Comparing Several Approaches for Hierarchical Classification of Proteins with Decision Trees

Eduardo P. Costa[1], Ana C. Lorena[2], André C. P. L. F. Carvalho[1], Alex A. Freitas[3], and Nicholas Holden[3]

[1]Depto. Ciências de Computação
ICMC/USP - Sao Carlos - Caixa Postal 668
13560-970 - Sao Carlos-SP, Brazil
{ecosta,andre}@icmc.usp.br
[2]Universidade Federal do ABC
09.210-170 - Santo André-SP, Brazil
ana.lorena@ufabc.edu.br
[3]Computing Laboratory
University of Kent, Canterbury, CT2 7NF, UK
{a.a.freitas,nh56}@kent.ac.uk

Abstract. Proteins are the main building blocks of the cell, and perform almost all the functions related to cell activity. Despite the recent advances in Molecular Biology, the function of a large amount of proteins is still unknown. The use of algorithms able to induce classification models is a promising approach for the functional prediction of proteins, whose classes are usually organized hierarchically. Among the machine learning techniques that have been used in hierarchical classification problems, one may highlight the Decision Trees. This paper describes the main characteristics of hierarchical classification models for Bioinformatics problems and applies three hierarchical methods based on the use of Decision Trees to protein functional classification datasets.

1 Introduction

In functional genomics, an important problem is the prediction of the function of proteins. Proteins are the main building blocks of the cell, and perform almost all the functions related to cell activity. The primary sequence of a protein consists of a linear string of amino acids, which is then folded into a specific 3-D shape necessary for the protein to function properly. Proteins often share common amino acid sub-sequences due to evolutionary processes.

An approach frequently used in the prediction of a protein function is to search for similar sequences in protein databases. The objective is to find a similar sequence whose function is known. If a similar protein sequence is found, its function is assigned to the new protein. Although this method is very useful in a large number of situations, it has also some limitations [1]. Two proteins might have very similar sequences and perform different functions, or have very different sequences and perform the same or a similar function. Additionally, the

proteins being compared may be similar in regions of the sequence that are not determinants of their function.

A second approach may be used alternatively or in complement to the similarity-based approach. The central idea of this approach consists of inducing a classification model for the prediction of protein function. Each protein is represented by an attribute set and a learning algorithm captures the most important relationships between the attributes and the classes present in the dataset.

As protein functional data is, frequently, organized hierarchically (for example, in the Gene Ontology [2] and in the Enzyme Commission hierarchy [3]), the use of hierarchical techniques for the induction of classification models in Bioinformatics is a promising research area.

This paper treats the main aspects concerned with hierarchical classification in Bioinformatics. Two datasets were used for a comparative study among different schemes for hierarchical classification. Decision Trees (DTs) [4] were used in the classifiers induction.

The main contribution of this paper is to compare several different approaches for the hierarchical classification of proteins. To the best of our knowledge, an empirical comparison of the approaches evaluated in this paper has not been reported yet in the literature.

The paper is organized as follows: Section 2 introduces important concepts of hierarchical classification; Section 3 presents the main approaches used in the induction of classifiers for hierarchical classification problems, as well as some literature related to the protein function prediction problem; Section 4 discusses the materials and methods employed in the experiments performed in this work; Section 5 presents the experimental results; and Section 6 has the main conclusions of this work.

2 Hierarchical Classification

Classification is one of the most important problems in Machine Learning (ML) and Data Mining (DM) [5]. A classification problem can be defined as the the process of finding a function, through a training or adjustment phase, wich maps each input instance T_i into one of the N classes of the problem, with $i = 1, 2, ..., n$, where n is the number of training instances.

The vast majority of classification problems reported in the literature involves flat classification, where each instance is assigned to a class out of a finite (and usually small) set of flat classes. Nevertheless, there are more complex classification problems, where the classes to be predicted are hierarchically related [1, 6]. In these classification problems, one or more classes can be divided into subclasses or grouped into superclasses. These problems are known as hierarchical classification problems in the ML literature.

There are two main ways in which the classes may be hierarchically disposed: as a tree or as a Directed Acyclic Graph (DAG). The main difference between the tree structure (Figure 1.a) and the DAG structure (Figure 1.b) is that, in

the tree structure, each node has just one parent node, while in the DAG each node may have more than one parent. For both flat and hierarchical classification schemes, the nodes represent the problem classes and the root node corresponds to "any class", denoting a total absence of knowledge about the class of an object. Hierarchical classification problems often have as objective the classification of a

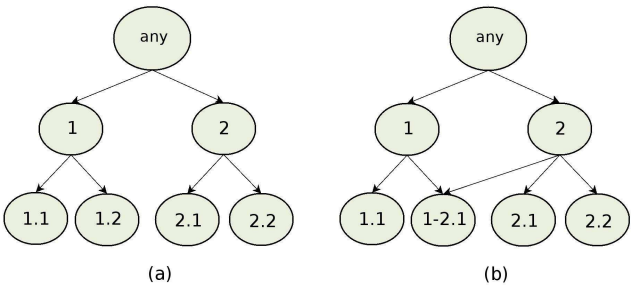

Fig. 1. Examples of hierarchies of classes: (a) structured as a tree and (b) structured as a DAG.

new input data into one of the leaf nodes. The deeper the class in the hierarchy, the more specific and useful is its associated knowledge. It may be the case, however, that the classifier does not have the desired reliability to classify a data into deeper classes. In this case, it would be safer to perform a classification into shallower levels of the hierarchy.

In tree structures, the deeper the level, in general the more difficult is the class prediction phase. This may be due to the fact that the classes in deeper levels represent more specific information and are produced by models that have been induced from a smaller number of instances. Therefore, they are more difficult to predict. For DAG structures, the analysis is more complex. As a child node may have more than one parent, some classification models in deeper levels may have been induced from more instances than their ancestral. Besides, in practice, even for DAGs, the prediction accuracy rate decreases with the increase in the class level (depth) [1].

Herewith, the closer the predicted class is to the root of the class tree, the lower the classification error tends to be. On the other hand, such classification becomes less specific and, as a consequence, less useful. Therefore, a hierarchical classifier must deal with the trade-off class specificity versus classification error rate.

In some problems, all instances must be associated to classes in leaf nodes. These problems are named "mandatory leaf-node prediction problems". When this obligation does not hold, the classification problem is a "optional leaf-node prediction problem".

3 Classification Approaches for Hierarchical Problems

Following the nomenclature in [1], four types of approaches to deal with these problems may be cited: transformation of the hierarchical problem into a flat classification problem, hierarchical prediction with flat classification algorithms, Top-Down classification and Big-Bang classification.

Several solutions have been proposed for the induction of classification models for hierarchical problems. Many hierarchical classification works have been published in the last years, mainly related to text mining problems [7, 6]. Nevertheless, due to the inherent hierarchical characteristic of several biological problems, hierarchical classification has found in the Bioinformatics area a vast and promising exploration field.

Next, a brief description of the four types of hierarchical classification schemes is presented, along with works related to the protein function prediction problem.

Transformation of the Hierarchical Problem into a Flat Classification Problem

Although in a hierarchical problem the classes are hierarchically organized, this approach reduces the original hierarchical problem to a flat classification problem. This idea is supported by the fact that a flat classification problem may be viewed as a particular case of hierarchical classification, in which there are no subclasses and superclasses. Traditional approaches for flat classification may be applied in this context, without the need to perform alterations or adjustments.

Jensen et al. [8] describe a method, named ProtFun, which uses an ensemble of simple Neural Networks, with a single completely connected intermediate layer, to predict protein categories. The method predicts functional categories as originally defined by Riley [9] for Escherichia coli. The classification model described in Weinert and Lopes [10] is based on Multilayer Perceptron Neural Networks. The model was applied to the classification of functional and structural characteristics of enzymes from the Protein Data Bank [11].

Hierarchical Prediction with Flat Classification Algorithms

This approach divides a hierarchical problem into a set of flat classification problems. The main difference to the previous approach is the possibility to consider several levels of the hierarchy. In this approach, each class level is treated as an independent classification problem. For each level, flat classification algorithms may then be used.

Clare and King [12] describe a classification method that uses the C4.5 algorithm [13] and was applied to S.cerevisiae phenotype data. Jensen et al. describe in [14] an extension of the method proposed in [8], used in the prediction of GO (Gene Ontology) classes. Laegreid et al. [15] predicts GO classes using a combination of a rule induction algorithm based on the Rough-Set theory [16] and Genetic Algorithms [17]. The method described produces rules that model the

relation between the gene expression levels over time in order to predict the biological paper of unknown genes. Tu et al. [18] propose a classification model that uses Neural Networks to infer annotations for more specific classes from known annotations for their superclasses. As application, data from serum response in serum-starved human fibroblasts were used. Barutcuoglu et al. [19] described a method where Support Vector Machines (SVMs) [20] classifiers are trained independently for each class. A Bayesian hierarchical combination scheme is later used to allow error correction collaboration among all nodes.

Top-Down Hierarchical Classification

In the Top-Down approach, one or more classifiers are trained for each level of the hierarchy. This produces a tree of classifiers. The root classifier is trained with all training instances. Then, at the next class level, a classifier is training with just the subset of instances belonging to the classes predicted by the classifier. E.g, in the class tree of Fig. 1(a), a classifier associated with the class node "1" would be trained only with instances belonging to class 1.1 or 1.2, but its training set would not include instances of class 2.1 or 2.2. The process of training classifiers proceeds in a top-down fashion until classifiers predicting the leaf class nodes are produced. Hence, the top-down approach follows the well-known "divide-and-conquer" principle.

In the test phase, beginning at the root node, an instance is classified in a Top-Down manner. When assigned to one class, the instance is then submitted to a new classifier in order to predict to which of this class' subclasses it belongs. This procedure is repeated until a leaf-node class is reached or until no additional prediction can be made from an internal node, such that the reliability is not affected. As this approach performs the classification through a modular process, the classifier induction is simpler when compared to the Big-Bang approach, described next. In particular, although it produces a tree of classifiers, each classifier is built by running a flat classification algorithm. Nevertheless, its disadvantage is that errors made in higher levels of the hierarchy are propagated to the more specific levels.

Holden and Freitas [21] proposed a hybrid algorithm that combines characteristics of the PSO (Particle Swarm Optimization) [22] and ACO (Ant Colony Optimization) [23] techniques for the induction of rule-based classifiers in a Top-Down manner. The hybrid algorithm was employed for the classification of enzymes. In [24], Holden and Freitas used the same hybrid algorithm proposed in [21], with some extensions, for the classification of G-Protein-Coupled Receptors [25].

Big-Bang Hierarchical Classification

One can consider that truly hierarchical classification algorithms are instances of the Big-Bang and the Top-Down approaches [1]. In the Big-Bang approach, a classification model is created in a single run of the algorithm, considering

the hierarchy of classes as a whole, presenting then a higher algorithmic complexity. After the classification model training, the prediction of the class of a new instance is carried out in just one step. For this reason, in contrast to the other approaches, Big-Bang cannot use pure flat classification techniques. If a flat classification technique is used in the Big-Bang approach, it must be adapted to consider the whole hierarchy.

Clare and King [26] modified the C4.5 learning algorithm to predict functional classes from S.cerevisiae ORFs (Open Reading Frames). Blockeel et al. [27] used a Decision Tree induction algorithm based on the notion of predictive clustering trees [28]. The algorithm generates as output one tree for the whole hierarchy of classes. Phenotype S.cerevisiae data were used in the experiments.

4 Materials and Methods

4.1 Datasets

The datasets used in this paper employ signatures (describing sequence similarity) generated directly from protein sequences to attempt to predict a given protein's function. The two datasets used in this paper involve G-Protein-Coupled Receptor (GPCR) and Enzyme protein families.

G-protein-coupled receptors are proteins involved in signalling. They span cell walls so that they influence the chemistry inside the cell by sensing the chemistry outside the cell. More specifically, when a ligand (a substance that binds to a protein) is received by a GPCR, it causes the attached G-proteins to activate and detach. This is a mechanical biological switch that causes the released G-Protein to affect other reactions within the cell. This kind of protein is particularly important for medical applications because it is believed that $40\% - 50\%$ of current medical drugs target GPCR activity [29].

Enzymes are another subset of proteins; they are catalysts which are used to speed up and make possible many of the chemical reactions that take place within the cell, without being altered themselves during the reaction. They are usually very specific, only catalysing one type of reaction within the cell. Often they can be turned on and off by a ligand (a small molecule that interacts with the enzyme). This is used to control both the speed of reaction and the course of overall reaction pathways that take place within the cell.

The protein functional classes are given unique hierarchical indexes by [25] in the case of GPCRs and by Enzyme Commission Codes [3] in the case of enzymes. In the case of GPCRs, proteins (data instances) have up to five class levels, but only four levels are used in the datasets created in this work, as the data in the $5th$ level is too sparse for training - i.e., in general there are too few instances of each class at the $5th$ level. All four levels of the Enzyme Commission Codes are used in the created Enzymes datasets.

The datasets used in our experiments were constructed from data extracted from UniProt [30] and GPCRDB [25]. UniProt is a well known biological database, containing sequence data and a rich annotation about a large number of different

kinds of proteins. It also has cross-references for other major biological databases. UniProt was extensively used in this work as a source of data for creating the datasets used in the experiments. Only the UniProtKB/Swiss-Prot was used as a data source, as it contains a higher quality, manually annotated set of proteins. Unlike Uniprot, GPCRDB is a biological database specialised on GPCR proteins.

The predictor attributes in the two datasets are Interpro entries [31, 32], along with the molecular weight and sequence length of each protein. Interpro integrates several protein signature databases (Gene3D, PANTHER, PIRSF, Pfam, PRINTS, PROSITE, SMART, SUPERFAMILY and TIGRFAM) giving a very powerful "representation language" to describe the main patterns or "motifs" (e.g., specific sub-sequences of amino acids) present in a given protein or group of proteins. The component protein signature databases from which Interpro entries are derived use three main methods of protein identification: PROSITE uses regular expressions, PRINTS uses groups of non-overlapping motifs and the rest rely on Hidden Markov Model methods.

Any duplicate instances (proteins) in a dataset are removed in a preprocessing step, i.e., before the hierarchical classification algorithm is run, to avoid redundancy. For both GPCR and Enzyme datasets, if there are fewer than ten instances in any given class in the class tree that class is merged with its parent class. If the parent class is the root node, the entire small class is removed from the dataset. This process helps to ensure there is enough instances per class to allow the classifier to perform a reasonably reliable prediction of each class. Any binary attribute that has a value which occurs in only one instance is removed from the corresponding dataset, since these binary attributes in general do not have a good predictive power. An initial random sample of 15000 enzymes from the UniProt database was used to generate the enzyme datasets. Less than the original 15000 instances occur in the final datasets because of the duplicate and small class removal process.

After preprocessing the datasets used in the experiments, the GPCR dataset ended up with 450 predictor attributes, 7461 instances (proteins) and 12/54/82/50 classes per level (number of classes at level 1/2/3/4, respectively). The Enzyme dataset presented 1216 predictor attributes, 14036 instances and 6/41/96/187 classes per level. Due to a high computational cost, the Enzyme dataset was reduced to 6925 instances and 2/21/48/87 classes per level.

Both datasets were divided according to the 5-fold cross-validation method. Accordingly, each dataset is divided into five parts of approximately equal size. At each round, one fold is left for test and the remaining folds are used in the classifiers training. This makes a total of five train and test sets. The final accuracy rate of a classification model is then given by the mean of the predictive accuracy on the test set obtained for each fold.

4.2 Decision trees

A Decision Tree (DT) is a data structure containing two types of nodes, namely: a leaf node that corresponds to a class or a decision node that contains a test

over some attribute. For each test result, there is an edge for a subtree. In the classification of a new instance in the DT, the tree is traversed according to the tests' results in a top-down fashion until a leaf node is reached. The instance is labeled with the class associated to this node. Examples of DT induction algorithms are the ID3 algorithm [4] and its successor C4.5 [13]. This work employed the C4.5 algorithm in the DT induction.

Besides being a practical method for concept learning from data [5], DT models have a high comprehensibility, that is, the knowledge acquired by the tree during its training is easy to understand and interpret. These were the motivations for the choice of this particular technique for classifier induction in this work.

4.3 Hierarchical Classification models

The four hierarchical methods described in Section 3 are compared in this work for the protein classification problems investigated. The first considers only the leaf nodes of the problem hierarchy, inducing a flat classifier that distinguishes all classes associated to this set. The idea is that the classification of a new instance in a class associated with a leaf node also implies in its classification in classes at higher (shallower) levels of the tree. E.g, if an instance is classified as 2.1.3.4, then the instance is considered assigned to class 2 at the first level, class 2.1 at the second level, and so on, in order to compute the predictive accuracy per level reported later. The second approach decomposes the hierarchical problem into a set of flat classification problems, each one distinguishing all classes present in a level of the hierarchy. The third method uses the Top-Down approach and the last one, the Big-Bang approach. All of them use DT induction algorithms to produce the classification models. These approaches were chosen in order to compare different schemes for hierarchical classification.

The flat and Top-Down approaches were implemented using the package TREE of the R tool [33]. The Big-Bang approach used was the one developed by Clare and King [26]. This method uses a modified version of the C4.5 algorithm, called HC4.5. The original code of HC4.5 can automatically assign a new instance to a class in any level of the tree, depending on the characteristics of the data at each level. Since the goal of this paper is to do an experiment comparing the Big-Bang and other approaches in a way which is as fair and controlled as possible, we modified HC4.5, including the restriction that it always assigns a new instance to a class in a leaf node of the class tree. This automatically assigns to the instance classes at higher levels of the class tree too.

5 Experiments

Experiments were performed in order to evaluate the hierarchical classification methods described in Section 4.3 using the datasets from Section 4.1.

Results

The mean accuracy results obtained in the GPCR dataset 5-fold cross-validation partitions are shown in Table 1. This table shows, for each level of the GPCR hierarchy, the mean accuracy rates of the hierarchical classifiers induced. The standard deviation rates of the accuracies obtained in the cross-validation data partitions are illustrated in parentheses. The accuracy rate corresponds to the percentage of correctly classified patterns in a dataset.

	Flat Classif. based on leaves	Flat Classif. all levels	Top-Down	Big-Bang
Level 1	61.33 (0.62)	87.80 (0.37)	87.80 (0.37)	91.13 (0.97)
Level 2	57.11 (0.54)	68.64 (0.43)	74.12 (0.65)	76.05 (1.69)
Level 3	21.97 (0.29)	29.22 (0.54)	46.17 (2.12)	43.38 (1.01)
Level 4	31.36 (1.28)	58.17 (2.73)	73.60 (4.46)	68.02 (4.96)

Table 1. Accuracy results in the GPCR dataset

Like Table 1, Table 2 shows the mean and standard deviation accuracy results observed for the Enzyme dataset partitions.

	Flat Classif. based on leaves	Flat Classif. all levels	Top-Down	Big-Bang
Level 1	82.73 (1.22)	89.78 (0.85)	89.78 (0.85)	88.97 (0.36)
Level 2	61.82 (1.03)	60.33 (1.98)	73.75 (1.34)	84.56 (0.84)
Level 3	58.24 (1.08)	53.79 (2.68)	61.38 (1.24)	84.13 (0.82)
Level 4	59.17 (1.48)	58.93 (0.66)	59.93 (0.13)	96.36 (0.43)

Table 2. Accuracy results in the Enzyme dataset

Discussion

The high performance obtained by all approaches in the first level of the EC dataset, shown in Table 2, occurred because the first level of this dataset has only 2 classes, different from the GPCR dataset, which presents 12 classes in the first level.

According to the results showed in tables 1 and 2, the Top Down and Big Bang approaches performed better than the flat approaches for all levels in both datasets. This was expected, once the Top Down and the Big Bang approaches consider the hierarchy during their training and test. This makes the prediction in deeper levels easier. For the flat approaches, the accuracy tends to decrease faster than the hierarchical approaches with the increase of the levels depth.

For the GPCR dataset, the flat approach based on all levels performs significantly better than the flat approach based on the leaf nodes. For the EC dataset, none of the flat approaches is clearly superior to the other.

Regarding the hierarchical approaches, for the GPCR dataset, the Top Down algorithm has a lower accuracy than the Big Bang algorithm for the first two levels and a higher accuracy for the last two levels. For the EC dataset, the Big Bang is clearly better than the Top Down in the last three levels. This difference may be due to the different hierarchical structure and the class (and instances per class) distribution in the hierarchy of these datasets.

Regarding the class distribution, the GCPR dataset has a reduced number of classes, and instances, in the fourth level, when compared with the EC dataset. This occurs because all classes in the fourth level of the GCPR dataset are part of the subtree rooted in the class 1 of the first level. The other classes in the first level have descendents only in the levels 2 and 3. The fourth level has 50 of the total 198 classes. The EC dataset, in opposite, has 87 of the total 158 classes in the fourth level.

The unbalanced nature of the distribution of classes in GPCR dataset seems to favour the correct prediction in the last levels by the Top Down algorithm. A possible reason is the error propagation mechanism employed by this algorithm (see Section 3). Since several leaf nodes are in the intermediate levels of the hierarchy, the errors are not propagated to the deepest levels. Besides, as most of the last level classes are descendents of the class 1, which has the highest correct prediction rate, the propagation of errors to the descendents of this class are less severe.

The GPCR dataset has 1544 of instances in the fourth level, from a total of 7500, making 20.59% of the instances. The EC dataset, on the other hand, has 4887 of instances in the fourth level, from a total of 6995, making 69.86% of the instances. A similar situation occurs in the third level. We believe that the reduced number of classes and instances in the last levels harms the performance of the Big Bang algorithm. This hapens because, unlike the Top Down algorithm, which uses a divide-and-conquer mechanism for the classification in the leaf nodes, the Big Bang predictions are made directly in the leaf nodes. For the previous reason, the high number of instances in the last level of the EC dataset may have favoured the Big Bang algorithm, see Table 2.

6 Conclusions

In this paper, we presented a comparative study of hierarchical approaches based on decision trees. Four approaches for hierarchical classification were investigated. Two approaches based on flat classification, the Big Bang approach and the Top Down approach.

In order to evaluate the performance of these approaches, experiments were performed using two bioinformatics datasets, which are related with G-Protein-Coupled Receptor (GPCR) and Enzyme Protein (EC) families. According to the experimental results, the Top Down and the Big Bang approaches performed

better than the two flat approaches for all levels in both datasets. In the EC dataset, the Big Bang approach outperformed the Top Down approach in the last 3 levels. In the GPCR dataset, the Top Down approach was clearly superior in the last two levels.

For future work, the authors plan to investigate the performance of the hierarchical approaches when the deepest classification level assigned to each test instance is automatically defined by the system, without the restriction of always assigning one of the leaf classes to every test instance. Other hierarchical classification algorithms will also be investigated. Finally, the authors plan to combine hierarchical classification with multi-label classification.

Acknowledgments. The authors would like to thank the Brazilian research councils FAPESP and CNPq for their financial support and Amanda Clare for the Big-Bang code used in the experiments.

References

1. A. A. Freitas, A. C. P. F. Carvalho, A Tutorial on Hierarchical Classification with Applications in Bioinformatics. In: D. Taniar (Ed.) Research and Trends in Data Mining Technologies and Applications, Idea Group, 2007, pp. 176–209.
2. J. Blake, Gene Ontology(GO) Tutorial, [Online; accessed April 07, 2006] (2003). URL `http://www.geneontology.org/teaching_resources/tutorials/2003_MBL_jblake.pdf`
3. E. Nomenclature, of the IUPAC-IUB, American Elsevier Pub. Co., New York, NY (1972) 104.
4. J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.
5. T. M. Mitchell, Machine Learning, McGraw-Hill Higher Education, 1997.
6. A. Sun, E. P. Lim, W. K. Ng, Hierarchical text classification methods and their specification, Cooperative Internet Computing 256 (2003) 18 p.
7. A. Sun, E. P. Lim, Hierarchical text classification and evaluation, in: Proceedings of the 2001 IEEE International Conference on Data Mining, IEEE Computer Society Washington, DC, USA, 2001, pp. 521–528.
8. L. J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, H. H. Stærfeldt, K. Rapacki, C. Workman, C. A. F. Andersen, S. Knudsen, A. Krogh, A. Valencia, S. Brunak, Prediction of human protein function from post-translational modifications and localization features, Journal of Molecular Biology 319 (5) (2002) 1257–1265.
9. M. Riley, Functions of the gene products of Escherichia coli., Microbiology and Molecular Biology Reviews 57 (4) (1993) 862–952.
10. W. R. Weinert, H. S. Lopes, Neural networks for protein classification, Applied Bioinformatics 3 (1) (2004) 41–8.
11. F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, M. Tasumi, The Protein Data Bank. A computer-based archival file for macromolecular structures, FEBS Journal 80 (2) (1977) 319–324.
12. A. Clare, R. D. King, Knowledge Discovery in Multi-label Phenotype Data, in: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery, Vol. 2168 of Lecture Notes in Computer Science, Springer, 2001, pp. 42–53.

13. J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
14. L. J. Jensen, R. Gupta, H. H. Stærfeldt, S. Brunak, Prediction of human protein function according to Gene Ontology categories, Bioinformatics 19 (5) (2003) 635–642.
15. A. Laegreid, T. R. Hvidsten, H. Midelfart, J. Komorowski, A. K. Sandvik, Predicting Gene Ontology Biological Process From Temporal Gene Expression Patterns, Genome Research 13 (5) (2003) 965–979.
16. Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers Norwell, MA, USA, 1992.
17. M. Mitchell, An Introduction to Genetic Algorithms, Mit Press, 1996.
18. K. Tu, H. Yu, Z. Guo, X. Li, Learnability-based further prediction of gene functions in Gene Ontology., Genomics 84 (6) (2004) 922–928.
19. Z. Barutcuoglu, R. E. Schapire, O. G. Troyanskaya, Hierarchical multi-label prediction of gene function, Bioinformatics 22 (7) (2006) 830–836.
20. N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000.
21. N. Holden, A. A. Freitas, A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data, in: Proceedings of the 2005 IEEE Swarm Intelligence Symposium, 2005, pp. 100–107.
22. T. Sousa, A. Silva, A. Neves, Particle swarm based Data Mining Algorithms for classification tasks, Parallel Computing 30 (5-6) (2004) 767–783.
23. R. S. Parpinelli, H. S. Lopes, A. A. Freitas, Data mining with an ant colony optimization algorithm, IEEE Transactions on Evolutionary Computation 6 (4) (2002) 321–332.
24. N. Holden, A. A. Freitas, Hierarchical Classification of G-Protein-Coupled Receptors with PSO/ACO Algorithm, in: Proceedings of the 2006 IEEE Swarm Intelligence Symposium, 2006, pp. 77–84.
25. GPCRDB, Information system for G protein-coupled receptors (GPCR), [Online; accessed July-2006] (2006).
URL `URLhttp://www.gpcr.org/7tm/`
26. A. Clare, R. D. King, Predicting gene function in Saccharomyces cerevisiae, Bioinformatics 19 (90002) (2003) 42–49.
27. H. Blockeel, M. Bruynooghe, S. Dzeroski, J. Ramon, J. Struyf, Hierarchical multi-classification, in: Proceedings of the ACM SIGKDD 2002 Workshop on Multi-Relational Data Mining (MRDM 2002), 2002, pp. 21–35.
28. H. Blockeel, L. D. Raedt, J. Ramon, Top-down induction of clustering trees, in: Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp. 55–63.
29. D. Filmore, It's a GPCR world, Modern drug discovery 1 (17) (2004) 24–28.
30. R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, et al., UniProt: the Universal Protein knowledgebase, Nucleic Acids Research 32 (2004) D115–D119.
31. Interpro, [Online; accessed July-2006] (2006).
URL `http://www.ebi.ac.uk/interpro/`
32. J. McDowall, InterPro: Exploring a Powerful Protein Diagnostic Tool, in: ECCB05, Tutorial, 2005, p. 14 p.
33. W. N. Venables, D. M. Smith, the R Development Core Team, An introduction to R - version 2.4.1, `http://cran.r-project.org/doc/manuals/R-intro.pdf` (2006).