# Top-Down Hierarchical Ensembles of Classifiers for Predicting G-Protein-Coupled-Receptor Functions

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

[1]Depto. Ciências de Computação
ICMC/USP - São Carlos - Caixa Postal 668
13560-970 - São 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 and Centre for BioMedical Informatics
University of Kent, Canterbury, CT2 7NF, UK
a.a.freitas@kent.ac.uk

**Abstract.** Despite the recent advances in Molecular Biology, the function of a large amount of proteins is still unknown. An approach that can be used in the prediction of a protein function consists of searching against secondary databases, also known as signature databases. Different strategies can be applied to use protein signatures in the prediction of function of proteins. A sophisticated approach consists of inducing a classification model for this prediction. This paper applies five hierarchical classification methods based on the standard Top-Down approach and one hierarchical classification method based on a new approach named Top-Down Ensembles - based on the hierarchical combination of classifiers - to three different protein functional classification datasets that employ protein signatures. The algorithm based on the Top-Down Ensembles approach presented slightly better results than the other algorithms, indicating that combinations of classifiers can improve the performance of hierarchical classification models.

## 1 Introduction

Proteins are large organic compounds that perform almost all the functions related to cell activity, such as biochemical reactions, cell signaling, structural and mechanical functions. These large molecules consist of long sequences of amino acids, which fold into specific structures so that the protein can function properly.

In functional genomic, an important problem is the prediction of the function of proteins. Due to the recent advances in Molecular Biology methods and the consequent generation of biological data in large scale, data analysis has become a central issue for the investigation of proteins whose functions are unknown.

An approach that can be used in the prediction of a protein function involves searching against secondary databases, also known as signature databases. These databases contain results of analysis performed in primary databases, which contain linear sequences of amino acids, and can be used to verify the presence of particular patterns in the query proteins. These patterns represent information about conserved motifs in proteins, which are frequently useful to help the prediction of protein functions. Protein signatures can be used to assign a query protein to a specific family of proteins and thus to formulate hypotheses about its function [1]. Examples of signature databases include InterPro [2], Prosite [3], Pfam [4] and Prints [5].

Different strategies can be applied to use protein signatures in the prediction of function of proteins. A sophisticated approach consists of inducing a classification model for this prediction. Accordingly, each protein is represented by an attribute set, describing the presence or absence of patterns in the protein, and a learning algorithm captures the most important relationships between the attributes and the classes involved in the classification problem.

In the context of prediction of protein function, a classification model needs to be induced according to a special kind of classification problem named hierarchical classification, since protein functional data is inherently hierarchical (for example, the Enzyme Commission hierarchy [6]).

In this paper, three protein function datasets are analyzed - each one employing one different kind of protein signature - for a comparative study among six hierarchical classification algorithms. The algorithm based on the Top-Down Ensembles approach - a variation of the Top-Down approach that uses combination of classifiers for the induction of the classification model - presented better results across the three different kinds of protein signatures - Prosite, Pfam and Prints. The main contribution of this paper is to show that combinations of classifiers can improve the performance of hierarchical classification models, a result that was consistent even for different types of protein signatures.

The paper is organized as follows: Section 2 introduces important concepts of hierarchical classification; Section 3 introduces the Top-Down Ensembles approach; 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 from this work.

## 2 Hierarchical Classification

Classification is one of the most important problems in Machine Learning (ML) and Data Mining (DM) [7]. Given a dataset composed of $n$ pairs $(\mathbf{x}_i, y_i)$, where each $\mathbf{x}_i$ is a data item (example) and $y_i$ represents its class, a classification algorithm must find a function, through a training or adjustment phase, which maps each data item to its correct class.

Conventional classification problems involve a finite (and usually small) set of flat classes. Each example is assigned to a class out of this set, in which the classes do not have direct relationships to each other, such as subclass and superclass relationships. For this reason, these classification problems are named flat classification problems. Nevertheless, there are more complex classification problems where the classes to be predicted are hierarchically related [8–10]. These problems are known in the ML literature as hierarchical classification problems.

The classes involved in a hierarchical classification problem can be disposed either as a tree or as a Directed Acyclic Graph (DAG). The main difference between these structures is that, in the tree structure (Figure 1.a), each node has just one parent node, while in the DAG structure (Figure 1.b), each node may have more than one parent. The nodes represent the classes involved in the classification problem and the root node corresponds to "any class", denoting a total absence of knowledge about the class of an object.

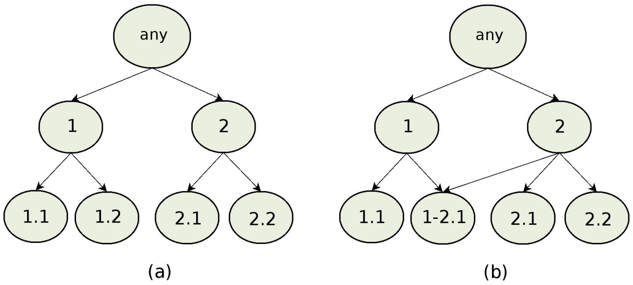

**Fig. 1.** Examples of hierarchies of classes: (a) Tree and (b) DAG.

The deeper the class in the hierarchy, the more specific and useful is its associated knowledge in the classification of a new data item. Hierarchical classification problems often have as objective to assigning a data item into one of the leaf nodes. It may be the case, however, that the classifier does not have the desired reliability to classify a data item into deeper classes. In this case, it would be safer to perform a classification into higher levels of the hierarchy. When all examples must be associated to classes in leaf nodes, the classification problem is named "mandatory leaf node prediction problem". When this obligation does not hold, the classification problem is an "optional leaf node prediction problem".

A simple approach to deal with a hierarchical classification problem consists of reducing it into one or more flat classification problems. This reduction is possible because a flat classification problem may be viewed as a particular case of hierarchical classification, in which there are no subclasses and superclasses.

However, the main disadvantage of this approach is to ignore the hierarchical relationships among the classes, which can provide valuable information for the induction of a classification model. Two more sophisticated approaches that consider these relationships are the Top-Down and Big-Bang approaches [8].

The Top-Down approach uses the "divide and conquer" principle to induce the classification model. The main idea of this approach is to produce one or more classifiers for each node of the hierarchy. Initially, a classifier is induced for the root node using all training examples in order to distinguish among the classes at the first level of the hierarchy. At the next class level, each classifier is trained with just a subset of the examples. As an example, consider the class tree of Fig. 1(a). In this structure, a classifier associated with class node 1 would be induced only with data belonging to classes 1.1 and 1.2, ignoring objects from classes 2.1 and 2.2. This process proceeds until classifiers predicting the leaf class nodes are produced. At the end of this training phase, a tree of classifiers is obtained. Although this approach considers the hierarchical relationships between the classes, each classifier is built by a flat classification algorithm. In the test phase, beginning at the root node, an example is classified in a top-down manner, according to the predictions produced by a classifier in each level. An inherent disadvantage of this approach is that errors made in higher levels of the hierarchy are propagated to the most specific levels.

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 induction, 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.

In this paper, only Top-Down algorithms were considered. The aim of the experiments were to compare standard Top-Down algorithms with an algorithm based on a variation of the Top-Down approach, described in the next section.

## 3 The Proposed Top-Down Ensembles Approach

A possible extension of the Top-Down approach consists of using various classifiers in each node of the tree of classifiers, instead of using just one classifier. This can be carried out through the combination of classifiers. This new approach was named Top-Down Ensembles.

Combination methods, also known as ensemble methods, use a set of classifiers to obtain the output (prediction) of the classification model [11]. The main idea behind these methods is to induce various classifiers, also named base classifiers, from the training data. In the test phase, the output for each unseen example is given by the combination of the outputs of the base classifiers.

For the combination of the outputs of the base classifiers, the strategy employed in this paper was to train a meta-classifier to perform this task. Initially, all base classifiers are trained by using training examples. A new training dataset is then produced, in which the input attributes are the outputs of the base classifiers. One alternative to generate this new training data consists of using the original training data as input for the base classifiers and storing the outputs produced by them. These outputs, along with the true class (expected output) for each example, are used to generate the new training dataset. This dataset is then used to induce the meta-classifier, which is in charge of combining the outputs from the base classifiers. In the test phase, the examples are given as inputs for each base classifier and the outputs of these classifiers are given as inputs of the meta-classifier, which performs the final classification.

The main motivation for exploiting Top-Down algorithms based on ensemble methods is the advantage of using the combined power of several techniques instead of choosing just one of them to induce the classifier in each node of the class hierarchy.

## 4 Materials and Methods

This section presents the materials and methods employed in the experiments.

### 4.1 Datasets

Three datasets involving G-Protein-Coupled Receptors (GPCRs) were used in the experiments reported in this paper. In each dataset, the GPCR sequences were described through one kind of protein signature, allowing the comparison of the results of an algorithm across three different protein signatures - Prosite, Pfam and Prints. These datasets were first proposed in [12], but they were modified for the purpose of our experiments, as explained later.

GPCRs are particularly important for medical applications due to the important influence of this type of protein in the chemical reactions within the cell. According to [13], 40% to 50% of current medical drugs interact with GPCRs. The protein functional classes of GPCR are given by unique hierarchical indexes in the GPCRDB [14]. The GPCR classes are arranged in the structure of a tree, with four levels - where the top-level refers to generic classes, which are divided into sub-classes, and so on, up to the fourth level.

In essence, the protein signatures used in the datasets have the following characteristics. Prosite signatures are regular expressions or patterns describing short fragments of protein sequences that can be used to identify protein domains, families and functional sites. Currently, the Prosite database stores patterns and profiles specific for more than a thousand protein families or domains. Each of these signatures comes with documentation providing background information on the structure and function of these proteins [15]. Pfam signatures are based on multiple alignments and Hidden Markov Models (HMMs), which consider probability theory methods, allowing a direct statistical approach to identify and score matches. Prints signatures are based on a pattern recognition approach named “fingerprinting”. Such signatures use several motifs to identify an unknown protein rather than just one motif. This renders fingerprinting a powerful diagnostic technique, because there is a higher chance of identifying a distant relative, even though mismatches with some motifs may have occurred.

The three datasets were constructed from data extracted from UniProt [16], a well-known protein database, and GPCRDB [14], a database specialised on GPCR proteins. In each of the three datasets, each protein signature was encoded as a binary attribute, where 1 indicates the presence of a protein signature and 0 its absence. Additionally, all datasets contain the attributes "molecular weight" and "sequence length".

Besides the preprocessing steps explained in [12], another preprocessing step was included because a small subset of data belonged only to internal nodes of the hierarchy. As the developed algorithms consider mandatory leaf node prediction, some problems could take place during the evaluation of the classification model. Suppose that an example belonging to an internal node was classified into a class represented by a leaf node. During the evaluation, it would not be possible to answer whether the prediction to the more specific node was successful or not. Therefore, examples belonging to internal nodes were not used in the experiments.

In Table 1, the configuration obtained after preprocessing of the three datasets, regarding the total number of examples, number of predictor attributes and number of classes per level (number of classes at level 1/2/3/4, respectively), are shown. As can be noticed in the table, the fourth level of the class hierarchies contain less classes than the third levels. It occurs because of the presence of several leaf nodes in the third level of these hierarchies. Some leaf-nodes are also present in the first and second levels.

**Table 1.** Total number of examples, number of predictor attributes and number of classes per level (number of classes at level 1/2/3/4, respectively) of the three datasets used in the experiments.

| | Examples | Attributes | Classes per level |
|---|---|---|---|
| Prosite | 5728 | 127 | 9/50/79/49 |
| Pfam | 6524 | 73 | 12/52/79/49 |
| Prints | 4880 | 281 | 8/46/76/49 |

All datasets were divided according to the 5-fold cross-validation methodology. 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 given by the mean of the predictive accuracies on the test sets from cross-validation.

### 4.2 Top-Down hierarchical classification techniques

For developing the algorithm based on the hierarchical combination of classifiers, five different ML techniques selected, following distinct learning paradigms: Decision Trees [17], induced with the C4.5 algorithm [18]; Sets of Rules induced by the RIPPER algorithm [19]; Support Vector Machines (SVMs) [20]; K-nearest neighbors (KNN) [21]; and Bayesian Networks (BayesNet) [22]. In order to combine the outputs of the base classifiers, another classifier was used. For each node of the class hierarchies, the technique that induces the meta-classifier is chosen among the ML techniques used to produce the base classifiers - the five ML techniques previously mentioned. The adopted criterion consists of selecting the technique whose classifier presents the highest accuracy for the original training set.

In order to compare the results from the algorithm based on the Top-Down Ensembles approach with the other algorithms, the experiments also included five standard Top-Down hierarchical algorithms: one algorithm for each one of the five ML techniques employed in the hierarchical combination of classifiers.

All the Top-Down algorithms were implemented using packages from the R tool [23]. The following packages were used: e1071 [24] and RWeka [25]. The package e1071 was used to generate classifiers based on SVMs. The package RWeka was used to generate classifiers for the other ML techniques. The default parameters were adopted for all techniques, except for SVM. For this technique, two parameters were modified: the cost was set to 100 and $\gamma$ in the Gaussian Kernel was set to 0.01. These values were adopted because they are often used in previous works involving SVMs, presenting good results. Besides, the continuous attributes were normalized before their use by SVMs. For the other techniques, the normalization was not necessary, either because this procedure does not affect their results or because the technique internally implements this procedure.

### 4.3 Evaluation of the classification models

The evaluation of the classification models was carried out level by level in the classification hierarchy. For each hierarchical level, a value resulting from the evaluation of the predictive performance in the level is reported through a measure called depth-dependent accuracy. This measure is based on an approach of attributing misclassification costs proposed in [26].

This approach takes into account that classes closer in the hierarchy tend to be more similar to each other than classes more distant, and that predictions in deeper levels are more difficult. Thus, misclassification costs for classes more distant are higher than misclassification costs for classes closer to each other, and misclassification costs in the shallower levels are higher than in the deeper levels. Accordingly, weights are attributed to the edges of the class tree and the misclassification costs are defined as the shortest weighted path between the true class and the predict class.

In the calculation of the depth-dependent accuracy, the misclassification cost of each prediction is initially estimated through the division of the shortest weighted path between the true class and the predicted class by the value of the farthest weighted path from the node that represents the true class (i.e, the more distant class). After calculating the normalized distance for each misclassification (for each test example), an average of all normalized distances is obtained. This average is the error rate of the classification model. Once the error rate is obtained, the accuracy is defined by the complement of this value. The final accuracy rate of the classification model is then given by the mean of the predictive depth-dependent accuracies on the test sets generated by using 5-fold cross-validation.

The weights used in the edges of the hierarchy for calculating the depth-dependent accuracy were: (0.26,0.13,0.07,0.04), where 0.26 is the weight of an edge between the root node and any of its subclasses (i.e, the classes of the first level), 0.13 is the weight of an edge between a class in the first level and any of its subclasses, and so on. These weights were used originally in [12].

Statistical tests were employed in order to verify statistical significances (at 95% of confidence level) among the results from the several hierarchical classification models induced. The statistical test employed was the corrected t-Student for paired data, which considers the differences of results between pairs of classifiers in the cross-validation test sets [27]. As multiple comparisons are performed, the significance level of the tests was adjusted with the Bonferroni correction strategy [28], so the level of significance was set to 1%.

## 5 Experiments

Experiments were performed in order to evaluate the hierarchical classification methods described in Section 4.2 using the datasets described in Section 4.1.

### 5.1 Results

The results obtained for the investigated algorithms in the GPCR datasets are illustrated in tables 2, 3 and 4. These tables show, for each level of the GPCR hierarchy, the mean depth-dependent accuracy rates of the hierarchical classifiers for the 5-fold cross-validation partitions. The standard deviation rates of the accuracies obtained in the cross-validation data partitions are shown in parentheses. The best results for each dataset and hierarchy level are highlighted in boldface.

### 5.2 Discussion

It can be observed from tables 2 to 4 that TD-Ens in general performed better for all levels of the three datasets employed. Only in two cases out of twelve TD-KNN showed a higher accuracy value. These results show that the Top-Down Ensembles approach may be considered promising and that combinations of classifiers can improve the performance of hierarchical classification models.

| TD-KNN | TD-C4.5 | TD-SVM | TD-RIPPER | TD-BayesNet | TD-Ens |
|---|---|---|---|---|---|
| 88.06 (0.51) | 87.92 (0.51) | 84.37 (0.28) | 86.70 (0.69) | 85.00 (0.88) | **88.35 (0.94)** |
| 82.68 (0.65) | 82.36 (0.60) | 77.83 (0.36) | 80.24 (0.78) | 78.37 (0.86) | **82.86 (0.86)** |
| 76.99 (0.52) | 76.68 (0.68) | 70.52 (0.31) | 73.53 (0.87) | 71.88 (0.44) | **76.83 (0.68)** |
| **73.40 (0.41)** | 72.31 (1.26) | 63.77 (0.65) | 70.63 (1.69) | 66.80 (0.83) | 72.73 (0.61) |

**Table 2.** Mean depth-dependent accuracy results in the GPCR dataset that employs Prosite signatures

| TD-KNN | TD-C4.5 | TD-SVM | TD-RIPPER | TD-BayesNet | TD-Ens |
|---|---|---|---|---|---|
| 92.90 (0.50) | 92.66 (0.46) | 92.55 (0.24) | 91.74 (0.30) | 89.88 (0.71) | **93.01 (0.68)** |
| 86.34 (0.44) | 85.99 (0.62) | 82.69 (0.34) | 83.83 (0.51) | 81.16 (0.61) | **86.62 (0.74)** |
| 78.34 (0.56) | 78.03 (0.63) | 75.86 (0.37) | 75.20 (0.60) | 72.77 (0.72) | **78.48 (0.73)** |
| 70.05 (1.25) | 68.51 (1.08) | 57.85 (0.68) | 66.47 (1.00) | 61.25 (1.20) | **70.15 (1.19)** |

**Table 3.** Mean depth-dependent accuracy results in the GPCR dataset that employs Pfam signatures

| TD-KNN | TD-C4.5 | TD-SVM | TD-RIPPER | TD-BayesNet | TD-Ens |
|---|---|---|---|---|---|
| 92.52 (0.55) | 91.02 (0.54) | 91.74 (0.75) | 90.43 (0.22) | 86.78 (0.71) | **92.75 (0.57)** |
| 90.72 (0.66) | 88.78 (0.48) | 89.18 (0.84) | 87.38 (0.17) | 83.36 (0.79) | **90.96 (0.69)** |
| **86.25 (0.77)** | 84.11 (0.48) | 84.23 (0.53) | 82.28 (0.13) | 77.24 (1.02) | 86.18 (0.77) |
| 85.25 (1.40) | 81.35 (1.58) | 81.22 (2.26) | 78.10 (1.60) | 72.21 (1.31) | **85.35 (2.40)** |

**Table 4.** Mean depth-dependent accuracy results in the GPCR dataset that employs Prints signatures

Among the standard Top-Down algorithms, TD-KNN obtained better results than the other algorithms for all datasets.

Comparing statistically the results of the standard top-down hierarchical classifiers to those of TP-Ens, some differences were detected at 95% of confidence. For instance, TD-Ens was better than TD-BayesNet for all levels of all datasets. TD-Ens was better than TD-SVM for all levels of Prosite dataset, for levels two and three from Pfam dataset and for the last level of the Prints dataset. Compared to RIPPER, TD-Ens was better in levels two and three from Prosite dataset, in the third level of Pfam and in all levels of Prints dataset. TD-Ens was also better than TD-C4.5 for levels two and four from Prints dataset. No statistical difference was found between the results of TD-KNN and TD-Ens.

For all algorithms a decrease of performance may also be observed for deeper classes in the hierarchies. This behavior can be attributed to two facts: (1) the propagation of errors from general levels to the specific levels, a characteristics inherent to the Top-Down approach; and (2) the predictions in deeper levels are more difficult.

In an analysis of the predictions of the different classifiers obtained by each classification technique in the test phase, a low diversity of results was observed. In other words, the classifiers commit in general common hits and mistakes, that is, similar predictions. A diversity of predictions is important to improve the predictive performance of an ensemble of classifiers. Although the diversity between the classifiers was not large, it was still useful to improve the predictive performance of TD-Ens compared to the isolate algorithms.

Regarding the results in different datasets, all algorithms showed a similar predictive behavior in terms of accuracy rate. In general, all algorithms performed better for Prints dataset, followed by Pfam and Prosite, in this order. In datasets Pfam and Prints the predictive performances were close in the first layer, but this difference raises for the other levels. The worst results were obtained in Prosite dataset, except from its last level.

## 6 Conclusions

In this paper, we presented a comparative study of six hierarchical classification algorithms for different kinds of protein signatures - Prosite, Pfam and Prints. Five of the algorithms were developed according to the standard Top-Down approach, using the following ML techniques: C4.5, RIPPER, SVMs, KNN and BayesNet. The results from these algorithms were compared with the results of an algorithm based on a variation of the Top-Down approach named Top-Down Ensembles approach, which combines results from classifiers induced by the five ML techniques previously mentioned.

In order to evaluate the performance of these algorithms, experiments were performed using three bioinformatics datasets, which are related with G-Protein-Coupled Receptors (GPCRs). Each dataset was generated based on one of the three protein signatures considered in this work, allowing the comparison of the results of an algorithm across different kinds of protein signatures.

According to the experimental results, TD-Ens outperformed the other algorithms for all datasets, with some exceptions. Therefore, the results of the Top-Down Ensembles approach may be considered promising. This indicates that combinations of classifiers can improve the performance of hierarchical classification models. Among the standard Top-Down algorithms, TD-KNN obtained better results than the other algorithms for all datasets.

As the algorithms investigated in this work were developed to deal with class hierarchies structured as trees, strategies to extend them to the context of hierarchies structured as DAGs should be addressed in future research. Besides, the authors plan to investigate the performance of the hierarchical approaches for optional leaf node predictions, eliminating the restriction that the classifications occur in the leaf nodes only. The authors also plan to investigate the use of diversity measures for the selection of base classifiers in the Top-Down Ensembles approach. Finally, it would be of great interest to investigate the use of different kinds of protein signatures in the same dataset.

**Acknowledgments.** The authors would like to thank the Brazilian research councils FAPESP and CNPq for their financial support and Nicholas Holden for making available the GPCR datasets.

## References

1. E. B. Institute, Protein function, [Online; Available in http://www.ebi.ac.uk/2can/tutorials/function/; accessed March 07, 2008].
2. R. Apweiler, T. Attwood, A. Bairoch, A. Bateman, E. Birney, M. Biswas, P. Bucher, L. Cerutti, F. Corpet, M. Croning, et al., The InterPro database, an integrated documentation resource for protein families, domains and functional sites, Nucleic Acids Research 29 (1) (2001) 37–40.
3. C. Sigrist, L. Cerutti, N. Hulo, A. Gattiker, L. Falquet, M. Pagni, A. Bairoch, P. Bucher, PROSITE: A documented database using patterns and profiles as motif descriptors, Briefings in Bioinformatics 3 (3) (2002) 265–274.
4. A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. Eddy, S. Griffiths-Jones, K. Howe, M. Marshall, E. Sonnhammer, The Pfam Protein Families Database, Nucleic Acids Research 30 (1) (2002) 276–280.
5. T. Attwood, The PRINTS database: A resource for identification of protein families, Briefings in Bioinformatics 3 (3) (2002) 252–263.
6. E. Nomenclature, of the IUPAC-IUB, American Elsevier Pub. Co., New York, NY (1972) 104.
7. T. M. Mitchell, Machine Learning, McGraw-Hill Higher Education, 1997.
8. 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. 175–208.
9. A. Sun, E. P. Lim, W. K. Ng, Hierarchical text classification methods and their specification, Cooperative Internet Computing 256 (2003) 18 p.

10. A. Sun, E. P. Lim, W. K. Ng, Performance measurement framework for hierarchical text classification, Journal of the American Society for Information Science and Technology 54 (11) (2003) 1014–1028.
11. L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
12. 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.
13. D. Filmore, It's a GPCR world, Modern drug discovery 1 (17) (2004) 24–28.
14. GPCRDB, Information system for G protein-coupled receptors (GPCR), [Online; available in http://www.gpcr.org/7tm/; accessed July-2006].
15. S. I. of Bioinformatics, Prosite - description, [Online; Available in http://us.expasy.org/prosite/prosite_details.html; accessed March 01, 2008].
16. 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.
17. J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.
18. J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
19. W. Cohen, Fast effective rule induction, Proceedings of the Twelfth International Conference on Machine Learning (1995) 115–123.
20. N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000.
21. T. Cover, P. Hart, Nearest neighbor pattern classification, Information Theory, IEEE Transactions on 13 (1) (1967) 21–27.
22. N. Friedman, D. Geiger, M. Goldszmidt, Bayesian Network Classifiers, Machine Learning 29 (2) (1997) 131–163.
23. 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).
24. E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, A. Weingessel, e1071: Misc Functions of the Department of Statistics (e1071), TU Wien (2006) 1–5.
25. K. Hornik, A. Zeileis, T. Hothorn, C. Buchta, RWeka: An R Interface to Weka, R package version 0.2-14. URL http://CRAN. R-project. org.
26. 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.
27. C. Nadeau, Y. Bengio, Inference for the Generalization Error, Machine Learning 52 (3) (2003) 239–281.
28. S. Salzberg, On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach, Data Mining and Knowledge Discovery 1 (3) (1997) 317–328.