Machine learning of functional class from phenotype data

Amanda Clare Ross D. King

Department of Computer Science,
University of Wales Aberystwyth, SY23 3DB, UK

Abstract

Motivation: Mutant phenotype growth experiments are an important novel source of functional genomics data which have received little attention in bioinformatics. We applied supervised machine learning to the problem of using phenotype data to predict the functional class of ORFs in *S. cerevisiae*. Three sources of data were used: TRIPLES, EUROFAN and MIPS. The analysis of the data presented a number of challenges to machine learning: multi-class labels, a large number of sparsely populated classes, the need to learn a set of accurate rules (not a complete classification), and a very large amount of missing values. We modified the algorithm C4.5 to deal with these problems.

Results: Rules were learnt which are accurate and biologically meaningful. The rules predict function of 83 ORFs of unknown function at an estimated accuracy of $\geq 80\%$.

Availability: The data and complete results are available at http://users.aber.ac.uk/ajc99/phenotype/.

Contact: ajc99@aber.ac.uk

1 Introduction

Computational biology is playing an increasingly important role in determining the function of genes. Sequence similarity searches such as PSI-BLAST (Altschul et al., 1997) are generally the starting point for inferring function of new genes or ORFs (Open Reading Frames). Moving on from sequence similarity, the different types of data available for use in functional genomics are growing each year, and new techniques are needed to keep pace.

Computational techniques currently of use in inferring function include using expression data clustering (DeRisi, Iyer, and Brown, 1997; Eisen et al., 1998), structure prediction (CASP, 1999), and combined approaches which can make use of several sources of data. Marcotte *et al.* (1999) describe a scheme of inter-linking yeast proteins according to various relations, including whether they are part of the same pathway, have similar expression patterns, or links from domain-fusion analysis. Links between proteins of known and unknown function can then be used to assign functions to the unknowns. Three machine learning methods were used for prediction by des Jardins *et al.* (1997), who predicted enzyme class of proteins from PDB and SwissProt using features computed from amino acid sequence. King *et al.* (2000; 2001) used data mining in the form of Inductive Logic Programming and the supervised machine learning program C4.5 to

learn rules based on sequence, structure and homology data from the *M. tuberculosis* and *E. coli* genomes. These rules could then be directly applied to predict function of unknown genes. Recently, other supervised learning algorithms have also been investigated. Support vector machines have been used to analyse yeast expression data (Brown et al., 2000), and Pavlidis *et al.* (2001) have extended this to learn from both expression and phylogenetic data.

One new source of data which is of increasing value in determining the function of ORFs is phenotypic growth data. This is *data from experiments about the sensitivity or resistance of disruption mutants under various growth conditions.* Novel experimental techniques have made it possible to collect such data on a genome-wide scale. This paper is, to the best of our knowledge, the first on developing data analysis methods for such data.

The best studied organism for phenotypic growth data is *S. cerevisiae.* Despite this being one of the most extensively studied of all organisms, the function of 30-40% of its ORFs are currently unknown. The MIPS database has several ORF functional classification schemes for yeast, one of these being a hierarchical "Functional Classification Catalogue" (http://mips.gsf.de/proj/yeast/catalogues/funcat/). This catalogue classifies the functions of ORFs under various general classes, such as "Metabolism", "Energy", "Transcription" and "Protein Synthesis". Each of these classes is then subdivided into more specific classes, and these are in turn subdivided, and then again subdivided, so the hierarchy is up to 4 levels deep. An example of a subclass of "Metabolism" is "amino-acid metabolism", and an example of a subclass of this is "amino-acid biosynthesis". An example of an ORF in this subclass is YPR145w (gene name ASN1, product "asparagine synthetase").

The existence of functional hierarchies, such as this MIPS catalogue, opens up the possibility of using supervised machine learning to predict the functional classification of ORFs (Kell and King, 2000). Most work on classification in computational biology has been with *unsupervised learning/clustering* classification algorithms (Eisen et al., 1998; Koonin et al., 1998; Törönen et al., 1999). Few have made use of the fact that we already have knowledge about the functions of many of the ORFs. *Supervised* classification algorithms can make use of this knowledge and are now beginning to be used in bioinformatics (Brown et al., 2000; des Jardins et al., 1997). For discussion of the use of functional hierarchies and suitable types of machine learning, see Kell and King (2000), and for systems for categorising functions see Riley (1998) and Andrade *et al.* (1999).

An ORF may have several different functions, and this is reflected in the MIPS classification scheme (where a single ORF can belong to up to 10 different functional classes). *This presents an unusual and interesting classification problem for machine learning.* It is a *multi-label* problem (as opposed to *multi-class* which usually refers to simply having more than two possible disjoint classes for the classifier to learn). There is only a limited literature on such problems, for example (Karalic and Pirnat, 1991; McCallum, 1999; Schapire and Singer, 2000). The simplest approach to the problem is to learn separate classifiers for each class (with all ORFs not belonging to a specific class used as negative examples for that class). However this is clearly cumbersome and time-consuming when there are many classes - as is the case in the functional hierarchy in yeast. Also, in sparsely populated classes there would be very few positive examples of a class and overwhelmingly many negative examples. We have therefore developed a new algorithm based on the successful decision tree algorithm C4.5 (Quinlan, 1993).

In summary our approach is to develop a

specific machine learning method to learn rules which map from phenotype data to functional class. Rules are learnt and their accuracy estimated using phenotype data from deletion mutants of ORFs of known function. These rules can then be applied to ORFs of unknown function for prediction.

2 Experimental method

2.1 Data

We used three separate sources of phenotypic data: TRIPLES (Kumar et al., 2000), EUROFAN (Oliver, 1996) and MIPS (Mewes et al., 1999).

- The TRIPLES (TRansposon-Insertion Phenotypes, Localization and Expression in *Saccharomyces*) data was generated by randomly inserting transposons into the yeast genome.
  URLs: http://ygac.med.yale.edu/triples/triples.htm, (raw data)
  http://bioinfo.mbb.yale.edu/genome/phenotypes/ (processed data)

- EUROFAN (European functional analysis network) is a large European network of research which has created a library of deletion mutants by using PCR-mediated gene replacement (replacing specific genes with a marker gene (kanMX)). We used data from EUROFAN 1.
  URL: http://mips.gsf.de/proj/eurofan/

- The MIPS (Munich Information Center for Protein Sequences) database contains a catalogue of yeast phenotype data.
  URL: http://mips.gsf.de/proj/yeast/

The data from the three sources were concatenated together to form a unified dataset http://users.aber.ac.uk/ajc99/phenotype/.

The phenotype data has the form of attribute-value vectors: with the attributes being the growth media, the values of the attributes being the observed sensitivity or resistance of the mutant compared with the wildtype, and the class the functional class of the ORF. Note that this data is not available for all ORFs due to some mutants being inviable or untested, and not all growth media were tested/recorded for every ORF, so there were *many missing values* in the data.

The values that the attributes could take were the following: n = no data, w = wild-type (no phenotypic effect), s = sensitive (less growth than for the wild-type), and r = resistance (better growth than for the wild-type).

There were 69 attributes, 68 of which were the various growth media (e.g. calcofluor_white, caffeine, sorbitol, benomyl), and one which was a discretised count of how many of the media this mutant had shown a reaction to (i.e. for how many of the attributes this mutant had a value of "s" or "r").

2.2 Algorithm

The machine learning algorithm we chose to adapt for the analysis of phenotype data was C4.5 (Quinlan, 1993). This is a well known decision tree algorithm which is robust, and efficient (Michie, Spiegelhalter, and Taylor, 1994). The output of C4.5 is a decision tree, or equivalently a set of symbolic rules. The use of symbolic rules allows the output to be interpreted and compared with existing biological knowledge - this is not generally the case with other machine learning methods, such as neural networks, or support vector machines.

A decision tree is a tree where each node is a test on the values of an attribute, and the leaves represent the class of an item which satisfies the tests. Rules can then be read off from the tree by following a path from the root node to a leaf and using the nodes along the path as

preconditions for the rule, to predict the class at the leaf. The rules can be pruned to remove unnecessary preconditions and duplication.

In C4.5 the tree is constructed top down. For each node the attribute is chosen which best classifies the remaining training examples. This is decided by considering the information gain, the difference between the entropy of the whole set of remaining training examples and the weighted sum of the entropy of the subsets caused by partitioning on the values of that attribute.

$$information_gain(S, A) = entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} * entropy(S_v)$$

where A is the attribute being considered, S is the set of training examples being considered, and S_v is the subset of S with value v for attribute A. The C4.5 algorithm is well documented and the code is open source, so this allowed the algorithm to be extended.

Multiple labels are a problem for C4.5, and almost all other learning methods, as it expects each example to be labeled as belonging to just one class. For yeast ORF function this is not the case, as an ORF may belong to several different classes. In the case of a single class label for each example the entropy for a set of examples is just

$$entropy(S) = -\sum_{i=1}^{N} p(c_i) \log p(c_i)$$

where $p(c_i)$ is the probability (relative frequency) of class c_i in this set.

We need to modify this formula for multiple classes. The information for an example is now the number of bits needed to describe the *classes* it belongs to. To estimate this we sum the number of bits needed to describe membership or non-membership of each class. In the general case where there are N classes and membership of each class c_i has probability $p(c_i)$ the total number of bits needed for an average example is given by

$$-\sum_{i=1}^{N} (p(c_i) \log p(c_i)) + (q(c_i) \log q(c_i))$$

where
$p(c_i)$ = probability (relative frequency) of class c
$q(c_i) = 1 - p(c)$ = probability of not being member of class c

The resulting information after a partition according to some attribute, can be calculated as a weighted sum of the entropy for each subset (calculated as above), where this time, weighted sum means if an item appears twice in a subset because it belongs to two classes then we count it twice.

In allowing multiple labels per example we have to allow leaves of the tree to potentially be a set of class labels, i.e. the outcome of a classification of an example can be a set of classes. When we label the decision tree this needs to be taken into account, and also when we prune the tree. When we come to generate rules from the decision tree, this can be done in the usual way, except when it is the case that a leaf is a set of classes, a separate rule will be generated for each class, prior to the rule-pruning part of the C4.5rules algorithm. We could have generated rules which simply output a set of classes - it was an arbitrary choice to generate separate rules, chosen for comprehensibility of the results.

2.3 Resampling

In most statistical and machine learning supervised classification problems the aim is to maximise the prediction accuracy on the test set. This is not the case for our problem. Instead, *we wish to learn a set of rules which*

accurately predict functional class. This resembles in some respects association rule learning in data mining. The problem is also unusual in machine learning terms in that there are a very large number of classes. For example there are 99 potential classes represented in the data for level 2 in the class hierarchy: in a typical machine learning problem there are at most a handful. These unusual features of the data made it necessary for us to develop a complicated resampling approach to estimating rule accuracy based on the bootstrap.

All accuracy measurements were made using the m-estimate (Cestnik, 1990) which is a generalisation of the Laplace estimate, taking into account the *a priori* probability of the class.

$$M(r) = \frac{p + m\frac{P}{P+N}}{p + n + m}$$

where P = total number of positive examples, N = total number of negative examples, p = number of positive examples covered by rule r, n = number of negative examples covered by rule r.

Using this formula, the accuracy for rules with zero coverage will be the *a priori* probability of the class. m is a parameter which can be altered to weight the *a priori* probability. We used m=1.

The data set in this case is small in machine learning terms. We have 2452 ORFs with some recorded phenotypes, of which 991 are classified by MIPS as "Unclassified" or "Classification not yet clear-cut". These ORFs of unknown classification cannot be used in supervised learning (though we can later make predictions for them). This leaves just 1461, each with many missing values. At the top level of the classification hierarchy (the most general classes), there are many examples for each class, but as we move to lower, more specific levels, the classes become more sparsely populated, and machine learning becomes difficult.

We aimed to learn rules for predicting functional classes which could be interpreted biologically. To this end we evaluated splitting the data set into 3 parts: training data, validation data to select the best rules from (rules were chosen that had an accuracy of at least 50% and correctly covered at least 2 examples), and test data. We used the validation data to avoid overfitting rules to the data. However, splitting the dataset into 3 parts means that the amount of data available for training will be even less. Similarly only a small amount will be available for testing. Initial experiments showed that the split of the data substantially affected the rulesets produced, sometimes producing many good rules, and sometimes none. The two standard methods for estimating accuracy under the circumstance of a small data set are 10-fold cross-validation and the bootstrap method (Kohavi, 1995; Efron and Tibshirani, 1993). Because we are interested in the rules themselves, and not just the accuracy, we opted for the bootstrap method, because a 10-fold cross validation would make just 10 rulesets, whereas bootstrap sampling can be used to create hundreds of samples of the data and hence hundreds of rulesets. We can then examine these and see which rules occur regularly and are stable, not just artifacts of the split of the data.

The bootstrap is a method where data is repeatedly sampled with replacement to make hundreds of training sets. A classifier is constructed for each sample, and the accuracies of all the classifiers can be averaged to give a final measure of accuracy. First a bootstrap sample was taken from the original data. Items of the original data not used in the sample made up the test set. Then a new sample was taken with replacement *from the sample.* This second sample was used as training data, and items that were in the first sample but not in the second made up the validation set. All three data sets are non-overlapping.

We measured accuracy on the held-out test set. We are aware that this will give a *pessimistic* measure of accuracy (i.e. the true accuracy on the whole data set will be higher), but this is acceptable.

3 Results

We attempted to learn rules for all classes in the MIPS functional hierarchy http://mips.gsf.de/proj/yeast/catalogues/funcat/, using the catalogue as it was on 27 September 1999. 500 bootstrap samples were made, and so C4.5 was run 500 times and 500 rulesets were generated and tested. To discover which rules were stable and reliable we counted how many times each rule appeared across the 500 rulesets. Accurate stable rules were produced for many of the classes at levels 1 and 2 in the hierarchy. At levels 3 and 4 (the most specific levels with the least populated classes) no useful rules were found. That is, at the lower levels, few rules were produced and these were not especially general or accurate. The topic of learning within a class hierarchy when child classes are sparsely populated, and making good use of the hierarchy, is something to consider in future experiments of this kind.

The good rules are generally very simple, with just one or two conditions necessary to discriminate the classes. This was expected, especially since most mutants were only sensitive/resistant to a few media. Some classes were far easier to recognise than others, for example, many good rules predicted class "CELLULAR BIOGENESIS" and its subclass "biogenesis of cell wall (cell envelope)".

Some examples of the rules and their accuracies follow. The full set of rules can be seen at http://users.aber.ac.uk/ajc99/phenotype/ along with the data sets used.

The 4 most frequently appearing rules at level 1 (the most general level in the functional catalogue) are all predictors for the class "CELLULAR BIOGENESIS". These rules suggest that sensitivity to zymolase or papulacandin_b, or any reaction (sensitivity or resistance) to calcofluor_white is a general property of mutants whose deleted ORFs belong to the CELLULAR BIOGENESIS class. All correct ORFs matching these rules in fact also belong to the subclass "biogenesis of cell wall (cell envelope)". The rules are far more accurate than the prior probability of that class would suggest should occur by chance.

Below are two of the rules regarding sensitivity/resistance to Calcofluor White.

```
if  the ORF deletant is sensitive to calcofluor
        white and
    the ORF deletant is sensitive to zymolyase
then its class is "biogenesis of cell
        wall (cell envelope)"
Mean accuracy:       90.9%
Prior prob of class:  9.5%
Std dev accuracy:     1.8%
Mean no. matching orfs: 9.3

if  the ORF deletant is resistant to calcofluor
        white
    then its class is "biogenesis of cell
        wall (cell envelope)"
Mean accuracy:       43.8%
Prior prob of class:  9.5%
Std dev accuracy:    14.4%
Mean no. matching orfs: 6.7
```

These rules confirm that Calcofluor White is useful for detecting cell wall mutations (Ram et al., 1994; Lussier et al., 1997). Calcofluor White is a negatively charged fluorescent dye that does not enter the cell wall. Its main mode of action is believed to be through binding to chitin and prevention of microfibril formation and so weakening the cell wall. The explanation for disruption mutations in the cell wall having increased sensitivity to Calcofluor White is believed to be that if the cell wall is weak, then the cell may not be able to withstand further disturbance. The explanation for

resistance is less clear, but the disruption mutations may cause the dye to bind less well to the cell wall. Zymolase is also known to interfere with cell wall formation (Lussier et al., 1997). Neither rule predicts the function of any ORF of currently unassigned function. This is not surprising given the previous large scale analysis of the Calcofluor White on mutants.

One rule that does predict a number of ORFs of unknown function is:

```
if the ORF deletant is sensitive to
     hydroxyurea
   then its class is "nuclear organization"
Mean accuracy:       40.2%
Prior prob of class: 21.5%
Std dev accuracy:     6.6%
Mean no. matching orfs: 33.4
```

This rule predicts 27 ORFs of unassigned function. The rule is not of high accuracy but it is statistically highly significant. Hydoxyurea is known to inhibit DNA replication (Sugimoto et al., 1995), so the rule is biologically consistent.

```
if the ORF deletant is sensitive to YPGlyc
   and the number of media the ORF deletant
       is sensitive or resistant to is low
   then its class is "mitochondrial
       organization"
Mean accuracy:       52.2%
Prior prob of class:  7.9%
Std dev accuracy:    11.8%
Mean no. matching orfs: 8.2
```

To grow aerobically on Glycerol as a sole carbon source yeast requires functioning mitochondria. YPGlyc is a growth media with glycerol as sole carbon source, therefore it is consistent that mutants lacking ORFs involved in mitochondrial organization will be sensitive to growth in this medium. This rule is more than 6 times more accurate than the *a priori* probability would suggest by chance. Examples of ORFs that this rule correctly predicted include YMR035W (mitochondrial inner membrane protease subunit) and YNR045W (translational activator, mitochondrial). The 9/24 ORFs that this rule wrongly predicted include YCL040W (aldohexose specific glucokinase) and YCR012W (phosphoglycerate kinase), both associated with gluconeogenesis/glycolysis. YCR012W (phosphoglycerate kinase) has no known isoenzyme and would therefore be expected to be essential for growth on glycerol. The sensitivity of the YCL040W (aldohexose specific glucokinase) deletant mutant is less clear and therefore perhaps more interesting. The functions of the the other wrongly predicted ORFs, such as YMR188C (weak similarity to bacterial ribosomal protein S17), have no known connection with carbon metabolism and the reason for the sensitivity to growth on glycerol is unclear.

The rules can appear in several similar forms, some more general than others. This rule about sensitivity to YPGlyc appears in various forms with other conditions (usually that the mutant ORF is equally as sensitive as the wildtype to various media). This leads to very specific rules supporting the general rule, such as the following:

```
if the ORF deletant is
       as sensitive as wildtype to EGTA
   and as sensitive as wildtype to SDS
   and sensitive to YPGlyc
   and as sensitive as wildtype to calcofluor_white
   and as sensitive as wildtype to hygromycin_b
   then its class is "mitochondrial organization"
Mean accuracy:       51.9%
Prior prob of class:  7.9%
Std dev accuracy:    20.8%
Mean no. matching orfs: 12.667
```

The following rule shows a very low accuracy, no better than the *a priori* probability, due to few examples.

```
if the ORF deletant is sensitive to canavanine
   then its class is "stress response"
```

```
Mean accuracy:       5.0%
Prior prob of class: 6.0%
Std dev accuracy:    1.6%
Mean no. matching orfs: 0.3
```

Lack of statistical significance does not necessarily mean that the rule is not biologically interesting. Canavanine is an analogue of arginine, and cells which have this present will take up canavanine instead of synthesising arginine. They will then have short lives. Ubiquitin overexpression is known to increase tolerance to canavanine (Chen and Piper, 1995), so mutants which affect ubiquitin may be expected to be more sensitive than the wildtype. YBR082C (E2 ubiquitin-conjugating enzyme) and YER125W (ubiquitin-protein ligase) are two of the ORFs matching this rule, both in the "stress response" class, so it is possible that this rule is correct, but we just do not have the number of examples needed for statistical confidence. However, analysis of the biological explanation for this rule can still be interesting.

Tables 1 and 2 show the number of ORFs of unassigned function predicted by the learnt rules at levels 1 and 2 in the functional hierarchy. These are plotted as a function of the estimated accuracy of the predictions and the significance (how many standard deviations the estimated accuracy is from the prior probability of the class). These figures record ORFs predicted by rules that have appeared at least 5 times during the bootstrap process.

estimated accuracy	std. deviations from prior 2	3	4
$\geq 80\%$	83	72	35
$\geq 70\%$	209	150	65
$\geq 50\%$	211	150	65

Table 1: Number of ORFs of unknown function predicted at Level 1.

estimated accuracy	std. deviations from prior 2	3	4
$\geq 80\%$	63	63	63
$\geq 70\%$	77	77	77
$\geq 50\%$	133	126	126

Table 2: Number of ORFs of unknown function predicted at Level 2.

It can be seen that analysis of the phenotype growth data allows the prediction of the functional class of many of the ORFs of currently unassigned function.

When we compare our results with other work in supervised learning of function, it can be seen that we have similar levels of prediction and accuracy. For example Marcotte *et al* (1999) used a nearest-neighbour type method on 5 different types of data, including expression data and metabolic pathways. They predicted 15% of the ORFs of unknown function at high accuracy (and 62% at lower accuracy). des Jardins *et al.* (1997) report an accuracy of 74% in prediction of level 1 of the enzyme classification, and 68% at level 2. Pavlidis *et al.* (2001) built separate classifiers for each of 27 of the MIPS functional classes, chosen after preliminary investigation into learnability of the data. However, they do not make predictions. Most other work on prediction of function has been either through unsupervised learning, or by choosing a very small number of classes to predict, such as in Brown *et al.* (2000).

4 Discussion and conclusion

Many accurate and simple rules have been found which can predict an ORF's functional class from mutant phenotype experiments. Biological relevance for several of the rules has been discussed.

The full set of rules can be found at http://users.aber.ac.uk/ajc99/phenotype/.

The technique can be added to the toolbox of biologists and computational biologists when making hypotheses about the roles of ORFs within a genome, and the method should be easily portable to other genomes for which mutant phenotype data can be collected.

The rules are also useful as they show future experimenters *which media provide the most discrimination between functional classes.* Many types of growth media are shown to be highly informative for identifying the functional class of disruption mutants (e.g. Calcofluor White), others are of little value (e.g. sodium chloride). The nature of the C4.5 algorithm is always to choose attributes which split the data in the most informative way. This knowledge can be used in the next round of phenotypic experiments.

It is also interesting to note that different functional classes are predicted better by different types of data. For example here we have seen that this phenotypic data is a very good predictor of class "biogenesis of cell wall (cell envelope)". In their investigations with SVMs, Pavlidis *et al.* (2001) also acknowledge that some classes are easier to learn than others. With expression data it would seem to be ribosomal proteins that are easy to learn, and with phylogenetic data, transporter proteins are predictable (we have also found these results in our current work). This independence in functional genomics data will be a great help in determining the functions of ORFs.

Working with the phenotypic growth data highlighted several machine learning issues which are interesting:

- We had to extend C4.5 to handle the problem of ORFs having more than one function, the multi-label problem.
- We needed to select rules for biological interest rather than predicting all examples, this required us to use an unusual rule selection procedure, and also led to our choice of the bootstrap to give a clearer picture of the rules themselves.

Our work illustrates the value of cross-disciplinary work. Functional genomics is enriched by a technique for improved prediction of the functional class of ORFs: and machine learning is enriched by provision of new data analysis challenges.

Acknowledgments: We would like to thank Ugis Sarkans for initial collection of the data and Stephen Oliver and Douglas Kell for useful discussions. Amanda Clare was supported by MRC grant G78/6609.

References

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

Andrade, M., C. Ouzounis, C. Sander, J. Tamames, and A. Valencia (1999). Functional classes in the three domains of life. *Journal of Molecular Evolution* 49: 551–557.

Brown, M., W. Nobel Grundy, D. Lin, N. Cristianini, C. Walsh Sugnet, T. Furey, M. Ares Jr., and D. Haussler (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Nat. Acad. Sci. USA* 97(1): 262–267.

CASP (1999). Third meeting on the critical assessment of techniques for protein structure prediction. Supplement in Proteins: Structure, Function and Genetics 37(S3).

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. In *Proceedings of the Ninth European Conference on Artificial Intelligence (ECAI90)*, pp. 147–149.

Chen, Y. and P.W. Piper (1995). Consequences of the overexpression of ubiquitin in yeast: elevated tolerances of osmostress, ethanol and canavanine, yet reduced tolerances of cadmium, arsenite and paromomycin. *Biochim Biophys Acta* 1268(1): 59–64.

DeRisi, J., V. Iyer, and P. Brown (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278: 680–686.

des Jardins, M., P. Karp, M. Krummenacker, T. Lee, and C. Ouzounis (1997). Prediction of enzyme classification from protein sequence without the use of sequence similarity. In *ISMB '97*.

Efron, B. and R. Tibshirani (1993). *An introduction to the bootstrap*. Chapman and Hall.

Eisen, M., P. Spellman, P. Brown, and D. Botstein (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Nat. Acad. Sci. USA* 95: 14863–14868.

Karalic, Aram and Vlado Pirnat (1991). Significance level based classification with multiple trees. *Informatica* 15(5).

Kell, D. and R. King (2000). On the optimization of classes for the assignment of unidentified reading frames in functional genomics programmes: the need for machine learning. *Trends Biotechnol.* 18: 93–98.

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

King, R., A. Karwath, A. Clare, and L. Dehaspe (2001). The utility of different representations of protein sequence for predicting functional class. *Bioinformatics* in press.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI 1995*.

Koonin, E., R. Tatusov, M. Galperin, and M. Rozanov (1998). Genome analysis using clusters of orthologous groups (COGS). In *RECOMB 98*, pp. 135–139.

Kumar, A., K.-H. Cheung, P. Ross-Macdonald, P.S.R. Coelho, P. Miller, and M. Snyder (2000). TRIPLES: a database of gene function in S. cerevisiae. *Nucleic Acids Res.* 28: 81–84.

Lussier, M., A. White, J. Sheraton, T. Paolo, J. Treadwell, S. Southard, C. Horenstein, J. Chen-Weiner, A. Ram, J. Kapteyn, T. Roemer, D. Vo, D. Bondoc, J. Hall, W. Zhong, A. Sdicu, J. Davies, F. Klis, P. Robbins, and H. Bussey (1997). Large scale identification of genes involved in cell surface biosynthesis and architecture in *Saccharomyces cerevisiae*. *Genetics* 147: 435–450.

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

McCallum, A. (1999). Multi-label text classification with a mixture model trained by EM. In *AAAI 99 Workshop on Text Learning*.

Mewes, H.W., K. Heumann, A. Kaps, K. Mayer, F. Pfeiffer, S. Stocker, and D. Frishman (1999). MIPS: a database for protein sequences and complete genomes. *Nucleic Acids Research* 27: 44–48.

Michie, D., D. J. Spiegelhalter, and C. C. Taylor, editors (1994). *Machine Learning, Neural and Statistical Classification.* Ellis Horwood, London. Out of print but available at http://www.amsta.leeds.ac.uk/~charles/statlog/.

Oliver, S. (1996). A network approach to the systematic analysis of yeast gene function. *Trends in Genetics* 12(7): 241–242.

Pavlidis, P., J. Weston, J. Cai, and W. Grundy (2001). Gene functional classification from heterogenous data. In *Proceedings of the Fifth International Conference on Computational Molecular Biology (RECOMB 2001).*

Quinlan, J. R. (1993). *C4.5: programs for Machine Learning.* Morgan Kaufmann, San Mateo, California.

Ram, A., A. Wolters, R. Ten Hoopen, and F. Klis (1994). A new approach for isolating cell wall mutants in *Saccharomyces cerevisiae* by screening for hypersensitivity to calcofluor white. *Yeast* 10: 1019–1030.

Riley, M. (1998). Systems for categorizing functions of gene products. *Current Opinion in Structural Biology* 8: 388–392.

Schapire, R. and Y. Singer (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning* 39(2/3): 135–168.

Sugimoto, K., Y. Sakamoto, O. Takahashi, and K. Matsumoto (1995). HYS2, an essential gene required for DNA replication in Saccharomyces cerevisiae. *Nucleic Acids Res* 23(17): 3493–500.

Törönen, P., M. Kolehmainen, G. Wong, and E. Castrén (1999). Analysis of gene expression data using self-organizing maps. *FEBS Lett.* 451(2): 142–6.