Hierarchical cost-sensitive algorithms for genome-wide gene function prediction

Nicolò Cesa-Bianchi and Giorgio Valentini

DSI, Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico 39, 20135 Milano, Italia
{cesa-bianchi,valentini}@dsi.unimi.it

Abstract. In this work we propose new ensemble methods for the hierarchical classification of gene functions. Our methods exploit the hierarchical relationships between the classes in different ways: each ensemble node is trained "locally", according to its position in the hierarchy; moreover, in the evaluation phase the set of predicted annotations is built so to minimize a global loss function defined over the hierarchy. We also address the problem of sparsity of annotations by introducing a cost-sensitive parameter that allows to control the precision-recall trade-off. Experiments with the model organism *S. cerevisiae*, using the FunCat taxonomy and 7 biomolecular data sets, reveal a significant advantage of our techniques over "flat" and cost-insensitive hierarchical ensembles.

1 Introduction

"In silico" gene function prediction can generate hypotheses to drive the biological discovery and validation of gene functions. Indeed, "in vitro" methods are costly in time and money, and the computational prediction can support the biologist in understanding the role of a protein or of a biological process, or in annotating a new genome at high level of accuracy, or more in general in solving problems in functional genomics.

Gene function prediction is a classification problem with the following distinctive features: (a) a large number of classes, with multiple functional annotations for each gene (a multiclass multilabel classification problem); (b) hierarchical relationships between classes governed by the "true path rule" [1]; (c) unbalance between positive and negative examples for most classes (sparse multilabels); (d) uncertainty of labels and incompleteness of annotations; (e) availability and need of integration of multiple sources of data.

This paper focuses on the three first items, proposing an ensemble approach for the hierarchical cost-sensitive classification of gene functions at genome and ontology-wide level. Indeed, in this context "flat" methods may introduce large inconsistencies in parent-child relationships between classes, and a hierarchical approach may correct "flat" predictions in order to improve the accuracy and the consistency of the overall annotations of genes [2]. We propose a hierarchical bottom-up Bayesian cost-sensitive ensemble that on the one hand respects the consistency of the taxonomy, and on the other hand exploits the hierarchical relationships between the classes. Our approach also takes into account the sparsity of annotations in order to improve the precision and the recall of the predictions. We also propose a simple variant of the hierarchical top-down algorithm that optimizes the decision threshold for maximizing the F-score.

Different research lines have been proposed for the hierarchical prediction of gene functions, ranging from structured-output methods, based on the joint kernelization of both input variables and output labels [3, 4], to ensemble methods, where different classifiers are trained to learn each class, and then combined to take into account the hierarchical relationships between functional classes [2, 5]. Our work goes along this latter line of research, and our main contribution is the introduction of a global cost-sensitive approach and the adaptation of a Bayesian bottom-up method to the hierarchical prediction of gene functions using the FunCat taxonomy [6].

Notation and terminology. We identify the N functional classes of the FunCat taxonomy with the nodes $i = 1, \ldots, N$ of a tree T. The root of T is a dummy class with index 0, which every gene belongs to, that we added to facilitate the processing. The FunCat *multilabel* of a gene is the nonempty subset of $\{1, \ldots, N\}$ corresponding to all FunCat classes that can be associated with the gene. We denote this subset using the incidence vector $\boldsymbol{v} = (v_1, \ldots, v_N) \in \{0,1\}^N$. The multilabel of a gene is built starting from the set of terms occurring in the gene's FunCat annotation. As these terms correspond to the most specific classes in T, we add to them all the nodes on paths from these most specific nodes to the root. This "transitive closure" operation ensures that the resulting multilabel satisfies the true path rule. Conversely, we say that a multilabel $\boldsymbol{v} \in \{0,1\}^N$ respects T if and only if $\boldsymbol{v}$ is the union of one or more paths in T, where each path starts from a root but need not terminate on a leaf. All the hierarchical algorithms considered in this paper generate multilabels that respect T. Finally, given a set of d features, we represent a gene with the normalized (unit norm) vector $\boldsymbol{x} \in \mathbb{R}^d$ of its feature values.

2 Methods

The HBAYES ensemble method [7, 8] is a general technique for solving hierarchical classification problems on generic taxonomies. The method consists in training a calibrated classifier at each node of the taxonomy. This is used to derive estimates $\widehat{p}_i(\boldsymbol{x})$ of the probabilities $p_i(\boldsymbol{x}) = \mathbb{P}\big(V_i = 1 \mid V_{\mathrm{par}(i)} = 1,\, \boldsymbol{x}\big)$ for all $\boldsymbol{x}$ and i, where $(V_1, \ldots, V_N) \in \{0,1\}^N$ is the vector random variable modeling the multilabel of a gene $\boldsymbol{x}$ and $\mathrm{par}(i)$ is the unique parent of node i in T. In order to enforce that only multilabels $\boldsymbol{V}$ that respect T should have nonzero probability, the base learner at node i is only trained on the subset of the training set including all examples $(\boldsymbol{x}, \boldsymbol{v})$ such that $v_{\mathrm{par}(i)} = 1$.

In the evaluation phase, HBAYES predicts the Bayes-optimal multilabel $\widehat{\boldsymbol{y}} \in \{0,1\}^N$ for a gene $\boldsymbol{x}$ based on the estimates $\widehat{p}_i(\boldsymbol{x})$ for $i = 1, \ldots, N$. Namely,

$\widehat{\boldsymbol{y}} = \operatorname{argmin}_{\boldsymbol{y}} \mathbb{E}\big[\ell_H(\boldsymbol{y},\boldsymbol{V}) \mid \boldsymbol{x}\,\big]$, where the expectation is w.r.t. the distribution of $\boldsymbol{V}$. Here $\ell_H(\boldsymbol{y},\boldsymbol{V})$ denotes the H-loss [7, 8], measuring a notion of discrepancy between the multilabels $\boldsymbol{y}$ and $\boldsymbol{V}$. The main intuition behind the H-loss is simple: *if a parent class has been predicted wrongly, then errors in its descendants should not be taken into account*. Given fixed *cost coefficients* $c_1,\dots,c_N > 0$, $\ell_H(\widehat{\boldsymbol{y}},\boldsymbol{v})$ is computed as follows: all paths in the taxonomy T from the root 0 down to each leaf are examined and, whenever a node $i \in \{1,\dots,N\}$ is encountered such that $\widehat{y}_i \neq v_i$, then c_i is added to the loss, while all the other loss contributions from the subtree rooted at i are discarded. As shown in [8], $\widehat{\boldsymbol{y}}$ can be computed via a simple bottom-up message-passing procedure whose only parameters are the probabilities $\widehat{p}_i(\boldsymbol{x})$.

We now describe a simple cost-sensitive variant, HBAYES-CS, of HBAYES, which is suitable for learning datasets whose multilabels are sparse. This variant introduces a parameter α that is used to trade-off the cost of false positive (FP) and false negative (FN) mistakes. We start from an equivalent reformulation of the HBAYES prediction rule

$$\widehat{y}_i = \operatorname*{argmin}_{y\in\{0,1\}} \left(c_i^- p_i(1-y) + c_i^+(1-p_i)y + p_i\{y=1\} \sum_{j\in\text{child}(i)} H_j \right) \qquad (1)$$

where $H_j = c_j^- p_j(1-\widehat{y}_j) + c_j^+(1-p_j)\widehat{y}_j + \sum_{k\in\text{child}(j)} H_k$ is recursively defined over the nodes j in the subtree rooted at i with each $\widehat{y}_j$ set according to (1), and $\{\,A\,\}$ is the indicator function of event A. Furthermore, $c_i^- = c_i^+ = c_i/2$ are the costs associated to a FN (resp., FP) mistake. In order to vary the relative costs of FP and FN, we now introduce a factor $\alpha \geq 0$ such that $c_i^- = \alpha c_i^+$ while keeping $c_i^+ + c_i^- = 2c_i$. Then (1) can be rewritten as

$$\widehat{y}_i = 1 \Longleftrightarrow p_i \left(2c_i - \sum_{j\in\text{child(i)}} H_j \right) \geq \frac{2c_i}{1+\alpha} \ .$$

This is the rule used by HBAYES-CS in our experiments.

Given a set of trained base learners providing estimates $\widehat{p}_1,\dots,\widehat{p}_N$, we compare the quality of the multilabels computed by HBAYES-CS with that of HTD-CS, a standard top-down hierarchical ensemble method with a cost sensitive parameter $\tau > 0$. The multilabel predicted by HTD-CS is defined by

$$\widehat{y}_i = \{\widehat{p}_i(\boldsymbol{x}) \geq \tau\} \times \{\widehat{y}_{\text{par}(i)} = 1\}$$

for $i = 1,\dots,N$ (we assume that the guessed label $\widehat{y}_0$ of the root of T is always 1). Note that both methods use the same estimates $\widehat{p}_i$. The only difference is in the way the classifiers are defined in terms of these estimates.

3 Experimental results

We predicted the functions of genes of the unicellular eukaryote *S. cerevisiae* at genome and ontology-wide level using the *FunCat* taxonomy [6] and 7 biomolecular data sets, whose characteristics are summarized in Tab. 1.

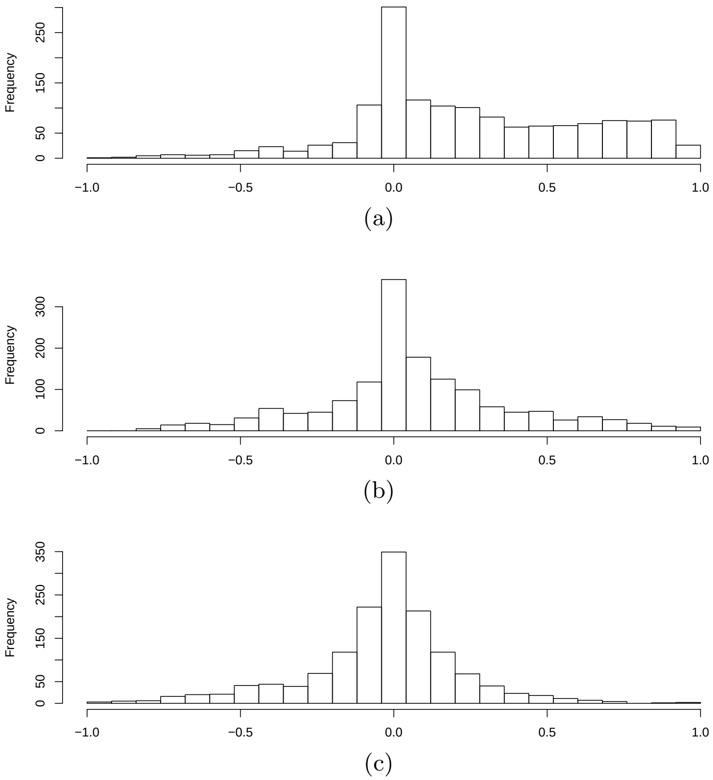
Table 1. Data sets

Data set	Description	num. of genes	num. of features	num. of classes
Pfam-1	protein domain binary data from *Pfam*	3529	4950	211
Pfam-2	protein domain log E data from *Pfam*	3529	5724	211
Phylo	phylogenetic data	2445	24	187
Expr	gene expression data	4532	250	230
PPI-BG	PPI data from *BioGRID*	4531	5367	232
PPI-VM	PPI data from von Mering experiments	2338	2559	177
SP-sim	Sequence pairwise similarity data	3527	6349	211

Pfam-1 data are represented as binary vectors: each feature registers the presence or absence of 4,950 protein domains obtained from the *Pfam* (Protein families) database [9]. Moreover, we also used an enriched representation of Pfam domains (Pfam-2) by replacing the binary scoring with log E-values obtained with the HMMER software toolkit [10]. The features of the phylogenetic data (Phylo) are the negative logarithm of the lowest E-value reported by BLAST version 2.0 in a search against a complete genome in 24 organisms [11]. The "Expr" data set merges the experiments of Spellman et al. (gene expression measures relative to 77 conditions) [12] with the transcriptional responses of yeast to environmental stress (173 conditions) by Gasch et al. [13]. Protein-protein interaction data (PPI-BG) have been downloaded from the *BioGRID* database, that collects PPI data from both high-throughput studies and conventional focused studies [14]. Data are binary: they represent the presence or absence of protein-protein interactions. We used also another data set of protein-protein interactions (PPI-VM) that collects binary protein-protein interaction data from yeast two-hybrid assay, mass-spectrometry of purified complexes, correlated mRNA expression and genetic interactions [15]. These data are binary too. The "SP-sim" data set contains pairwise similarities between yeast genes represented by Smith and Waterman log-E values between all pairs of yeast sequences [16].

In order to get a not too small set of positive examples for training, for each data set we selected only the FunCat-annotated genes and the classes with at least 20 positive examples. As negative examples we selected for each node/class all genes not annotated to that node/class, but annotated to its parent class. From the data sets we also removed uninformative features (e.g., features with the same value for all the available examples).

We used gaussian SVMs with probabilistic output [17] as base learners. Given a set $\widehat{p}_1, \ldots, \widehat{p}_N$ of trained estimates, we compared on these estimates the results of HTD-CS and HBAYES-CS ensembles with HTD (the cost-insensitive version of HTD-CS, obtained by setting $\tau = 1/2$) and FLAT (each classifier outputs its prediction disregarding the taxonomy). For HTD-CS we set the decision threshold τ by internal cross-validation of the F-measure with training data, while for HBAYES-CS we set the cost factor α to 5 in all experiments. This value provides a reasonable trade-off between between positive and negative examples, as shown by the plots in Figure 2. We compared the different ensemble methods using



(a)

(b)

(c)

Fig. 1. Histograms of the distribution of the normalized differences between F-measures across FunCat classes and data sets. (a) HBAYES-CS vs. FLAT ensembles; (b) HBAYES-CS vs. HTD ensembles; (c) HBAYES-CS vs. HTD-CS ensembles.

external 5-fold cross-validation (thus without using test set data to tune the hyper-parameters).

For the first set of experiments we used the classical F-score to aggregate precision and recall for each class of the hierarchy. Figure 1 shows the distribution, across all the classes of the taxonomy and the data sets, of the normalized differences $\frac{F_{\text{Bayes}} - F_{\text{ens}}}{\max(F_{\text{Bayes}}, F_{\text{ens}})}$ between the F-measure of HBAYES-CS and the F-measure of each one of the other ensemble methods. The shape of the distribution offers a synthetic visual clue of the comparative performances of the ensembles: values larger than 0 denote better results for HBAYES-CS. In Figure 1.(a) we can observe that HBAYES-CS largely outperforms FLAT, since most of the values are cumulated on the right part of the distribution. The comparison with HTD, Figure 1.(b), shows that HBAYES-CS on average improves on HTD, while essentially a tie is observed with HTD-CS —Figure 1.(c). Indeed the average F-measure across classes and data sets is 0.13 with FLAT ensembles, 0.18 with HTD and 0.22 and 0.23, respectively, with HBAYES-CS and HTD-CS ensembles.

Table 2. Left: Hierarchical F-measure comparison between HTD, HTD-CS, and HBAYES-CS ensembles. Right: win-tie-loss between the different hierarchical methods according to the 5-fold cross-validated paired t-test at 0.01 significance level.

Methods	Data sets							
	Pfam-1	Pfam-2	Phylo	Expr	PPI-BG	PPI-VM	SP-sim	Average
HTD	0.3771	0.0089	0.2547	0.2270	0.1521	0.4169	0.3370	0.2533
HTD-CS	0.4248	0.2039	0.3008	0.2572	0.3075	0.4593	0.4224	0.3394
HBAYES-CS	0.4518	0.2030	0.2682	0.2555	0.2920	0.4329	0.4542	0.3368

win-tie-loss		
Methods	HTD-CS	HTD
HBAYES-CS	2-4-1	6-1-0
HTD-CS	-	7-0-0

In order to better capture the hierarchical and sparse nature of the gene function prediction problem we also applied the *hierarchical F-measure*, expressing in a synthetic way the effectiveness of the structured hierarchical prediction [18]. In brief, viewing a multilabel as a set of paths, hierarchical precision measures the average fraction of each predicted path that is covered by some true path for that gene. Conversely, hierarchical recall measures the average fraction of each true path that is covered by some predicted path for that gene. Table 2 shows that the proposed hierarchical cost-sensitive ensembles outperform the cost-insensitive HTD approach. In particular, win-tie-loss summary results (according to the 5-fold cross-validated paired t-test [19] at 0.01 significance level) show that the hierarchical F-scores achieved by HBAYES-CS and HTD-CS are significantly higher than those obtained by HTD ensembles, while ties prevail in the comparison between HBAYES-CS and HTD-CS (more precisely 2 wins, 4 ties and 1 loss in favour of HBAYES-CS, Table 2, right-hand side). FLAT ensembles results with the hierarchical F-measure are not shown because they are significantly worse than those obtained with any other hierarchical method evaluated in these experiments.

Table 3 shows the per level F-measure results with Pfam-1 protein domain data and Pairwise sequence similarity data (SP-sim). Level 1 refers to the root

Table 3. Per level precision, recall, F-measure and accuracy comparison between FLAT, top-down (HTD), hierarchical top-down cost sensitive (HTD-CS), and hierarchical Bayesian cost sensitive (HBAYES-CS) ensembles. Top: Pfam protein domain data. Bottom: Pairwise sequence similarity data.

Pfam Protein domain																			
FLAT					HTD					HTD-CS					HBAYES-CS				
L.	Prec.	Rec.	F	Acc.	L.	Prec.	Rec.	F	Acc.	L.	Prec.	Rec.	F	Acc.	L.	Prec.	Rec.	F	Acc.
1	0.76	0.31	0.43	0.88	1	0.76	0.31	0.43	0.88	1	0.66	0.37	0.47	0.88	1	0.74	0.35	0.47	0.89
2	0.40	0.47	0.35	0.80	2	0.69	0.29	0.39	0.95	2	0.61	0.35	0.43	0.95	2	0.65	0.33	0.43	0.96
3	0.31	0.46	0.27	0.77	3	0.62	0.25	0.35	0.97	3	0.55	0.30	0.38	0.97	3	0.58	0.30	0.38	0.98
4	0.15	0.63	0.15	0.54	4	0.56	0.23	0.31	0.98	4	0.53	0.27	0.35	0.98	4	0.54	0.27	0.34	0.98
5	0.15	0.38	0.17	0.85	5	0.47	0.20	0.27	0.99	5	0.46	0.22	0.29	0.99	5	0.45	0.20	0.26	0.99

Sequence similarity																			
FLAT					HTD					HTD-CS					HBAYES-CS				
L.	Prec.	Rec.	F	Acc.	L.	Prec.	Rec.	F	Acc.	L.	Prec.	Rec.	F	Acc.	L.	Prec.	Rec.	F	Acc.
1	0.55	0.41	0.47	0.87	1	0.55	0.41	0.47	0.87	1	0.42	0.58	0.49	0.83	1	0.44	0.56	0.49	0.85
2	0.08	0.34	0.11	0.74	2	0.30	0.17	0.21	0.94	2	0.24	0.42	0.30	0.90	2	0.27	0.42	0.32	0.92
3	0.03	0.29	0.05	0.73	3	0.23	0.09	0.12	0.97	3	0.13	0.32	0.18	0.93	3	0.19	0.25	0.20	0.96
4	0.02	0.49	0.03	0.52	4	0.21	0.07	0.09	0.97	4	0.10	0.37	0.15	0.92	4	0.15	0.18	0.14	0.96
5	0.01	0.29	0.01	0.68	5	0.04	0.03	0.03	0.98	5	0.05	0.29	0.08	0.94	5	0.10	0.07	0.05	0.98

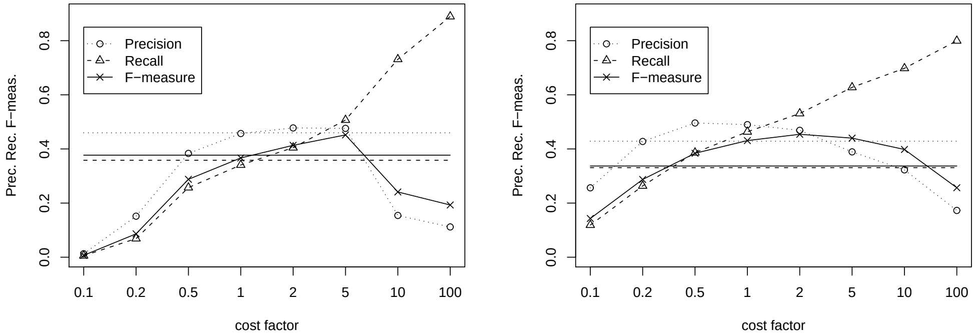


Fig. 2. Hierarchical precision, recall and F-measure as a function of the cost modulator factor in HBAYES-CS ensembles. Left: Protein domain data (Pfam-1). Right: Pairwise sequence similarity data (SP-sim). Horizontal lines refer to hierarchical precision, recall and F-score of HTD ensembles.

nodes of the FunCat hierarchy, level i, $2 \leq i \leq 5$, to nodes at depth i. We can observe that FLAT ensembles tend to have the highest recall, HTD the highest precision, while HBAYES-CS and HTD-CS tend to stay in the middle with respect to both the recall and precision, thus achieving the best F-measure at each level.

The precision/recall characteristics of HBAYES-CS ensemble can be tuned via a single global parameter, the cost factor $\alpha = c_i^- / c_i^+$ (Sect. 2). By setting $\alpha = 1$ we obtain the original version of the hierarchical Bayesian ensemble and by incrementing α we introduce progressively lower costs for positive predictions, thus encouraging the ensemble to make positive predictions. Indeed, by incrementing the cost factor, the recall of the ensemble tends to increase (Fig. 2). The behaviour of the precision is more complex: it tends to increase and then to decrease after achieving a maximum. Quite interestingly, the maximum of the hierarchical F-measure is achieved for values of α between 2 and 5 not only for the two data sets reported in Figure 2, but also for all the considered data sets (data not shown).

The improvement in performance of HBAYES-CS w.r.t. to HTD ensembles has a twofold explanation: the bottom-up approach permits the uncertainty in the decisions of the lower-level classifiers to be propagated across the network, and the cost sensitive setting allows to favor positive or negative decisions according to the value of cost factor. In all cases, a hierarchical approach (cost-sensitive or not) tends to achieve significantly higher precision than a flat approach, while cost-sensitive hierarchical methods are able to obtain a better recall at each level of the hierarchy, without a consistent loss in precision w.r.t. HTD methods — Table 3. We can note for all the hierarchical algorithms a degradation of both precision and recall (and as a consequence of the F-measure) by descending the levels of the trees (Table 3). This fact could be at least in part due to the lack of annotations at the lowest levels of the hierarchy, where we may have several genes with unannotated specific functions. Despite the fact that the overall performances of HBAYES-CS and HTD-CS are comparable, we can note that HBAYES-CS achieves a better precision (Tab. 3). This is of paramount importance in real applications, when we need to reduce the costs of the biological validation of new gene functions discovered through computational methods. Finally, it is worth noting that the accuracy is high at each level (at least with hierarchical ensemble methods), but these results are not significant, considering the large unbalance between positive and negative genes for each functional class.

4 Conclusions

The experimental results show that the prediction of gene functions needs a hierarchical approach, confirming previous recently published findings [5, 2]. Our proposed hierarchical methods, by exploiting the hierarchical relationships between classes, significantly improve on "flat" methods. Moreover, by introducing a cost-sensitive parameter, we are able to increase the hierarchical F-score with respect to the cost-insensitive version HTD. We observed that the precision/recall characteristics of HBAYES-CS can be tuned by modulating a single global parameter, the cost factor, according to the experimental needs. On the other hand, on our data sets the Bayesian ensemble HBAYES-CS did not exhibit a significant advantage over the simpler cost-sensitive top-down ensemble HTD-CS (see Fig. 1 and Tab. 2). We conjecture this might be due to the excessive noise in the annotations at lower levels of the hierarchy. It remains an open problem to devise ensemble methods whose hierarchical performance is consistently better than top-down approaches even on highly noisy data sets.

In our experiments we used only one type of data for each classification task, but it is easy to use state-of-the-art data integration methods to significantly improve the performance of HBAYES-CS. Indeed, for each node/class of the tree we may substitute the classifier trained on a specific type of biomolecular data with a classifier trained on concatenated vectors of different data [5], or trained on a (weighted) sum of kernels [20], or with an ensemble of learners each trained on a different type of data [21]. This is the subject of our planned future research.

Acknowledgments

We would like to thank the anonymous reviewers for their comments and suggestions. This work was partially supported by the PASCAL2 Network of Excellence (EC grant no. 216886).

References

1. The Gene Ontology Consortium: Gene ontology: tool for the unification of biology. Nature Genet. **25** (2000) 25–29
2. Obozinski, G., Lanckriet, G., Grant, C., M., J., Noble, W.: Consistent probabilistic output for protein function prediction. Genome Biology **9** (2008)

3. Sokolov, A., Ben-Hur, A.: A structured-outputs method for prediction of protein function. In: MLSB08, the Second International Workshop on Machine Learning in Systems Biology. (2008)
4. Astikainen, K., Holm, L., Pitkanen, E., Szedmak, S., Rousu, J.: Towards structured output prediction of enzyme function. BMC Proceedings **2** (2008)
5. Guan, Y., Myers, C., Hess, D., Barutcuoglu, Z., Caudy, A., Troyanskaya, O.: Predicting gene function in a hierarchical context with an ensemble of classifiers. Genome Biology **9** (2008)
6. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Guldener, U., Mannhaupt, G., Munsterkotter, M., Mewes, H.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Research **32** (2004) 5539–5545
7. Cesa-Bianchi, N., Gentile, C., Tironi, A., Zaniboni, L.: Incremental algorithms for hierarchical classification. In: Advances in Neural Information Processing Systems. Volume 17., MIT Press (2005) 233–240
8. Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Hierarchical classification: Combining Bayes with SVM. In: Proc. of the 23rd Int. Conf. on Machine Learning, ACM Press (2006) 177–184
9. Finn, R., Tate, J., Mistry, J., Coggill, P., Sammut, J., Hotz, H., Ceric, G., Forslund, K., Eddy, S., Sonnhammer, E., Bateman, A.: The Pfam protein families database. Nucleic Acids Research **36** (2008) D281–D288
10. Eddy, S.: Profile hidden markov models. Bioinformatics **14** (1998) 755–763
11. Pavlidis, P., Weston, J., Cai, J., Noble, W.: Learning gene functional classification from multiple data. J. Comput. Biol. **9** (2002) 401–411
12. Spellman, P., et al.: Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomices cerevisiae by microarray hybridization. Mol. Biol. Cell **9** (1998) 3273–3297
13. Gasch, P., et al.: Genomic expression programs in the response of yeast cells to environmental changes. Mol.Biol.Cell **11** (2000) 4241–4257
14. Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. Nucleic Acids Res. **34** (2006) D535–D539
15. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S., Fields, S., Bork, P.: Comparative assessment of large-scale data sets of protein-protein interactions. Nature **417** (2002) 399–403
16. Lanckriet, G., Gert, R.G., Deng, M., Cristianini, N., Jordan, M., Noble, W.: Kernel-based data fusion and its application to protein function prediction in yeast. In: Proceedings of the Pacific Symposium on Biocomputing. (2004) 300–311
17. Lin, H., Lin, C., Weng, R.: A note on Platt's probabilistic outputs for support vector machines. Machine Learning **68** (2007) 267–276
18. Verspoor, K., Cohn, J., Mnizewski, S., Joslyn, C.: A categorization approach to automated ontological function annotation. Protein Science **15** (2006) 1544–1549
19. Dietterich, T.: Approximate statistical test for comparing supervised classification learning algorithms. Neural Computation **10** (1998) 1895–1924
20. Lanckriet, G., De Bie, T., Cristianini, N., Jordan, M., Noble, W.: A statistical framework for genomic data fusion. Bioinformatics **20** (2004) 2626–2635
21. Re, M, Valentini, G,: Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction. JMLR: MLSB 09, 3rd International Workshop on Machine Learning in Systems Biology (2009)