Weighted True Path Rule: a multilabel hierarchical algorithm for gene function prediction

Giorgio Valentini and Matteo Re

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

Abstract. The genome-wide hierarchical classification of gene functions, using biomolecular data from high-throughput biotechnologies, is one of the central topics in bioinformatics and functional genomics. In this paper we present a multilabel hierarchical algorithm inspired by the "true path rule" that governs both the Gene Ontology and the Functional Catalogue (FunCat). In particular we propose an enhanced version of the *True Path Rule* (TPR) algorithm, by which we can control the flow of information between the classifiers of the hierarchical ensemble, thus allowing to tune the precision/recall characteristics of the overall hierarchical classification system. Results with the model organism *S. cerevisiae* show that the proposed method significantly improves on the basic version of the TPR algorithm, as well as on the *Hierarchical Top-down* and *Flat* ensembles.

1 Introduction

Gene function prediction is a multiclass, multilabel classification problem characterized by hundreds or thousands of functional classes structured according to a predefined hierarchy (a directed acyclic graph for the Gene Ontology [1] or a tree forest for FunCat [2]). Functional classes are usually unbalanced (with positive examples usually less than negatives), with labels that can be uncertain and in many cases unknown or only partially known.

From a general standpoint several approaches have been proposed for multilabel classification, with applications ranging from protein function classification, to music categorization and semantic scene classification [3].

Different approaches to the hierarchical multilabel classification of gene function have been proposed [4, 5], but schematically we can individuate two main research lines: a) structured-output methods, based on the joint kernelization of both input variables and output labels using perceptron-like learning algorithms [6] or maximum-margin algorithms [7]; b) ensemble methods by which different classifiers are trained to learn each class, and then combined to take into account the hierarchical relationships between functional classes [8, 9, 10].

Along this second line of research, we propose a multilabel ensemble algorithm, specialized for tree-structured taxonomies, to predict the functional classes of genes.

Our proposed approach is directly inspired by the *true path rule* that governs the annotations of both GO and FunCat taxonomies [1]:

> "An annotation for a class in the hierarchy is automatically transferred to its ancestors, while genes unannotated for a class cannot be annotated for its descendants".

According to this rule the proposed ensemble method is characterized by a two-way asymmetric flow of information that traverses the graph-structured ensemble: positive predictions for a node influence in a recursive way its ancestors, while negative predictions influence its offsprings. The resulting ensemble embeds the functional relationships between functional classes that characterize the hierarchical taxonomy.

The proposed method predicts the annotations of genes at the level of the entire taxonomy or considering specific subsets of the hierarchical functional classes, and provides probabilistic and structured predictions of gene annotations. Moreover, by tuning a single global parameter, it allows to regulate the trade-off between precision and recall that characterizes gene function prediction problems. We apply the *True Path Rule (TPR)* hierarchical ensemble methods to the prediction of gene functions in yeast, using probabilistic SVMs as base learners [11], but the algorithm is general enough to be used with any probabilistic base learner and with other model organisms. Considering that data integration is crucial to improve prediction performances [12], TPR ensembles can be easily integrated with state-of-the-art biomolecular data integration methods [13], such as vector-space integration [14], kernel fusion [15] or ensembles of learning machines [16], without any modification of the algorithmic scheme.

This paper is organized as follows: in Sect. 2 the ensemble method inspired by the *true path rule* is presented. Sect. 3 summarizes the experimental setup, while Sect. 4 show genome-wide gene function prediction results obtained with the model organism *S. cerevisiae* using the proposed method compared with hierarchical top-down and "flat" ensemble approaches. The conclusions and future developments end the paper.

2 Methods

2.1 Basic Definitions

Genome-wide gene function prediction can be modeled as a hierarchical, multiclass and multilabel classification problem. Indeed a gene/gene product x can be assigned to one or more functional classes of the set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$. The assignments can be coded through a vector of multilabels $\mathbf{y} =< y_1, y_2, \ldots, y_m > \in \{0,1\}^m$, by which if x belongs to class ω_j, then $y_j = 1$, otherwise $y_j = 0$.

In both the *Gene Ontology (GO)* and *FunCat* taxonomies the functional classes are structured according to a hierarchy and can be represented by a

directed graph, where nodes correspond to classes, and arcs to relationships between classes. Hence the node corresponding to the class ω_i can be simply denoted by i. We represent the set of children nodes of i by child(i), and the set of its parents by par(i). Moreover $y_{child(i)}$ denotes the labels of the children classes of node i and analogously $y_{par(i)}$ denotes the labels of the parent classes of i. Note that in FunCat only one parent is permitted, since the overall hierarchy is a tree forest, while in the GO, more parents are allowed, because the relationships are structured according to a directed acyclic graph. A classifier $D : X \rightarrow \{0,1\}^m$ computes the multilabel associated to each example $x \in X$, and $d_i(x) \in \{0,1\}$ is the label predicted by the classifier for class ω_i. For the sake of simplicity if there is no ambiguity we represent $d_i(x)$ simply by d_i.

2.2 An algorithm inspired by the "True Path Rule"

In both FunCat and GO ontologies, genes annotated to a specific functional class automatically belong to all its ancestors. Moreover, in FunCat, if a gene is not annotated to a given class, none of its offsprings can be annotated [1].

These basic rules constitute the so called "True Path Rule" that govern both GO and FunCat. Fig. 1 illustrates an example of the application of the true path rule.

For a given example x, considering the parents of a given node i, a classifier that respects the true path rule needs to obey the following rules:

$$\begin{cases} d_i = 1 \Rightarrow d_{par(i)} = 1 \\ d_i = 0 \nRightarrow d_{par(i)} = 0 \end{cases} \tag{1}$$

On the other hand, considering the children of a given node i, a classifier that respects the true path rule needs to obey the following rules:

$$\begin{cases} d_i = 1 \nRightarrow d_{child(i)} = 1 \\ d_i = 0 \Rightarrow d_{child(i)} = 0 \end{cases} \tag{2}$$

The proposed hierarchical ensemble algorithm puts together the predictions made at each node by local "base" classifiers to realize an ensemble that obeys the "true path rule".

The basic ideas behind the *true path rule ensemble algorithm* can be summarized as follows:

1. Training of the base learners: for each node of the hierarchy a suitable learning algorithm (e.g. a multi-layer perceptron or a support vector machine) provides a classifier for the associated functional class
2. In the evaluation phase the trained classifiers associated to each class/node of the graph provide a local decision about the assignment of a given example to a given node.

[1] For the GO, this rule is slightly more complicated, because the GO is structured according to a directed acyclic graph, and even if a gene is not annotated to a class i, it can be annotated to a child of i, say j, if it is annotated to at least one of its parents $k \neq i$.

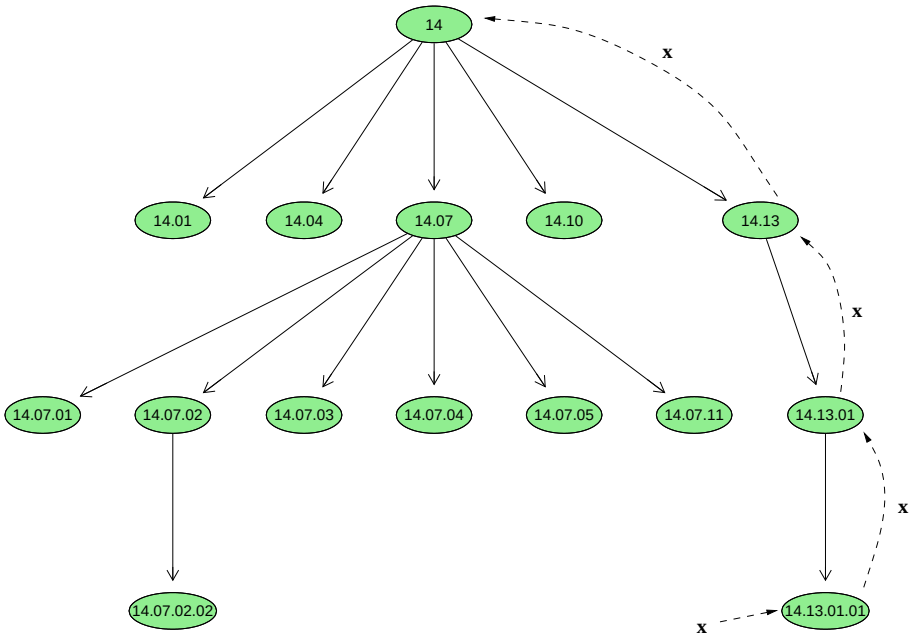

Fig. 1: FunCat tree rooted at class 14 (Protein fate): if example x belongs to class 14.03.01.01 then it belongs also to class 14.03.01, 14.03 and 14. On the contrary, if an example x does not belong to class 14.07 it cannot belong to any of its children (e.g. 14.07.01, 14.07.02, 14.07.03, 14.07.04, 14.07.05, 14.07.11).

3. Positive decisions may propagate from bottom to top across the graph: they influence the decisions of the parent nodes and of their ancestors in a recursive way, by traversing the graph towards higher level nodes/classes. On the contrary negative decisions do no affect decisions of the parent node (that is they do not propagate from bottom to top, eq. 1).
4. Negative predictions for a given node (taking into account the local decision of its descendants) are propagated to the descendants, to preserve the consistency of the hierarchy according to the true path rule. On the contrary positive decisions do not influence decisions of child nodes (eq. 2).

The ensemble combines the local predictions of the base learners associated to each node with the positive decisions that come from the bottom of the hierarchy, and with the negative decisions that spring from the higher level nodes. More precisely, base classifiers estimates local probabilities $\hat{p}_i(x)$ that a given example x belongs to class ω_i, but the core of the algorithm is represented by the evaluation phase, where the ensemble provides an estimate of the "consensus" global probability $p_i(x)$.

In [17] we proposed a basic algorithm based on the "True Path Rule" (the TPR algorithm), by which, given the set $\phi_i(x)$ of the children of node i for which

we have a positive prediction for a given example x:

$$\phi_i(x) = \{j | j \in \text{child}(i), d_j(x) = 1\} \tag{3}$$

we can compute the consensus probability of the ensemble. The global consensus probability $p_i(x)$ of the ensemble depends both on the local prediction $\hat{p}_i(x)$ and on the prediction of the nodes belonging to $\phi_i(x)$:

$$p_i(x) = \frac{1}{1 + |\phi_i(x)|} \left(\hat{p}_i(x) + \sum_{j \in \phi_i(x)} p_j(x) \right) \tag{4}$$

The decision $d_i(x)$ at node/class i is set to 1 if $p_i(x) > t$, and to 0 otherwise (a natural choice for t is 0.5). In the leaf nodes the sum of eq. 4 disappears and eq. 4 reduces to $p_i(x) = \hat{p}_i(x)$. In this way positive predictions propagate from bottom to top. On the contrary if for a given node $d_i(x) = 0$, then this decision is propagated to its subtree.

Note that with this basic version of the TPR algorithm there is no way to explicitly balance the local prediction $\hat{p}_i(x)$ at node i with the positive predictions coming from its offsprings (eq. 4). By balancing the local predictions with the positive predictions coming from the ensemble we can explicitly modulate the interplay between local and descendant predictors. To this end we introduce a *parent weight* w_p, $0 \leq w_p \leq 1$, such that if $w_p = 1$ the decision at node i depends only by the local predictor, otherwise the prediction is shared proportionally to w_p and $1 - w_p$ between respectively the local parent predictor and the set of its children:

$$p_i(x) = w_p \cdot \hat{p}_i(x) + \frac{1 - w_p}{|\phi_i(x)|} \sum_{j \in \phi_i(x)} p_j(x) \tag{5}$$

In this way we can balance the weight of the prediction between the local component at node i and the component coming from its children, thus obtaining the *weighted TPR (TPR-w)* hierarchical ensemble algorithm.

The pseudocode of the TPR-w method is presented in Algorithm 1.

The algorithm is characterized by two main for loops: the external for (from row 1 to 30) handles a per level bottom-up traversal of the tree, while the internal (from row 2 to 29) scans the nodes at each level. If a node is a leaf (row 3), then the consensus probability p_i is equal to the local probability $\hat{p}_i(x)$. Note that a positive decision is taken if $p_i(x)$ is larger than a threshold t (row 5). If a node is not a leaf (row 10), at first the set $\phi_i(x)$ collects all the children nodes for which we have a positive prediction, and the consensus probability p_i of the ensemble is computed by considering the weighted local estimate of the probability $\hat{p}_i$ and the weighted probabilities computed by the children nodes for which a positive decision has been taken (row 13). In case of a negative decision for a node i, all the predictions relative to the subtree rooted at i are set to negative and their probabilities are set to p_i if larger than p_i. (rows 19-27). The algorithm provides both the multilabels associated to the example x and the probabilities p_i that a given example belongs to the class i, $1 \leq i \leq m$.

Algorithm 1 Weighted True Path Rule (TPR-w) hierarchical ensemble

Input:
- a test example x
- tree T of the m hierarchical classes
- set of m classifiers (one for each node) each predicting $\hat{p}_i(x)$, $1 \leq i \leq m$
- the weight w_p of the local prediction.

for all levels k of T from bottom to top **do**
  for all nodes i at level k **do**
    if i is a leaf **then**
      $p_i(x) \leftarrow \hat{p}_i(x)$
      if $p_i(x) > t$ **then**
        $d_i(x) \leftarrow 1$
      else
        $d_i(x) \leftarrow 0$
      end if
    else
      $\phi(x) \leftarrow \{j | j \in \text{child}(i), d_j(x) = 1\}$
      if $|\phi_i(x)| > 0$ **then**
        $p_i(x) \leftarrow w_p \cdot \hat{p}_i(x) + \frac{1-w_p}{|\phi_i(x)|} \sum_{j \in \phi_i(x)} p_j(x)$
      else
        $p_i(x) \leftarrow \hat{p}_i(x)$
      end if
      if $p_i(x) > t$ **then**
        $d_i(x) \leftarrow 1$
      else
        $d_i(x) \leftarrow 0$
        for all $j \in$ subtree(i) **do**
          $d_j(x) \leftarrow 0$
          if $p_j(x) > p_i(x)$ **then**
            $p_j(x) \leftarrow p_i(x)$
          end if
        end for
      end if
    end if
  end for
end for

Output:
For each node i:
- the ensemble decisions $d_i(x) = \begin{cases} 1 & \text{if } x \text{ belongs to node } i \\ 0 & \text{otherwise} \end{cases}$
- the probabilities $p_i(x)$ that x belongs to the node $i \in T$

The bottom-up per level traversal of the tree assures that all the offsprings of a given node i are taken into account for the ensemble prediction. For the same reason we can safely set the classes belonging to the subtree rooted at i

to negative, when $d_i(x)$ is set to 0. It is worth noting that we have a two-way asymmetric flow of information across the tree: positive predictions for a node influence its ancestors, while negative predictions influence its offsprings.

3 Experimental set-up

We predicted the functions of genes of the unicellular eukaryote *S. cerevisiae* using 7 different data sets and the *FunCat* taxonomy.

For each data set we evaluated the performance of four different ensembles: the *Flat* ensemble, that does not take into account the hierarchical structure of the data, the Hierarchical *Top-down* [18, 19], the basic True Path Rule (*TPR*) hierarchical ensemble and the proposed weighted variant (*TPR-w* described in the previous section). The hierarchical Top-down algorithm classifies an example x, where $d_i(x)$ is the classifier decision at node i and $root(T)$ denotes the set of nodes at the first level of the tree T, in the following way:

$$y_i = \begin{cases} d_i(x) \text{ if } i \in root(T) \\ d_i(x) \text{ if } i \notin root(T) \wedge y_{par(i)} = 1 \\ 0 \quad\quad \text{if } i \notin root(T) \wedge y_{par(i)} = 0 \end{cases}$$

For each ensemble we used as base learners linear Support Vector Machines (SVMs) with probabilistic output [11]. The performance of the ensembles have been compared using 5-fold cross-validation techniques. The selection of the w_p parameter in *TPR-w* ensembles have been performed by internal cross-validation. The threshold t of *TPR* ensembles has been set to 0.5 in all the experiments.

For the prediction of gene function in the yeast we used 7 bio-molecular data sets. For each data set we selected only the genes annotated to FunCat [2], using the *HCgene* R package [20]. We also removed the genes annotated only with the "99" FunCat class ("Unclassified proteins") and selected classes with at least 20 positive examples, in order to get a not too small set of positive examples for training. As negative examples we selected at each node/class genes not annotated to that node, but annotated to its parent. From the data sets we removed also uninformative features (e.g. features with the same value for all the available examples). At the end of these pre-processing steps we obtained data whose characteristics are summarized in Tab. 1.

Considering the unbalance between positive and negative examples, we adopted the classical F-score to jointly take into account the precision and recall of the ensemble for each class of the hierarchy.

Moreover, we used also the *Hierarchical F-measure* that represents a generalization of the classical F-measure, in order to take into account the hierarchical nature of functional annotation [27].

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

[2] We used funcat-2.1 scheme, and funcat-2.1_data_20070316, available from the MIPS web site (http://mips.gsf.de/projects/funcat).

Table 1: Data sets

Data set	Description	n.samples	n. feat.	n.class
Pfam-1	protein domain binary data from *Pfam* [21]	3529	4950	211
Pfam-2	protein domain log E data from *Pfam* [22]	3529	5724	211
Phylo	phylogenetic data [14]	2445	24	187
Expr	gene expression data [23, 24]	4532	250	230
PPI-BG	PPI data from *BioGRID* [25]	4531	5367	232
PPI-VM	PPI data from von Mering experiments [26]	2338	2559	177
SP-sim	Sequence pairwise similarity data [15]	3527	6349	211

that gene. Conversely, hierarchical recall measures the average fraction of each true path that is covered by some predicted path for that gene. More precisely, given a general taxonomy G representing the graph of the functional classes, for a given gene/gene product x consider the graph $P(x) \subset G$ 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. Given a leaf $p \in P(x)$, let be $\uparrow p$ the set of ancestors of the node p that belong to $P(x)$, and given a leaf $c \in C(x)$, let be $\uparrow c$ the set of ancestors of the node c that belong to $C(x)$. The original definitions of Hierarchical Precision (HP), Hierarchical Recall (HR) and Hierarchical F-score (HF) [27], with the tree forests of FunCat can be simplified as follows:

$$HP = \frac{1}{|l(P(x))|} \sum_{p \in l(P(x))} \frac{|C(x) \cap \uparrow p|}{|\uparrow p|}$$
$$HR = \frac{1}{|l(C(x))|} \sum_{c \in l(C(x))} \frac{|\uparrow c \cap P(x)|}{|\uparrow c|}$$
$$HF = \frac{2 \cdot HP \cdot HR}{HP + HR} \tag{6}$$

An overall high hierarchical precision is indicative of most predictions being ancestors of the correct predictions, or in other words that the predictor is able to detect the most general functions of genes/gene products. On the other hand a high average hierarchical recall indicates that most predictions are successors of the actual, or that the predictors are able to detect the most specific functions of the genes.

4 Results

At first we compared the performance of ensemble methods considering the "per class" F-measure averaged across all FunCat classes for each data set. The results show that hierarchical methods largely outperform flat ensembles: flat ensembles obtain an average F-measure across the 7 data sets used in the experiments of

Table 2: Per class F-measure results. Flat: flat ensemble; HTD: Hierarchical Top-Down ensembles; TPR: True Path Rule hierarchical ensembles; TPR-w True Path Rule weighted hierarchical ensembles.

Data set	Flat	HTD	TPR	TPR-w
Pfam-1	0.2816	0.4041	0.3622	0.4037
Pfam-2	0.1153	0.2056	0.1562	0.2197
Phylo	0.0711	0.0067	0.0625	0.0906
Expr	0.0752	0.0623	0.0702	0.0773
PPI-BG	0.1730	0.2690	0.2344	0.2946
PPI-VM	0.2145	0.3589	0.2613	0.3558
SP-sim	0.1121	0.2489	0.1306	0.2540
Average	0.1489	0.2222	0.1824	0.2414

0.15 against respectively 0.22, 0.18 and 0.24 with Top-down, TPR and TPR-w ensembles (Tab. 2).

As explained in the experimental set-up (Sect. 3), the F hierarchical measure is a more appropriate performance metric for the hierarchical classification of gene functions. Tab. 3 shows that on the average TPR-w achieves the best results: 0.34 versus 0.25 (TPR) and 0.29 (Top-down ensembles). Note that TPR-w obtains equal or better results than Top-down ensembles with respect to all the data sets. More precisely considering 5-fold cross validation results for each of the 7 considered data sets TPR-w reported better results than Top-down at 0.05 significance level on 5 tasks, according to the 5-fold cross-validated paired t-test [28]. The basic TPR ensemble on the contrary achieves slightly worse results than the Top-down. These results show that we need the weighted version of TPR ensembles to significantly enhance Top-down predictions.

Table 3: Hierarchical F-measures results. HTD: Hierarchical Top-Down ensembles; TPR: True Path Rule hierarchical ensembles; TPR-w True Path Rule weighted hierarchical ensembles. Statistically significant difference at 0.05 significance level are in boldface.

Data set	HTD	TPR	TPR-w
Pfam-1	0.4123	0.3080	0.4131
Pfam-2	0.3406	0.2684	**0.3700**
Phylo	0.0497	0.2010	**0.2174**
Expr	0.1166	0.1696	**0.1784**
PPI-BG	0.3226	0.2670	**0.3485**
PPI-VM	0.3977	0.2796	0.4000
SP-sim	0.4251	0.2398	**0.4472**
Average	0.2949	0.2468	0.3392

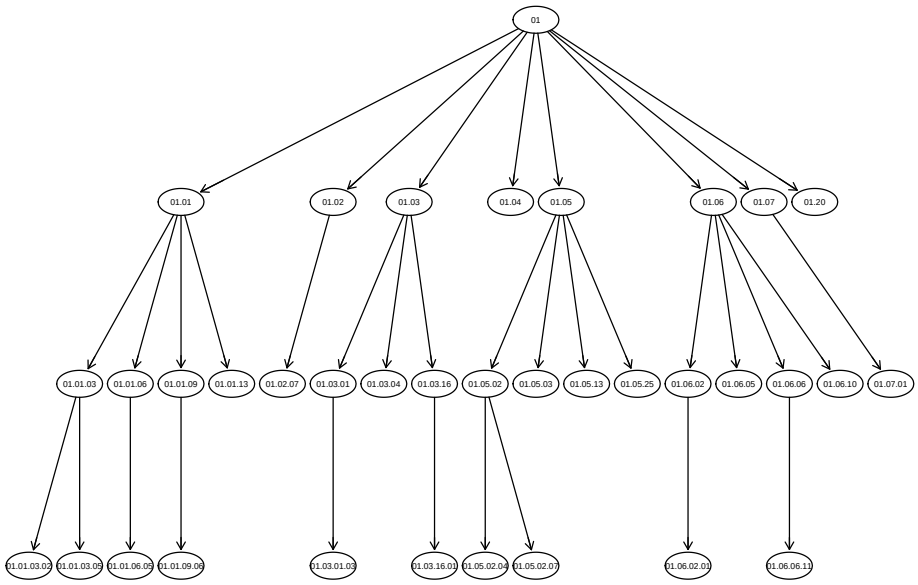

Fig. 2: Tree of the FunCat classes rooted at Funcat ID=01 (Metabolism).

Even if the main goal of this work consists in the development of a hierarchical algorithm that can be applied to the prediction of the overall taxonomy of a gene, we can restrict the analysis to specific subtrees of the taxonomy. For instance, Tab. 4 shows the results restricted to the subtree rooted at the "Metabolism" FunCat class (FunCat ID = 01, Fig.2).

A specific advantage of the TPR-w ensembles is the capability of tuning precision and recall rates, through the parameter parent-weight w_p (eq. 5). Fig. 3 shows, the hierarchical precision, recall and F-measure as functions of the parameter w_p. For small values of w_p (w_p can vary from 0 to 1) the weight of the decision of the parent local predictor is small, and the ensemble decision depends mainly by the positive predictions of the offsprings nodes (classifiers): as a consequence we obtain a higher hierarchical recall for the TPR-w ensemble. On the contrary higher values of w_p correspond to a higher weight of the "parent" local predictor, with a resulting higher precision. The opposite trends of precision and recall are quite clear in all graphs of Fig. 3. The best F-score is in "middle" values of the parameter parent-weight: in practice in most of the analyzed data sets the best F-measure is achieved for w_p between 0.5 and 0.8, but if we need higher recall rates (at the expense of the precision) we can choose lower w_p val-

Table 4: Classification results on the FunCat tree rooted at "Metabolism", using Pfam-1 data. Each row represents a functional class of the FunCat taxonomy. Prec. stands for precision, Rec. recall, Sp. specificity, F F-measure, Acc. accuracy.

FunCat ID	Description	Prec.	Rec.	Sp.	F	Acc.
01	Metabolism	0.83	0.59	0.93	0.69	0.80
01.01	amino acid metabolism	0.62	0.34	0.98	0.45	0.94
01.01.03	assimilation of ammonia, metabolism of the glutamate group	0.27	0.15	0.99	0.19	0.98
01.01.03.02	metabolism of glutamate	0.37	0.32	0.99	0.34	0.99
01.01.03.05	metabolism of arginine	0.00	0.00	0.99	0.00	0.99
01.01.06	metabolism of the aspartate family	0.38	0.22	0.99	0.28	0.98
01.01.06.05	metabolism of methionine	0.53	0.29	0.99	0.37	0.99
01.01.09	metabolism of the cysteine - aromatic group	0.49	0.26	0.99	0.34	0.97
01.01.13	regulation of amino acid metabolism	0.10	0.03	0.99	0.05	0.98
01.02	nitrogen, sulfur and selenium metabolism	0.55	0.20	0.99	0.29	0.97
01.02.07	regulation of nitrogen, sulfur and selenium metabolism	0.27	0.11	0.99	0.16	0.99
01.03	nucleotide/nucleoside/nucleobase metabolism	0.65	0.35	0.98	0.46	0.95
01.03.01	purin nucleotide/nucleoside/nucleobase metabolism	0.72	0.40	0.99	0.52	0.98
01.03.01.03	purine nucleotide/nucleoside/nucleobase anabolism	0.61	0.29	0.99	0.39	0.99
01.03.04	pyrimidine nucleotide/nucleoside/nucleobase metabolism	0.63	0.42	0.99	0.51	0.98
01.03.16	polynucleotide degradation	0.52	0.27	0.99	0.36	0.98
01.03.16.01	RNA degradation	0.54	0.29	0.99	0.37	0.98
01.04	phosphate metabolism	0.81	0.61	0.98	0.70	0.94
01.05	C-compound and carbohydrate metabolism	0.79	0.50	0.97	0.61	0.91
01.05.02	sugar, glucoside, polyol and carboxylate metabolism	0.65	0.35	0.99	0.46	0.98
01.05.02.04	sugar, glucoside, polyol and carboxylate anabolism	0.55	0.33	0.99	0.41	0.99
01.05.02.07	sugar, glucoside, polyol and carboxylate catabolism	1.00	0.09	1.00	0.18	0.98
01.05.03	polysaccharide metabolism	0.78	0.25	0.99	0.38	0.98
01.05.25	regulation of C-compound and carbohydrate metabolism	0.47	0.16	0.99	0.24	0.96
01.06	lipid, fatty acid and isoprenoid metabolism	0.75	0.44	0.98	0.56	0.95
01.06.02	membrane lipid metabolism	0.76	0.41	0.99	0.54	0.98
01.06.02.01	phospholipid metabolism	0.69	0.36	0.99	0.48	0.98
01.06.05	fatty acid metabolism	0.42	0.15	0.99	0.22	0.99
01.06.06	isoprenoid metabolism	0.65	0.34	0.99	0.45	0.99
01.06.06.11	tetracyclic and pentacyclic triterpenes metabolism	0.61	0.23	0.99	0.34	0.99
01.06.10	regulation of lipid, fatty acid and isoprenoid metabolism	0.86	0.24	0.99	0.37	0.99
01.07	metabolism of vitamins, cofactors, and prosthetic groups	0.74	0.29	0.99	0.42	0.96
01.07.01	biosynthesis of vitamins, cofactors, and prosthetic groups	0.72	0.32	0.99	0.44	0.97
01.20	secondary metabolism	0.80	0.11	0.99	0.20	0.98

ues, and higher values of w_p are needed if precision is our first aim. It is worth noting that we may vary the threshold t to obtain precision recall curves for a fixed value of w_p. In other words we may obtain different precision-recall curves for different value of w_p: the parent weight is a global parameter that affect the general precision/recall characteristics of the ensemble.

5 Conclusions

F hierarchical measures results show that TPR-w achieves equal or better results than the basic TPR algorithm and the Top-down hierarchical strategy, and all the hierarchical strategies achieve significantly better results than flat classification methods, using the classical "per-class" F-measure.

Another advantage of TPR-w consists in the possibility of tuning precision and recall by using a global strategy: large values of the parent weight improve

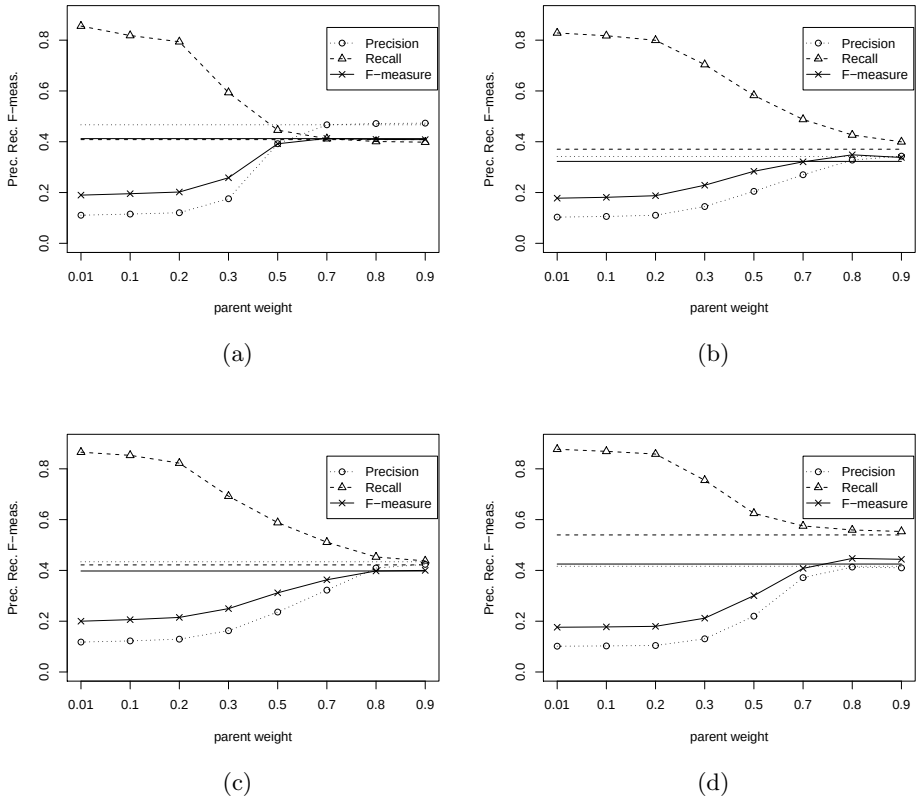

Fig. 3: Hierarchical Precision, Recall and F-measure as a function of the parent weight in TPR-w ensembles. Horizontal lines refers to top-down ensembles. (a) Protein domain binary data; (b) PPI BioGRID data; (c) PPI Von Mering data (d) Pairwise sequence similarity data.

the precision, and small values the recall. The choice to favour precision or recall depends on the researcher's experimental objectives. In most data sets the best compromise between precision and recall is achieved for weights in the range between 0.5 and 0.8, that is giving a weight equal or larger to the local predictor with respect to the predictions taken by its offsprings.

Results show that also using a single source of evidence we can obtain a very high precision and recall for specific trees of the FunCat forest, but the overall results need to be improved for the genome-wide prediction of gene function. To this end, we need to integrate multiple data sources to obtain methods to predict function of hypothetical genes, or to discover or complete the functional annotation of genes whose function is incomplete or unknown. To this end the proposed approach can be easily integrated with at least three different general

strategies for biomolecular data integration: vector space integration [14], kernel fusion [15] and ensemble methods [16]. Indeed for each node/class of the tree we may substitute a classifier trained on a specific type of biomolecular data with a classifier trained on concatenated vectors of different data, or trained on a (weighted) sum of kernels, or with an ensemble of learners each trained on a different type of data.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments and suggestions, and gratefully acknowledge partial support by the PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the authors' views.

References

[1] The Gene Ontology Consortium: Gene ontology: tool for the unification of biology. Nature Genet. **25** (2000) 25–29

[2] 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

[3] Tsoumakas, G., Katakis, I.: Multi label classification: An overview. International Journal of Data Warehousing and Mining **3** (2007) 1–13

[4] Barutcuoglu, Z., Schapire, R., Troyanskaya, O.: Hierarchical multi-label prediction of gene function. Bioinformatics **22** (2006) 830–836

[5] Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning **73** (2008) 185–214

[6] 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)

[7] Astikainen, K., Holm, L., Pitkanen, E., Szedmak, S., Rousu, J.: Towards structured output prediction of enzyme function. BMC Proceedings **2** (2008)

[8] Blockeel, H., Schietgat, L., Clare, A.: Hierarchcial multilabel classification trees for gene function prediction. In Rousu, J., Kaski, S., Ukkonen, E., eds.: Probabilistic Modeling and Machine Learning in Structural and Systems Biology, Tuusula, Finland, Helsinki University Printing House (2006)

[9] 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)

[10] Obozinski, G., Lanckriet, G., Grant, C., M., J., Noble, W.: Consistent probabilistic output for protein function prediction. Genome Biology **9** (2008)

[11] Lin, H., Lin, C., Weng, R.: A note on Platt's probabilistic outputs for support vector machines. Machine Learning **68** (2007) 267–276

[12] Valencia, A.: Automatic annotation of protein function. Curr. Opin. Struct. Biol. **15** (2005) 267–274

[13] Noble, W., Ben-Hur, A.: Integrating information for protein function prediction. In Lengauer, T., ed.: Bioinformatics - From Genomes to Therapies. Volume 3. Wiley-VCH (2007) 1297–1314
[14] Pavlidis, P., Weston, J., Cai, J., Noble, W.: Learning gene functional classification from multiple data. J. Comput. Biol. **9** (2002) 401–411
[15] Lanckriet, G., De Bie, T., Cristianini, N., Jordan, M., Noble, W.: A statistical framework for genomic data fusion. Bioinformatics **20** (2004) 2626–2635
[16] Re, M., Valentini, G.: Ensemble based data fusion for gene function prediction. In Kittler, J., Benediktsson, J., Roli, F., eds.: Multiple Classifier Systems. Eighth International Workshop, MCS 2009, Reykjavik, Iceland. Volume 5519 of Lecture Notes in Computer Science., Springer (2009) 448–457
[17] Valentini, G.: True path rule hierarchical ensembles. In Kittler, J., Benediktsson, J., Roli, F., eds.: Multiple Classifier Systems. Eighth International Workshop, MCS 2009, Reykjavik, Iceland. Volume 5519 of Lecture Notes in Computer Science., Springer (2009) 232–241
[18] Rousu, J., Saunders, C., Szdemak, S., Shawe-Taylor, J.: Learning hierarchical multi-category text classification models. In: Proc. of the 22nd Int. Conf. on Machine Learning, Omnipress (2005) 745–752
[19] 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
[20] Valentini, G., Cesa-Bianchi, N.: Hcgene: a software tool to support the hierarchical classification of genes. Bioinformatics **24** (2008) 729–731
[21] Deng, M., Chen, T., Sun, F.: An integrated probabilistic model for functional prediction of proteins. In: Proc 7th Int Conf Comp Mol Biol. (2003) 95–103
[22] 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
[23] 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
[24] Gasch, P., et al.: Genomic expression programs in the response of yeast cells to environmental changes. Mol.Biol.Cell **11** (2000) 4241–4257
[25] 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
[26] 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
[27] Verspoor, K., Cohn, J., Mnizewski, S., Joslyn, C.: A categorization approach to automated ontological function annotation. Protein Science **15** (2006) 1544–1549
[28] Dietterich, T.: Approximate statistical test for comparing supervised classification learning algorithms. Neural Computation **10** (1998) 1895–1924