Hierarchical Cost-Sensitive Algorithms for Genome-Wide Gene Function Prediction

Nicolò Cesa-Bianchi and **Giorgio Valentini** {CESA-BIANCHI,VALENTINI}@DSI.UNIMI.IT
DSI, Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico 39, 20135 Milano, Italy

Editor: Saso Dzeroski, Pierre Geurts, and Juho Rousu

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 seven biomolecular data sets, reveal a significant advantage of our techniques over "flat" and cost-insensitive hierarchical ensembles.

Keywords: Hierarchical classification, Gene function prediction, Bayesian ensembles, Cost-sensitive classification, FunCat taxonomy.

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 automatic prediction methods 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 of functional genomics.

Gene function prediction is a classification problem with distinctive features, which include: (a) a large number of classes with multiple functional annotations for each gene — i.e., a multiclass multilabel classification problem; (b) hierarchical relationships between classes governed by the "true path rule" (The Gene Ontology Consortium, 2000); (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 (Obozinski et al., 2008). 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 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 lines of research 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 (Sokolov and Ben-Hur, 2008; Astikainen et al., 2008), 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 (Obozinski et al., 2008; Guan et al., 2008; Jiang et al., 2008). Our work follows this second line of research. 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 (Ruepp et al., 2004).

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 (Cesa-Bianchi et al., 2005, 2006) 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. In principle any algorithm whose classifications are obtained by thresholding a real prediction $\widehat{p}$, e.g., $\widehat{y} = \text{SGN}(\widehat{p})$, can be used as base learner. In this work we use support vector machines with Platt calibration. The real-valued outputs $\widehat{p}_i(\boldsymbol{x})$ of the calibrated classifier for node i on input $\boldsymbol{x}$ are viewed as estimates of the probabilities $p_i(\boldsymbol{x}) = \mathbb{P}\big(V_i = 1 \mid V_{\text{par}(i)} = 1, \boldsymbol{x}\big)$, where $\boldsymbol{V} = (V_1, \ldots, V_N) \in \{0,1\}^N$ is the vector random variable modeling the multilabel of a gene $\boldsymbol{x}$ and $\text{par}(i)$ is the unique parent of node i in T. The distribution of the random boolean vector $\boldsymbol{V}$ is assumed to be

$$\mathbb{P}\big(\boldsymbol{V} = \boldsymbol{v}\big) = \prod_{i=1}^{N} \mathbb{P}\big(V_i = v_i \mid V_{\text{par}(i)} = 1, \boldsymbol{x}\big) \qquad \text{for all } \boldsymbol{v} \in \{0,1\}^N$$

where, in order to enforce that only multilabels $\boldsymbol{V}$ that respect T have nonzero probability, we impose that $\mathbb{P}\left(V_i = 1 \mid V_{\mathrm{par}(i)} = 0,\, \boldsymbol{x}\right) = 0$ for all nodes $i = 1, \ldots, N$ and all $\boldsymbol{x}$. This implies that 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,

$$\widehat{\boldsymbol{y}} = \operatorname*{argmin}_{\boldsymbol{y} \in \{0,1\}^n} \mathbb{E}\big[\ell_H(\boldsymbol{y}, \boldsymbol{V}) \mid \boldsymbol{x}\,\big] \tag{1}$$

where the expectation is w.r.t. the distribution of $\boldsymbol{V}$. Here $\ell_H(\boldsymbol{y}, \boldsymbol{V})$ denotes the H-loss (Cesa-Bianchi et al., 2005, 2006), 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* $c_1, \ldots, c_N > 0$, $\ell_H(\widehat{\boldsymbol{y}}, \boldsymbol{v})$ is computed as follows: all paths in the taxonomy T from the root 0 down to each leaf are examined and, whenever a node $i \in \{1, \ldots, N\}$ 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.

Let $c_i^- = c_i^+ = c_i/2$ be the costs respectively associated to a false negative (FN) and a false positive (FP) mistake. Let $\{\, A \,\}$ be the indicator function of event A. Given $\boldsymbol{x}$ and the probabilities $p_i = p_i(\boldsymbol{x})$ for $i = 1, \ldots, N$, the HBAYES prediction rule can be formulated as follows.

HBAYES prediction rule: Initially, set the labels of each node i to

$$\widehat{y}_i = \operatorname*{argmin}_{y \in \{0,1\}} \left(c_i^- p_i (1 - y) + c_i^+ (1 - p_i) y + p_i \{y = 1\} \sum_{j \in \mathrm{child}(i)} H_j(\widehat{\boldsymbol{y}}) \right) \tag{2}$$

where

$$H_j(\widehat{\boldsymbol{y}}) = c_j^- p_j (1 - \widehat{y}_j) + c_j^+ (1 - p_j) \widehat{y}_j + p_j \{\widehat{y}_j = 1\} \sum_{k \in \mathrm{child}(j)} H_k(\widehat{\boldsymbol{y}}) \tag{3}$$

is recursively defined over the nodes j in the subtree rooted at i with each $\widehat{y}_j$ set according to (2). Then, if $\widehat{y}_i$ is set to zero, set all nodes in the subtree rooted at i to zero as well.

As shown in (Cesa-Bianchi et al., 2006), $\widehat{\boldsymbol{y}}$ can be computed for a given $\boldsymbol{x}$ via a simple bottom-up message-passing procedure whose only parameters are the probabilities p_i. Unlike standard top-down hierarchical methods —see the description of HTD at the end of this section, each $\widehat{y}_i$ also depends on the classification of its child nodes. In particular, if all child nodes k of i have p_k close to a half, then the Bayes-optimal label of i tends to be 0 irrespective of the value of p_i. Vice versa, if i's children all have p_k close to either 0 or 1, then the Bayes-optimal label of i is based on p_i only, ignoring the children —see also (6).

The following theorem by Cesa-Bianchi et al. (2005), whose proof we include here for completeness, shows that assignments $\widehat{y}$ given by (1) and the HBAYES prediction rule are indeed equivalent.

Theorem 1 *For any tree T and all unit-norm $\boldsymbol{x} \in \mathbb{R}^d$, the multilabel generated according to the* HBAYES *prediction rule is the Bayes-optimal classification of $\boldsymbol{x}$ for the H-loss.*

Proof Let $\widehat{\boldsymbol{y}}$ be the multilabel assigned by HBAYES and $\boldsymbol{y}^*$ be any multilabel minimizing the expected H-loss. Omitting the indexing on $\boldsymbol{x}$, we can write

$$\mathbb{E}\,\ell_H(\widehat{\boldsymbol{y}}, \boldsymbol{V}) = \sum_{i=1}^{N} \left(c_i^- p_i(1-\widehat{y}_i) + c_i^+(1-p_i)\widehat{y}_i\right) \prod_{j \in \mathrm{anc}(i)} p_j \left\{\widehat{y}_j = 1\right\}$$

where $\mathrm{anc}(i)$ is the set of nodes $j > 0$ that are ancestors of i. Note that we may decompose the expected H-loss as

$$\mathbb{E}\,\ell_H(\widehat{\boldsymbol{y}}, \boldsymbol{V}) = \sum_{i \in \mathrm{root}(T)} H'_i(\widehat{\boldsymbol{y}})$$

where $\mathrm{root}(T)$ are the children of the dummy root node 0, and $H'_i(\widehat{\boldsymbol{y}})$ is recursively defined as

$$H'_i(\widehat{\boldsymbol{y}}) = \left(c_i^- p_i(1-\widehat{y}_i) + c_i^+(1-p_i)\widehat{y}_i\right) \prod_{j \in \mathrm{anc}(i)} p_j \left\{\widehat{y}_j = 1\right\} + \sum_{k \in \mathrm{child}(i)} H'_k(\widehat{\boldsymbol{y}}) \ . \tag{4}$$

Pick a node i. If i is a leaf, then the sum in the right-hand side of (4) disappears and $y_i^* = \{p_i \geq 1/2\}$ —recall $c_i^- = c_i^+$. As this is also the minimizer of $H_i(\widehat{\boldsymbol{y}}) = c_i^- p_i(1-\widehat{y}_i) + c_i^+(1-p_i)\widehat{y}_i$, we get that $\widehat{y}_i = y_i^*$.

Now let i be an internal node and inductively assume $\widehat{y}_j = \widehat{y}_j^*$ for all $j \in \mathrm{subtree}(i)$. By expanding a term $H'_k(\widehat{\boldsymbol{y}})$ of the sum in the right-hand side of (4) we get

$$H'_k(\widehat{\boldsymbol{y}}) = \left(c_k^- p_k(1-\widehat{y}_k) + c_k^+(1-p_k)\widehat{y}_k\right) p_i \left\{\widehat{y}_i = 1\right\} \prod_{j \in \mathrm{anc}(i)} p_j \left\{\widehat{y}_j = 1\right\} + \sum_{m \in \mathrm{child}(k)} H'_m(\widehat{\boldsymbol{y}}) \ .$$

Note that the factor $\prod_{j \in \mathrm{anc}(i)} p_j \left\{\widehat{y}_j = 1\right\}$ occurs in both terms in the right-hand side of (4). Hence y_i^* does not depend on this factor, thus instead of (4) we can equivalently minimize

$$H_i(\widehat{\boldsymbol{y}}) = c_i \left(p_i(1-\widehat{y}_i) + (1-p_i)\widehat{y}_i\right) + p_i\{\widehat{y}_i = 1\} \sum_{k \in \mathrm{child}(i)} H_k(\widehat{\boldsymbol{y}}) \tag{5}$$

where $H_k(\widehat{\boldsymbol{y}})$ is defined as in (3). Now observe that $\widehat{y}_i$ minimizing (5) is equivalent to the assignment produced by HBAYES.

To conclude the proof note that setting $y_j^* = 0$ for all nodes $j \in \mathrm{subtree}(i)$ whenever $y_i^* = 0$ minimizes the H-loss, and this is exactly what the HBAYES prediction rule does. ■

We now introduce 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 parametrize the relative costs of FP and FN by introducing a factor $\alpha \geq 0$ such that $c_i^- = \alpha c_i^+$ while keeping $c_i^+ + c_i^- = 2c_i$. Then (2) can be rewritten as

$$\widehat{y}_i = 1 \Longleftrightarrow p_i \left(2c_i - \sum_{j \in \mathrm{child(i)}} H_j\right) \geq \frac{2c_i}{1+\alpha} \ . \tag{6}$$

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

Given a set of trained base learners providing estimates $\widehat{p}_1, \ldots, \widehat{p}_N$, we compare the quality of the multilabels computed by HBAYES-CS with that of HTD-CS. This is a cost-sensitive version of the basic top-down hierarchical ensemble method HTD whose predictions are computed in a top-down fashion (i.e., assigning $\widehat{y}_i$ before the label of any j is the subtree rooted at i) using the rule $\widehat{y}_i = \left\{\widehat{p}_i(\boldsymbol{x}) \geq \frac{1}{2}\right\} \times \{\widehat{y}_{\mathrm{par}(i)} = 1\}$ for $i = 1, \ldots, N$ (we assume that the guessed label $\widehat{y}_0$ of the root of T is always 1). The variant HTD-CS introduces a single cost sensitive parameter $\tau > 0$ which replaces the threshold $\frac{1}{2}$. The resulting rule for HTD-CS is then $\widehat{y}_i = \{\widehat{p}_i(\boldsymbol{x}) \geq \tau\} \times \{\widehat{y}_{\mathrm{par}(i)} = 1\}$.

Note that both methods HBAYES-CS and HTD-CS 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 (Ruepp et al., 2004). The FunCat provides an universal set of gene functional classes available for all organisms: it consists of 28 main functional categories (or branches) that cover general fields like cellular transport, metabolism and cellular communication/signal transduction. These main functional classes are divided into a set of subclasses with up to six levels of increasing specificity, according to a tree-like structure that accounts for different functional characteristics of genes and gene products. In the FunCat taxonomy genes may belong at the same time to multiple functional classes, since several classes are subclasses of more general ones, and because a gene may participate to different biological processes and may perform different biological functions.

3.1 Experimental set-up

In our experiments we used 7 biomolecular data sets, whose characteristics are summarized in Table 1.

Table 1: Data sets

Data set	Description	# of genes	# of features	# 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 (Finn et al., 2008). 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 (Eddy, 1998). The features of the phylogenetic data (Phylo) are the negative loga-

rithm of the lowest E-value reported by BLAST version 2.0 in a search against a complete genome in 24 organisms (Pavlidis et al., 2002). The "Expr" data set merges the experiments of (Spellman et al., 1998) about gene expression measures relative to 77 conditions with the transcriptional responses of yeast to environmental stress (173 conditions) by (Gasch et al., 2000). 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 (Stark et al., 2006). 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 (von Mering et al., 2002). 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 (Lanckriet et al., 2004b).

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).

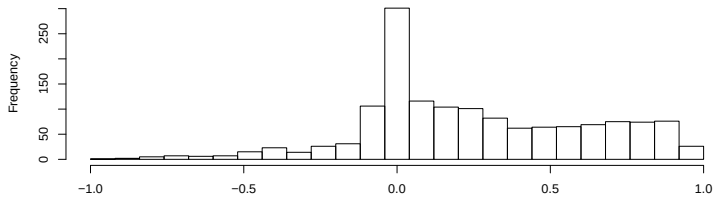
We used gaussian SVMs with probabilistic output (Lin et al., 2007) as base learners. Given a set $\widehat{p}_1, \ldots, \widehat{p}_N$ of trained estimates, we compared on these estimates the results of HTD-CS and HBAYES-CS ensembles with HTD (the cost-insensitive version of HTD-CS, obtained by setting $\tau = 1/2$) and FLAT (each classifier outputs its prediction disregarding the taxonomy). The decision threshold τ for HTD-CS and the cost factor α for HBAYES-CS have been set by internal cross-validation of the F-measure with training data. We compared the different ensemble methods using external 5-fold cross-validation (thus without using test set data to tune the hyper-parameters).

3.2 Per-class F-score results

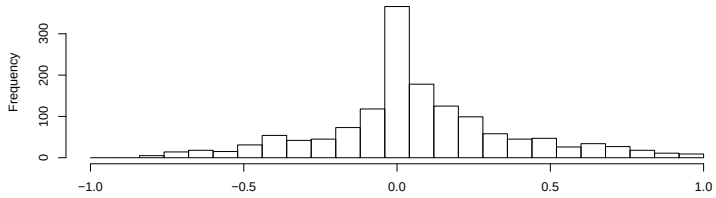
For the first set of experiments we used the classical F-score to aggregate precision and recall for each class of the hierarchy. Figure 1 shows the distribution, across all the classes of the taxonomy and the data sets, of the normalized differences $\frac{F_{\text{Bayes}} - F_{\text{ens}}}{\max(F_{\text{Bayes}}, F_{\text{ens}})}$ 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.

3.3 Hierarchical F-score results

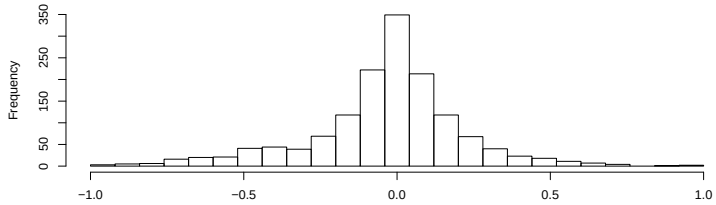
In order to better capture the hierarchical and sparse nature of the gene function prediction problem we also applied the *hierarchical F-measure*, expressing in a synthetic way the



(a)



(b)



(c)

Figure 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.

effectiveness of the structured hierarchical prediction (Verspoor et al., 2006). Given a general taxonomy T representing the graph of the functional classes, for a given gene or gene product x consider the graph $P(x) \subset T$ of the predicted classes and the graph $C(x)$ of the correct classes associated to x, and let be $l(P)$ the set of the leaves (nodes without children) of the graph P. For a leaf $f \in P(x)$ and $c \in C(x)$, let be $\uparrow f$ and $\uparrow c$ the set of their ancestors that belong, respectively, to $P(x)$ and $C(x)$. It is easy to see that the Hierarchical Precision (HP), Hierarchical Recall (HR) and Hierarchical F-score (HF), originally introduced for the DAGs of the GO (Verspoor et al., 2006), can be adapted to the tree-structure of the FunCat taxonomy in the following way:

$$HP = \frac{1}{|l(P(x))|} \sum_{f \in l(P(x))} \frac{|C(x) \cap \uparrow f|}{|\uparrow f|} \qquad HR = \frac{1}{|l(C(x))|} \sum_{c \in l(C(x))} \frac{|\uparrow c \cap P(x)|}{|\uparrow c|}$$

$$HF = \frac{2 \cdot HP \cdot HR}{HP + HR} \tag{7}$$

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: Upper table: Hierarchical F-measure comparison between HTD, HTD-CS, and HBAYES-CS ensembles. Lower table: 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

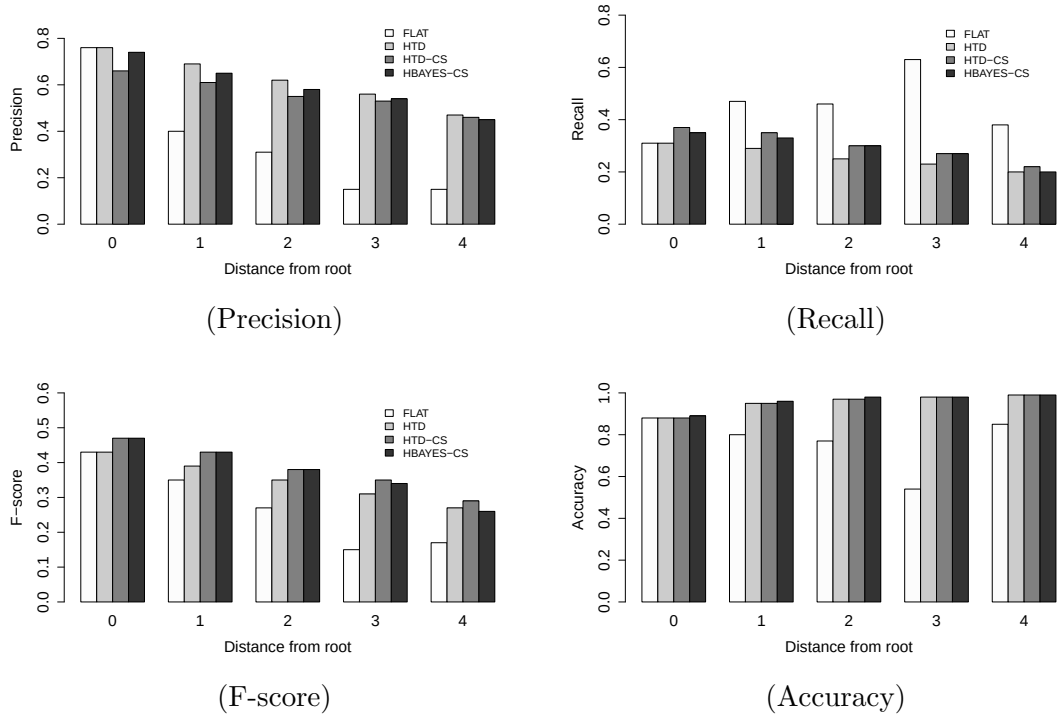
win-tie-loss		
Methods	HTD-CS	HTD
HBAYES-CS	2-4-1	6-1-0
HTD-CS	-	7-0-0

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 (Dietterich, 1998) 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.

3.4 Performance on the most specific classes

At first, in order to characterize the behaviour of the different ensemble methods with respect to the overall structure of the FunCat taxonomy, we performed an analysis of the performance of the algorithms at different levels of the hierarchy. Figure 2 shows the per-level F-measure results with Pfam-1 protein domain data. We can observe that FLAT ensembles tend to have the highest recall at each level, 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 hierarchical methods show a precision and recall (and consequently F-score) that decrease with the distance from the root. Flat ensembles, even if achieve relatively good recall results, show a precision too low to be useful in practical applications for genome-wide gene function prediction. Note that the accuracy is very high at all levels of the hierarchy (at least for the hierarchical methods), but this is not significant considering the high unbalance between positive and negative examples.

As a second step, to understand whether the proposed hierarchical methods are able to correctly classify the most specific classes of the hierarchy, that is the terms that better



(Precision) (Recall)

(F-score) (Accuracy)

Figure 2: Averages of various performance scores at different levels of the FunCat taxonomy on Pfam-1 data of FLAT, HTD, HTD-CS and HBAYES-CS ensembles. 0 represents nodes with distance 0 from the root, i.e. root nodes; 1, 2, 3 and 4 represent nodes at a given distance from the root.

characterize the functions of a given gene/gene product, we performed an analysis of the performance with respect to the F-score, precision and recall achieved at the leaves and at nodes at a given distance from the leaves. More precisely, we analyzed the average precision, recall, and F-score of nodes at a given distance from any leaf. For distance of a node from any leaf we mean the length of the longest path from the node to a leaf belonging to the subtree rooted at the node itself. Figure 4 shows the results obtained with the *Pfam-1* data set. The Pfam-1 data set includes 211 classes (Figure 3), with 124 leaf nodes/classes, 45 nodes with distance 1 from a leaf, and 25, 12 and 5 nodes with distance respectively equal to 2, 3 and 4 from any leaf node. The value 0 in abscissa corresponds to the nodes at distance 0, that is the leaves of the FunCat tree. With hierarchical methods, precision, recall and F-score increase with the distance from leaves. HTD ensembles show the largest precision, and FLAT the largest recall, while both HTD-CS and HBAYES-CS are in between and achieve the largest F-score at any distance from the leaves, with a slightly better precision for HBAYES-CS and a slightly better recall for HTD-CS. These results show that in any case both HTD-CS and HBAYES-CS improve the prediction performance on the most specific classes with respect to both HTD and FLAT ensembles. No significant differences in performances can be observed

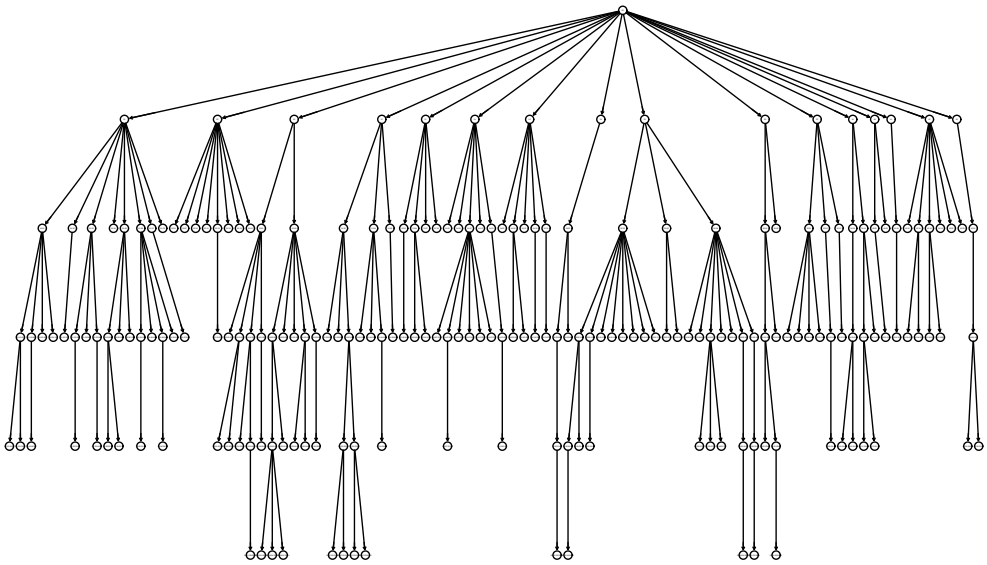


Figure 3: The FunCat hierarchy for the protein domain data (Pfam-1 data set).

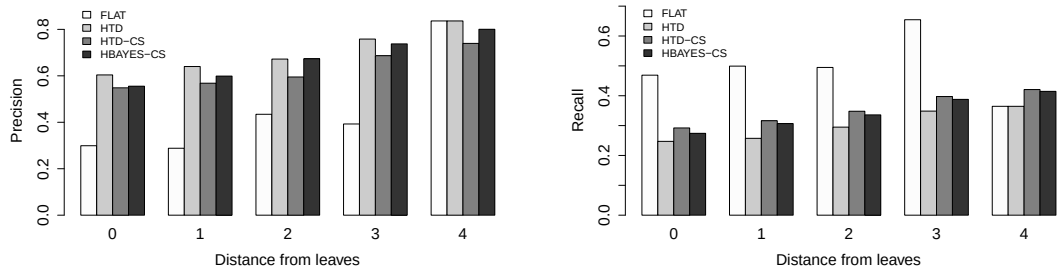
between HTD-CS and HBAYES-CS. A similar trend can be observed also with the other data sets (data not shown).

3.5 Tuning precision and recall with a single global parameter

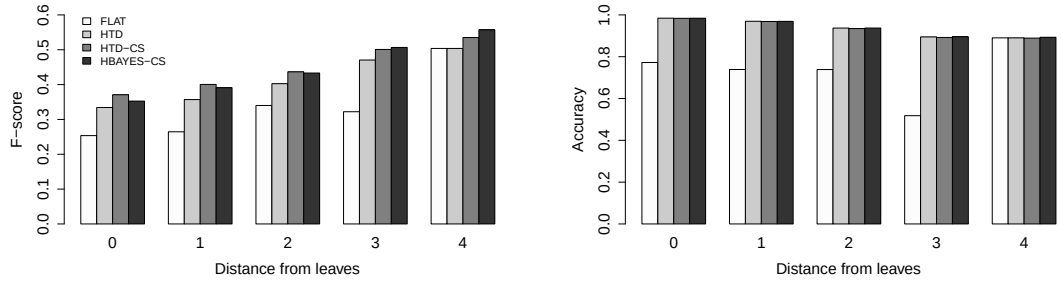
The precision/recall characteristics of HBAYES-CS ensemble can be tuned via a single global parameter, the cost factor $\alpha = c_i^-/c_i^+$ (Sect. 2). By setting $\alpha = 1$ we obtain the original version of the hierarchical Bayesian ensemble and by incrementing α we introduce progressively lower 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 (Figure 5). 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 5, but also for all the considered data sets (data not shown).

3.6 Automatic setting of the parameter α at each node

In Section 3.5 we showed that by appropriately tuning the global parameter α we can obtain HBAYES-CS ensembles with different precision/recall characteristics. In principle, we could appropriately choose the α parameter at each node, but this leads to a complex optimization problem. Considering that α represents a factor to balance the misclassification cost between positive and negative examples, we could simply choose a cost factor α_i for each node i to explicitly take into account the unbalance between the number of positive n_i^+ and negative



(Precision) (Recall)



(F-measure) (Accuracy)

Figure 4: Averages of various performance scores for nodes at different distances from the closest leaf node. A value of 0 represents leaf nodes.

n_i^- examples:

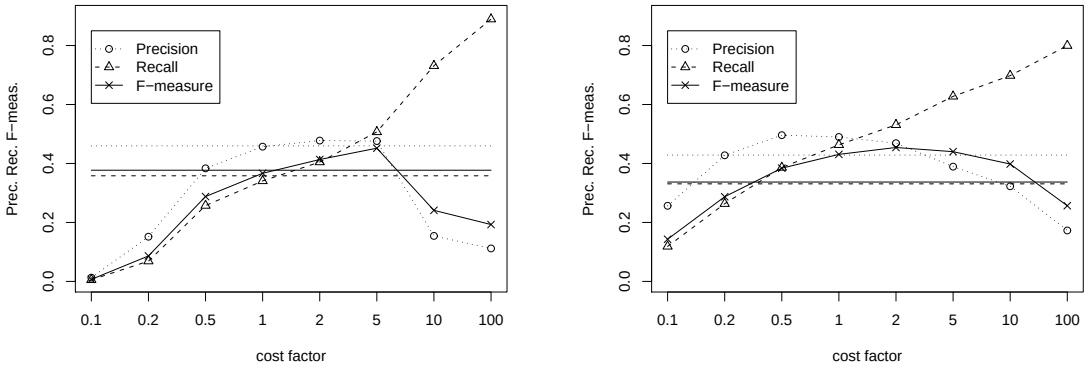
$$\alpha_i = \frac{n_i^-}{n_i^+} \Rightarrow c_i^+ = \frac{2}{(\frac{n_i^-}{n_i^+}) + 1} c_i = \frac{2n_i^+}{n_i^- + n_i^+} c_i \tag{8}$$

The decision rule (6) at each node becomes:

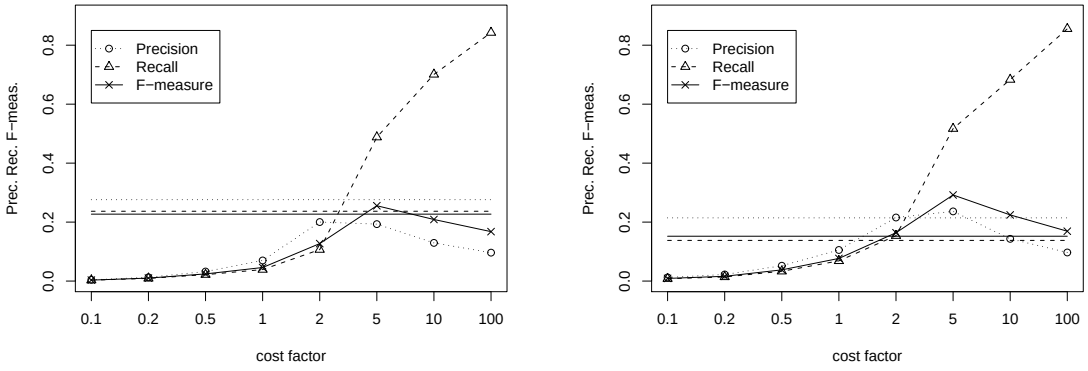
$$\widehat{y}_i = 1 \Longleftrightarrow p_i \left(2c_i - \sum_{j \in \text{child(i)}} H_j \right) \geq \frac{2c_i}{1 + \alpha_i} = \frac{2c_i n_i^+}{n_i^- + n_i^+} \ . \tag{9}$$

The number of positive n_i^+ and negative n_i^- examples can be estimated from the training data.

We compared this simple heuristic with the tuning of α as a single global parameter by internal cross-validation, and with the heuristic variant $\alpha_i = \max\{n_i^-/n_i^+, 3\}$. This second heuristic aims to improve the detection of positive examples also for classes where we have an unbalance in favour of positive examples in the training set. Indeed, the constraint that only examples $(\boldsymbol{x}, \boldsymbol{v})$ such that $v_{\text{par}(i)} = 1$ are included in the training set for node i (Section 2) may induce for some classes a number of positive examples comparable or also



(Pfam-1) (SP-sim)



(Expr) (PPI-BG)

Figure 5: Hierarchical precision, recall and F-measure as a function of the cost modulator factor in HBAYES-CS ensembles for the following data sets: protein domain data (Pfam-1); pairwise sequence similarity data (SP-sim); gene expression data (Expr); protein-protein interaction data (PPI-BG). Horizontal lines refer to hierarchical precision, recall and F-score of HTD ensembles.

larger than that of negatives, thus resulting in too low values for the corresponding values of α_i.

Table 3 shows the results of the comparison of these variants of HBAYES-CS. *CV* stands for HBAYES-CS whose global α values has been selected by internal cross-validation, *auto* (automatic setting of α) for the heuristic per-node setting of $\alpha_i = n_i^-/n_i^+$, and *auto*3 (automatic setting of α constrained to be larger or equal to 3 for each node). The results show that the automatic tuning of α_i values constrained to be larger or equal to 3 (*auto*3) leads to results comparable with those obtained by tuning α through cross-validation (*CV*). The application of the Wilcoxon signed-ranks test (Demsar, 2006) (p-value=0.53) confirms

Table 3: Compared performances (Hierarchical F-score) of HBAYES-CS ensembles with parameter α set by cross-validation (CV), and by heuristic per-node settings (*auto* and *auto*3).

HBAYES-CS	**Data sets**							
methods	Pfam-1	Pfam-2	Phylo	Expr	PPI-BG	PPI-VM	SP-sim	Average
CV	0.4518	0.2030	0.2682	0.2555	0.2920	0.4329	0.4542	0.3368
auto	0.4070	0.0163	0.1966	0.1566	0.2142	0.3998	0.4234	0.2591
*auto*3	0.4410	0.2067	0.2590	0.2410	0.3058	0.4442	0.4583	0.3365

that there is no significant difference between *auto*3 and CV. The heuristic of setting $\alpha_i = n_i^- / n_i^+$ for each node (*auto*) is reasonable with some data sets (e.g., with *SP-sim*), but fails with other data sets, since in some cases it cannot detect positive examples for very unbalanced classes when less informative data are used (see, e.g., results of *auto* with *Pfam-2*, Table 3).

3.7 Discussion

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 —Fig 2. For all the hierarchical algorithms we note a degradation of both precision and recall (and as a consequence of the F-measure) by descending the levels of the trees (Figure 2). 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.

HBAYES-CS shows better performances than HTD also w.r.t. the most specific classes, that better characterize the functions of the genes (Figure 4). 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. 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.

Another advantage of HBAYES-CS vs. HTD-CS consists in the automatic tuning of the factor α at each node avoiding internal cross-validation (Sect. 3.6). For HTD-CS, instead, the setting of the threshold τ requires an expensive internal cross-validation.

From Table 2 we observe that with certain data sets HTD-CS outperforms HBAYES-CS (Phylo data set), while the opposite is true with other data sets (e.g. Pfam-1 and SP-sim). We do not have a clear explanation of this fact, and this could be the subject of future investigations.

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 due to 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 (Guan et al., 2008; Obozinski et al., 2008). 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 Figure 1 and Table 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 our methods. 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 (Guan et al., 2008), or trained on a (weighted) sum of kernels (Lanckriet et al., 2004a), or with an ensemble of learners each trained on a different type of data (Re and Valentini, 2010). This is the subject of our planned future research.

Acknowledgments

The authors gratefully acknowledge partial support by the PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the authors' views.

References

K. Astikainen, L. Holm, E. Pitkanen, S. Szedmak, and J. Rousu. Towards structured output prediction of enzyme function. *BMC Proceedings*, 2(Suppl 4:S2), 2008.

N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni. Incremental algorithms for hierarchical classification. In *Advances in Neural Information Processing Systems*, volume 17, pages 233–240. MIT Press, 2005.

N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Hierarchical classification: Combining Bayes with SVM. In *Proc. of the 23rd Int. Conf. on Machine Learning*, pages 177–184. ACM Press, 2006.

J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

T.G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, 1998.

SR Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–763, 1998.

R.D. Finn, J. Tate, J. Mistry, P.C. Coggill, J.S. Sammut, H.R. Hotz, G. Ceric, K. Forslund, S.R. Eddy, E.L. Sonnhammer, and A. Bateman. The Pfam protein families database. *Nucleic Acids Research*, 36:D281–D288, 2008.

P. Gasch et al. Genomic expression programs in the response of yeast cells to environmental changes. *Mol.Biol.Cell*, 11:4241–4257, 2000.

Y Guan, C.L. Myers, D.C. Hess, Z. Barutcuoglu, A. Caudy, and O.G. Troyanskaya. Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, 9(S2), 2008.

X. Jiang, N. Nariai, M. Steffen, S. Kasif, and E. Kolaczyk. Integration of relational and hierarchical network information for protein function prediction. *BMC Bioinformatics*, 9 (350), 2008.

G.R. Lanckriet, T. De Bie, N. Cristianini, M. Jordan, and W.S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626–2635, 2004a.

G.R. Lanckriet, R. G. Gert, M. Deng, N. Cristianini, M. Jordan, and W.S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 300–311, 2004b.

H.T. Lin, C.J. Lin, and R.C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68:267–276, 2007.

G. Obozinski, G. Lanckriet, C. Grant, Jordan. M., and W.S. Noble. Consistent probabilistic output for protein function prediction. *Genome Biology*, 9(S6), 2008.

P. Pavlidis, J. Weston, J. Cai, and W.S. Noble. Learning gene functional classification from multiple data. *J. Comput. Biol.*, 9:401–411, 2002.

M. Re and G. Valentini. Simple ensemble methods are competitive with state-of-the-art data integration methods for gene function prediction. *Journal of Machine Learning Research, W&C Proceedings*, (in this issue), 2010.

A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and H.W. Mewes. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.

A. Sokolov and A. Ben-Hur. A structured-outputs method for prediction of protein function. In *MLSB08, the Second International Workshop on Machine Learning in Systems Biology*, 2008.

P. Spellman et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomices cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297, 1998.

C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.*, 34:D535–D539, 2006.

The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.

K. Verspoor, J. Cohn, S. Mnizewski, and C. Joslyn. A categorization approach to automated ontological function annotation. *Protein Science*, 15:1544–1549, 2006.

C. von Mering, R. Krause, B. Snel, M. Cornell, S. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417: 399–403, 2002.