, 21-27.

Dandekar, T., Snel, B., Huynen, M. and Bork, P. (1999) Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.*, **23**, 324-328.

Ding, C. H. Q. and Dubchak, I. (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics,* **17**, 349-358.

Dujon, B. (1998) European Functional Analysis Network (EUROFAN) and the functional analysis of the Saccharomyces cerevisiae genome. *Electrophoresis,* **19**, 617-624.

Eisenberg, D., Marcotte, E. M., Xenarios, I. and Yeates, T. O. (2000) Protein function in the post-genomic era. *Nature,* **405**, 823-826.

Friedman, J. H., Baskett, F. and Shustek, L. J. (1975) An Algorithm for Finding Nearest Neighbors. *IEEE Transactions on Computers*, **C-24**, 1000-1006.

Goffeau, A., Barrell, B. G., Bussey, H., Davis, R. W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J. D., Jacq, C., Johnston, M., Louis, E. J., Mewes, H. W., Murakami, Y., Philippsen, P., Tettelin, H. and Oliver, S. G. (1996) Life with 6000 genes. *Science,* **274**, 546, 563-567.

Gracy, J. and Argos, P. (1998) Automated protein sequence database classification. I. Integration of compositional similarity search, local similarity search, and multiple sequence alignment. *Bioinformatics*, **14***,* 164-173.

Hegyi, H. and Gerstein, M. (1999) The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. *J. Mol. Biol.*, **288**, 147-64.

Henikoff, S., Henikoff, J. G. and Pietrokovski, S. (1999) Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics,* **15**, 471-9.

Hofmann, K., Bucher, P., Falquet, L. and Bairoch, A. (1999) The PROSITE database, its status in 1999. *Nuc. Acids Res.*, **27**, 215-9.

Jensen, L. J., Gupta, R., Blom, N., Devos, D., Tamames, J., Kesmir, C., Nielsen, H., Staerfeldt, H. H., Rapacki, K., Workman, C., Andersen, C. A. F., Knudsen, S., Krogh, A., Valencia, A. and Brunak, S. (2002) Prediction of human protein function from post-translational modifications and localization features. *J. Mol. Biol.*, **319**, 1257-1265.

Joachims, T. (1999) Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning (B. Scholkopf, C. Burges and A. Smola, eds.), 169-184.

Joachims, T. (1998) Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning*.

Joachims, T. (1999) Transductive Inference for Text Classification using Support Vector Machines. *International Conference on Machine Learning (ICML)*.

Kasuya, A. and Thornton, J. M. (1999) Three-dimensional srtucture analysis of Prosite patterns. *J. Mol. Biol.*, **286**, 1673-1691.

King, R. D., Karwath, A., Clare, A. and Dehaspe, L. (2000) Accurate prediction of protein functional class in the *M. tuberculosis* and *E. coli* genomes using data mining. *Yeast*, **17**, 283-293.

King, R. D., Karwath, A., Clare, A. and Dehaspe, L. (2001) The utility of different representations of protein sequences for predicting functional class. *Bioinformatics*, **17**, 445-454.

Marcotte, E. M. (2000) Computational genetics: finding protein function by nonhomology methods. *Curr. Op. Struct. Biol.*, **10**, 359-365.

Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O. and Eisenberg, D. (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751-3.

Marcotte, E. M., Pellegrini, M., Thompson, M. J., Yeates, T. O. and Eisenberg, D. (1999) A combined algorithm for genome-wide prediction of protein function. *Nature*, **402**, 83-86.

Mewes, H. W., Albermann, K., Bahr, M., Frishman, D., Gleissner, A., Hani, J., Heumann, K., Kleine, K., Maierl, A., Oliver, S. G., Pfeiffer, F. and Zollner, A. (1997) Overview of the yeast genome. *Nature*, **387**, Suppl 7-65.

Mewes, H. W., Jeumann, K., Kaps, A., Mayer, K., Pfeiffer, F., Stocker, S. and Frishman, D. (1999) Mips: a database for genomes and protein sequences. *Nuc. Acids Res.*, **27**, 44-48.

Morrison, R. G., Cai, Y., Doig, A. J. and Mortishire-Smith, R. J. , unpublished.

Mulder, N. J., Apweiler, R., Attwood, T. K., Bairoch, A., Barrell, D., Bateman, A., Binns, D., Biswas, M., Bradley, P., Bork, P., Bucher, P., Copley, R. R., Courcelle, E., Das, U., Durbin, R., Falquet, L., Fleischmann, W., Griffiths-Jones, S., Haft, D., Harte, N., Hulo, N., Kahn, D., Kanapin, A., Krestyaninova, M., Lopez, R., Letunic, I., Lonsdale, D., Silventoinen, V., Orchard, S. E., Pagni, M., Peyruc, D., Ponting, C. P., Selengut, J. D., Servant, F., Sigrist, C. J. A., Vaughan, R. and Zdobnov, E. M. (2003) The InterPro Database, 2003 brings increased coverage and new features. *Nuc. Acids. Res.*, **31**, 315-318.

Nevill-Manning, C. G., Wu, T. D. and Brutlag, D. L. (1998) Highly specific protein sequence motifs for genome analysis. *Proc. Nat. Acad. Sci. U.S.A.*, **95**, 5865-5871.

Oliver, S. G. (1997) EUROFAN's analysis of yeast gene function enters its second phase. *Genome Digest*, **4**.

Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D. and Maltsev, N. (1999) The use of gene clusters to infer functional coupling. *Proc. Nat. Acad. Sci. U.S.A.*, **96**, 2896-2901.

Pearson, W. R. (1996) Effective protein sequence comparison. *Meth. Enzymol.*, **266**, 227-258.

Pellegrini, M. (2001) Computational methods for protein function analysis. *Curr. Op. Chem. Biol.*, **5**, 46-50.

Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D. and Yeates, T. O. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Nat. Acad. Sci. U.S.A.*, **96**, 4285-4288.

Perez, A. J., Rodriguez, A., Trelles, O. and Thode, G. (2002) A computational strategy for protein function assignment which addresses the multidomain problem. *Comp. Funct. Genomics*, **3**, 423-440.

Ponting, C. P., Schultz, J., Milpetz, F. and Bork, P. (1999) SMART: identification and annotation of domains from signalling and extracellular protein sequences. *Nuc. Acids Res.*, **27**, 229-232.

Skolnick, J. and Fetrow, J. S. (2000) From genes to protein structure and function: novel applications of computational approaches in the genomic era. *TIBTECH*, **18**, 34-39.

Thompson, J. D., Higgins, D. G. and Gibson, T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choices. *Nuc. Acids Res.*, **22**, 4673-4680.

Vapnik, V. (1998) Statistical Learning Theory, Wiley-Interscience, New York.

Vapnik, V. N. (1995) The Nature of Statistical Learning Theory, Springer Verlag, New York

Vlahovicek, K., Murvai, J., Barta, E. and Pongor, S. (2002) The SBASE protein domain library, release 9.0: an online resource for protein domain identification. *Nuc Acids. Res.*, **30**, 273-275.

Zagulski, M., Herbert, C. J. and Rytka, J. (1998) Sequencing and functional analysis of the yeast genome. *Acta Biochim. Pol.*, **45**, 627-643.

Table 1. Predictions for YAL005C (true classes: 6, 8, 11 and 13)

Predicted Class	Score	Ranking
6	0.000825	1
8	0.000825	1
11	0.000825	1
13	0.000825	1
9	0.001331	5
3	0.001331	5
10	0.001864	7
5	0.002152	8
7	0.003327	9
4	0.003665	10
1	0.274759	11
12	0.778458	12
2	0.933667	13

Table 2. Predictions for YAL020C (true class: 13)

Predicted Class	Score	Ranking
3	0.000000	1
4	0.000000	1
8	0.000000	1
13	0.000000	1
6	1.000000	-
11	1.000000	-
1	1.000000	-
7	1.000000	-
5	1.000000	-
2	1.000000	-
10	1.000000	-
9	1.000000	-
12	1.000000	-

Table 3. Success Rate of Predicted Sequences for SBASE-A using Nearest Neighbour Algorithm

Most likely class Least likely class

Class	1	2	3	4	5	6	7	8	9	10	11	12	13
rate (%)	72	12	5	2	1	1	1	0	0	0	0	0	0

Table 4. Success Rate by Class for SBASE-A

Predicted Class	Number Correct / Total	% Correct
1. Metabolism	581 / 725	80
2. Energy	108 / 153	71
3. Cell Growth, Cell Division, DNA Synthesis	295 / 507	58
4. Transcription	352 / 504	70
5. Protein Synthesis	195 / 240	81
6. Protein Destination	229 / 330	69
7. Transport Facilitation	194 / 215	90
8. Intracellar Transport	205 / 295	69
9. Cellular Biogenesis	17 / 112	15
10. Signal Transduction	37 / 86	43
11. Cell Rescue	113 / 249	45
12. Ionic Homeostasis	46 / 72	64
13. Cellular Organization	1142 / 1414	81
Overall	3514 / 4902	72

Table 5. Success Rate for SBASE-B using Nearest Neighbour Algorithm

Most likely class Least likely class

Class	1	2	3	4	5	6	7	8	9	10	11	12	13
rate (%)	53	18	10	6	4	3	2	2	1	1	0	0	0

Table 6. Success Rate by Class for SBASE-B

Predicted Class	Number Correct / Total	% Correct
1. Metabolism	458 / 778	59
2. Energy	82 / 186	44
3. Cell Growth, Cell Division, DNA Synthesis	261 / 621	42
4. Transcription	284 / 578	49
5. Protein Synthesis	122 / 252	48
6. Protein Destination	153 / 406	38
7. Transport Facilitation	137 / 250	55
8. Intracellar Transport	154 / 367	42
9. Cellular Biogenesis	21 / 138	15
10. Signal Transduction	26 / 100	26
11. Cell Rescue	88 / 279	32
12. Ionic Homeostasis	28 / 96	29
13. Cellular Organization	1203 / 1692	71
Overall	3017 / 5743	52

Table 7. Success Rate for SBASE-A using Support Vector Machines

Most likely class Least likely class

Class	1	2	3	4	5	6	7	8	9	10	11	12	13
rate (%)	36	22	14	9	6	5	3	3	2	2	2	1	1

Table 8. Success Rate by Class for InterPro

Predicted Class	Number Correct / Total	% Correct
1. Metabolism	698 / 778	71
2. Energy	110 / 170	90
3. Cell Growth, Cell Division, DNA Synthesis	302 / 489	65
4. Transcription	344 / 497	62
5. Protein Synthesis	191 / 253	75
6. Protein Destination	249 / 350	71
7. Transport Facilitation	171 / 208	82
8. Intracellar Transport	209 / 297	70
9. Cellular Biogenesis	27 / 110	25
10. Signal Transduction	44 / 88	50
11. Cell Rescue	137 / 248	55
12. Ionic Homeostasis	51 / 82	62
13. Cellular Organization	1062 / 1474	72
Overall	3595 / 5044	71

Table 9. Success Rate by Class for BLAST

Predicted Class	Number Correct / Total	% Correct
1. Metabolism	636 / 880	72
2. Energy	136 / 209	65
3. Cell Growth, Cell Division, DNA Synthesis	418 / 644	65
4. Transcription	441 / 608	73
5. Protein Synthesis	198 / 302	66
6. Protein Destination	280 / 435	64
7. Transport Facilitation	200 / 252	79
8. Intracellar Transport	236 / 378	62
9. Cellular Biogenesis	51 / 142	36
10. Signal Transduction	62 / 102	61
11. Cell Rescue	173 / 294	59
12. Ionic Homeostasis	54 / 96	56
13. Cellular Organization	1412 / 1825	77
Overall	4297 / 6167	70

Table 10. Success Rates and Coverage for Different Methods

Method	% Correct	% Coverage
SBASE-A Nearest Neighbour	72	79
SBASE-B Nearest Neighbour Algorithm	53	92
SBASE-A Support Vector Machines	36	79
InterPro	71	81
BLAST	70	100