Machine Learning manuscript No.
(will be inserted by the editor)

Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference

Nicolò Cesa-Bianchi · Matteo Re · Giorgio Valentini

Received: date / Accepted: date

Abstract Gene function prediction is a complex multilabel classification problem with several distinctive features: the hierarchical relationships between functional classes, the presence of multiple sources of biomolecular data, the unbalance between positive and negative examples for each class, the complexity of the whole-ontology and genome-wide dimensions. Unlike previous works, which mostly looked at each one of these issues in isolation, we explore the interaction and potential synergy of hierarchical multilabel methods, data fusion methods, and cost-sensitive approaches on whole-ontology and genome-wide gene function prediction. Besides classical top-down hierarchical multilabel ensemble methods, in our experiments we consider two recently proposed multilabel methods: one based on the approximation of the Bayesian optimal classifier with respect to the hierarchical loss, and one based on a heuristic approach inspired by the true path rule for the biological functional ontologies. Our experiments show that key factors for the success of hierarchical ensemble methods are the integration and synergy among multilabel hierarchical, data fusion, and cost-sensitive approaches, as well as the strategy of selecting negative examples.

Keywords hierarchical multilabel classification · data integration · cost-sensitive classification · ensemble methods · gene function prediction

1 Introduction

Multilabel learning (see, e.g., Tsoumakas and Katakis (2007) for a review) is an emerging thread in machine learning research, as witnessed by the number of recent papers and workshops on this topic (Zhang and Zhou 2007; Amit *et al.* 2007; Dembczynski *et al.* 2010a; Zhang *et al.* 2010; Tsoumakas *et al.* 2010). The applications of multilabel classification span a large range of real-world applications, such as music categorization, web search and mining, semantic scene classification, directed marketing and functional genomics (Zhang and Zhou 2006; Trohidis *et al.* 2008; Dimou *et al.* 2009).

DSI, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
Tel.: +39-02-503-16225
Fax: +39-02-503-16373
E-mail: {cesa-bianchi,re,valentini}@dsi.unimi.it

Constraints between labels and, more in general, the issue of label dependence have been recognized to play a central role in multilabel learning (Dembczynski *et al.* 2010b). For instance, gene function prediction (*GFP*) is a complex multilabel classification problem where functional classes are structured according to a predefined hierarchy — a directed acyclic graph in the Gene Ontology (The Gene Ontology Consortium 2000) or a forest of trees in the Functional Catalogue (Ruepp *et al.* 2004). In this respect gene function prediction can be regarded as a paradigmatic multilabel classification problem, where the exploitation of a priori knowledge about the hierarchical relationships between the labels can dramatically improve classification performance (Obozinski *et al.* 2008; Mostafavi and Morris 2009; Cesa-Bianchi and Valentini 2010).

GFP is challenging for machine learning because of several reasons:

- *Large number of functional classes*: hundreds for Functional Catalogue (FunCat) or thousands for the Gene Ontology (GO).
- *Multiple annotations for each gene*: since each gene may belong to more than one class (sometimes to tens of classes) at the same time, the classification problem is multilabel.
- *Hierarchical relationships between functional classes*: labels are not independent because functional classes are hierarchically organized; in general, known functional relationships (such as taxonomies) can be exploited to incorporate a priori knowledge in learning algorithms or to introduce explicit constraints between labels.
- *Multiple sources of data*: high-throughput biotechnologies make available an increasing number of sources of genomic and proteomic data. Hence, in order to exploit all the information available for each gene, we need learning methods that are able to integrate different data sources.
- *Complex and noisy data*: data are usually complex (e.g., high-dimensional, large-scale, graph-structured) and noisy.
- *Unbalanced classes*: typically functional classes are severely unbalanced, with positive examples largely outnumbered by negatives.
- *Definition of negative examples*: since we only have positive annotations, the notion of negative example is not uniquely determined, and different strategies of choosing negative examples can be in principle applied.
- *Different reliability of functional labels*: functional annotations have different degrees of evidence; that is, each label is assigned to a gene with a specific level of reliability.

Several machine learning approaches have been proposed to deal with the above issues. Some take advantage of the intrinsic hierarchical nature of gene function prediction by explicitly considering the relationships between functional classes (Eisner *et al.* 2005; Blockeel *et al.* 2006; Shahbaba and Neal 2006; Vens *et al.* 2008). In particular, in order to improve the multilabel classification performance on the overall functional taxonomy, ensemble methods hierarchically combine predictions of base learners, where each base learner is trained on a specific functional class (Barutcuoglu *et al.* 2006; Obozinski *et al.* 2008).

Other approaches focus primarily on the integration of multiple sources of data, since each type of genomic data captures only some aspects of the genes to be classified, and a specific source can be useful to learn a specific functional class while being irrelevant to others. In the literature, many approaches have been proposed to deal with this topic. For example, functional linkage networks integration (Chua *et al.* 2007), kernel fusion (Lanckriet *et al.* 2004b), vector space integration (Pavlidis *et al.* 2002), and ensemble systems (Re and Valentini 2010c).

Without taking into account the hierarchical relationships between the functional classes, data integration exhibits serious inconsistencies due to the violation of the *true path rule*,

governing the functional annotations of genes both in the GO and in FunCat taxonomies (The Gene Ontology Consortium 2000; Ruepp *et al.* 2004). Similarly, hierarchical approaches which do not consider different sources of data do not have enough information to provide reliable predictions. Finally, it is well known that unbalanced classification problems, such as *GFP* problems, require cost-sensitive learning strategies to effectively predict the examples belonging to the less represented classes.

Most of the proposed *GFP* methods consider only some of the aforementioned issues. For instance, several methods provide multilabel classifications (Troyanskaya *et al.* 2003; Tsuda *et al.* 2005; Xiong *et al.* 2006) or integrate multiple data sources (Lanckriet *et al.* 2004b; Re and Valentini 2010c), yet they do not take into account the hierarchical relationships between classes. Other methods, instead, are hierarchical but disregard the unbalance between positive and negative examples (Barutcuoglu *et al.* 2006; Karaoz *et al.* 2004). In this respect, it becomes difficult to assess the impact of each issue on the overall prediction performance.

In this work we perform an analysis of the specific contribution of each issue in the context of *GFP*. In particular, we investigate whether hierarchical constraints embedded in multilabel prediction can boost performance on *GFP* problems, and whether data fusion or cost-sensitive techniques may lead to further significant improvements. Indeed, in the context of automatic document classification hierarchical cost-sensitive approaches have been proven to enhance the classification performance with respect to classical multiclass-multilabel flat methods (Cai and Hofmann 2004). More specifically, the main aim of this paper is to study and quantify the synergy among learning strategies, addressing specific aspects of the *GFP* problem. To this end, we integrate data fusion methods based on kernel fusion and ensemble algorithms (Re and Valentini 2010c) with hierarchical multilabel cost-sensitive algorithms (Cesa-Bianchi and Valentini 2010; Valentini 2011). The resulting system is tested on genome and ontology-wide classification of genes according to the FunCat taxonomy. Our experiments reveal the impact of each learning component on the overall performance. Finally, we propose a new general methodology for integrating hierarchical multilabel techniques, data fusion, and cost-sensitive methods for the *GFP* problem.

In the next section the data fusion, multilabel hierarchical, and cost-sensitive methods are introduced. Section 2 provides an overview of the machine learning methods applied to *GFP*. Then extensive empirical results on real genome-wide and whole-ontology *GFP* problems are presented, together with a discussion on the synergic effects among the different learning components. The paper is concluded summarizing the main findings and proposing new research directions.

2 Related work

Historically, the first attempts to computationally predict the function of genes or gene products were based on algorithms able to infer similarities between sequences (Altschul *et al.* 1990, 1997). Today this is one of the standard methods of assigning functions to proteins in newly sequenced organisms (Juncker *et al.* 2009). Similarly, functional properties can be detected using global or local structure comparison algorithms between proteins —see, e.g., Loewenstein *et al.* (2009) for a recent review. In this context, the integration of different sequence and structure-based prediction methods represents a major challenge (Prlic *et al.* 2007).

2.1 Machine learning-based gene function prediction methods

Recently, several *GFP* methods, mostly based on a machine learning approach, have been proposed. They can be schematically grouped in four main families

1. Label propagation methods
2. Methods based on decision trees
3. Kernel methods for structured output spaces
4. Hierarchical ensemble methods

This grouping is neither exhaustive nor strict, meaning that certain methods do not belong to any of these groups, and others belong to more than one. It is worth noting that the term ensemble is used in this paper in a very wide sense: indeed, we apply it to both learners predicting different targets (such as in hierarchical ensemble methods), and to learners predicting the same task (such as in bagging or random forests).

Label propagation methods. Also known in literature as network-based methods or functional association or linkage networks, these methods usually represent each dataset through an undirected graph $G = (V,E)$, where nodes $v \in V$ correspond to gene/gene products, and edges $e \in E$ are weighted according to the evidence of co-functionality implied by data source (Marcotte *et al.* 1999; Vazquez *et al.* 2003). By exploiting "proximity relationships" between connected nodes, these algorithms are able to transfer annotations from previously annotated (labeled) nodes to unannotated (unlabeled) ones through a learning process inherently transductive in nature. Indeed, these methods are based on transductive label propagation algorithms: they predict the labels of unannotated examples without using a global predictive model (Troyanskaya *et al.* 2003; Chua *et al.* 2007; Mostafavi *et al.* 2008).

Label propagation algorithms adopt different strategies to learn the unlabeled nodes. For example, simple "guilt-by-association" methods (Oliver 2000; McDermott and Samudrala 2005), methods based on the evaluation of the functional flow in graphs (Vazquez *et al.* 2003; Nabieva *et al.* 2005), methods based on Hopfield networks (Karaoz *et al.* 2004), and methods based on Markov (Deng *et al.* 2004) and Gaussian Random Fields (Tsuda *et al.* 2005; Mostafavi *et al.* 2008).

Bengio *et al.* (2006) showed that different graph-based algorithms can be cast into a common framework where a quadratic cost objective function is minimized. In this framework closed form solutions can be derived by solving a linear system of size equal to the cardinality of nodes (proteins), or using fast iterative procedures such as the Jacobi method (Saad 1996). A network-based approach, alternative to label propagation and exhibiting strong theoretical predictive guarantees in the so-called mistake bound model, has been recently proposed by Cesa-Bianchi *et al.* (2010b). This alternative method is extremely efficient: in most cases training and prediction take both time *sublinear* in the network size.

Decision tree-based methods. Clare and King (2003) proposed a hierarchical multilabel classification decision tree to predict gene functions by extending the classical C4.5 decision tree algorithm for multiclass classification (Quinlan 1986).

Vens *et al.* (2008) showed that separate decision tree models are less accurate than a single decision tree trained to predict all classes at once. In the context of the predictive clustering tree framework (Blockeel *et al.* 1998), Blockeel *et al.* (2006) proposed an improved version which they applied to the prediction of gene function in the yeast. Moreover, Schietgat *et al.* (2010) showed that ensembles of hierarchical multilabel decision trees are competitive with state-of-the-art statistical learning methods for DAG-structured prediction of gene function.

Kernel methods for structured output spaces. In this framework the multilabel hierarchical classification problem is solved globally: the multilabels are viewed as elements of a structured space modeled by suitable kernel functions (Tsochantaridis *et al.* 2005; Rousu *et al.* 2006; Lampert and Blaschko 2009). In particular, these methods treat structured prediction as a maximum a-posteriori prediction problem (Bakir *et al.* 2007). A structured Perceptron, and a variant of the structured support vector machine (Tsochantaridis *et al.* 2005), have been implemented in the *GOstruct* system and successfully applied to the the prediction of GO terms in mouse and other model organisms (Sokolov and Ben-Hur 2010). Structured output maximum-margin algorithms have been also applied to the tree-structured prediction of enzyme functions (Astikainen *et al.* 2008; Rousu *et al.* 2006).

Hierarchical ensemble methods. Several methods attempt to take advantage of the intrinsic hierarchical nature of GFP, explicitly considering the relationships between functional classes (Eisner *et al.* 2005; Blockeel *et al.* 2006; Shahbaba and Neal 2006; Vens *et al.* 2008; Jiang *et al.* 2008). Indeed, *flat* methods may introduce large inconsistencies in parent-child relationships between classes, and a hierarchical approach corrects "flat" predictions improving accuracy and consistency of the multilabel annotations of genes (Obozinski *et al.* 2008). In particular, hierarchical ensemble methods generally work via a two-step strategy:

1. Flat learning of the protein function on a per-term basis (a set of independent classification problems)
2. Combination of the predictions by exploiting the relationships between terms that govern the hierarchy of the functional classes.

In principle, any supervised learning algorithm can be used for step 1. Step 2 requires a proper combination of the predictions made at step 1.

Based on this algorithmic scheme, Barutcuoglu *et al.* (2006) proposed an ensemble algorithm that initially provides flat (possibly inconsistent) predictions for each term/class, and then combine them through a Bayesian network scheme acting as a "collaborative" error-correction step over all nodes. As an extension of this approach, two local strategies that take into account the relationships between GO nodes and a composite ensemble method have been proposed (Guan *et al.* 2008). Different strategies to hierarchically reconcile the output of an ensemble of learning machines trained to predict separately each GO term have been proposed by Obozinski *et al.* (2008): the results demonstrated that hierarchical multilabel methods can play a crucial role in improving gene function prediction performances. The multilabel hierarchical approaches studied in this paper belong to this research line (Cesa-Bianchi and Valentini 2010; Valentini 2011).

2.2 Data fusion methods for gene function prediction

The integration of multiple sources of heterogeneous biomolecular data is the key to the prediction of gene function at genome-wide level. Indeed, high-throughput biotechnologies make available increasing quantities of biomolecular data of different types, and several works pointed out that data integration plays a central role to improve the accuracy in *GFP* (Friedberg 2006).

The main approaches proposed in the literature can be schematically grouped in four categories (Noble and Ben-Hur 2007):

1. Functional association networks integration

2. Vector subspace integration
3. Kernel fusion
4. Ensemble methods

Functional association networks integration. In functional association networks, different graphs are combined to obtain the composite resulting network (Karaoz *et al.* 2004; Chua *et al.* 2007). This network is then processed by a transduction algorithm that assigns all missing labels. The simplest approaches adopt conjunctive/disjunctive techniques (Marcotte *et al.* 1999), or probabilistic evidence integration schemes (Troyanskaya *et al.* 2003). More recently, function specific composite networks have been constructed by weighting each data source: Tsuda *et al.* (2005) solved this problem by simultaneously optimize the Gaussian Random Fields applied to each data set and the weights associated to each network, while Myers and Troyanskaya (2007) construct a combined network by applying a Naive Bayes classifier. Another network-based approach models data fusion as a constrained linear regression problem (Mostafavi *et al.* 2008). Recently, the same authors showed that better performances can be achieved by optimizing weights on subsets of related GO terms exploiting the relationships between functional classes (Mostafavi and Morris 2010).

Vector Space Integration. In vector space integration vectorial data are concatenated to combine different data sources (desJardins *et al.* 1997). For instance, Pavlidis *et al.* (2002) concatenate different vectors, each one corresponding to a different source of genomic data, in order to obtain a larger vector that is used to train a standard SVM. A similar approach has been proposed by Guan *et al.* (2008), but they separately normalized each data source in order to take into account the data distribution in each individual vector space.

Kernel fusion. Thanks to the closure property with respect to the sum and other algebraic operators, kernels provide another valuable research direction for the integration of biomolecular data. Besides combining kernels linearly with fixed coefficients (Pavlidis *et al.* 2002), one may also use semidefinite programming to learn the coefficients (Lanckriet *et al.* 2004b). As methods based on semi-definite programming do not scale well to multiple data sources, more efficient methods for multiple kernel learning have been recently proposed (Sonnenburg *et al.* 2006; Rakotomamonjy *et al.* 2007). Kernel fusion methods, both with and without weighting the data sources, have been successfully applied to the classification of gene functions (Lanckriet *et al.* 2004a; Lewis *et al.* 2006; Cesa-Bianchi *et al.* 2010a)

Ensemble methods. Even if it seems quite natural to apply ensemble methods to genomic data fusion (Noble and Ben-Hur 2007), only a few ensemble methods have been so far applied to this task. Some examples include "late integration" of kernels trained on different sources (Pavlidis *et al.* 2002), Naive Bayes integration of the outputs of SVMs trained with multiple sources (Guan *et al.* 2008), and logistic regression for combining the output of several SVMs trained with different biomolecular data and kernels (Obozinski *et al.* 2008).

Recently, Re and Valentini (2010c) showed that simple ensemble methods, such as weighted voting or Decision Templates (Kuncheva *et al.* 2001) give results comparable to state-of-the-art data integration methods, exploiting at the same time the modularity and scalability that characterize most ensemble algorithms. Moreover, ensembles of learning machines are able to include new types of biomolecular data, or updates of data contained in

public databases, by training only the base learners associated with the new data, without retraining the entire ensemble (Re and Valentini 2010a). Compared to kernel fusion methods, ensemble methods are also more robust to noisy data (Re and Valentini 2010b).

3 Methods

In this section we describe the methods we applied to analyze the impact on the *GFP* problem of multilabel hierarchical strategies, data fusion, and cost-sensitive techniques. More precisely, in Sect. 3.1 we introduce the basic notation used throughout the paper. In Sect. 3.2 we briefly describe the weighted linear combination and kernel fusion techniques analyzed in this work. Next, we introduce three hierarchical classification methods based on ensembles of learning machines: Hierarchical Top-Down (HTD), Hierarchical Bayesian (HBAYES), and Hierarchical True Path Rule (TPR) ensembles. In Sect. 3.6 we introduce their cost-sensitive counterparts. In the last subsection 3.7 we briefly describe how we integrate hierarchical multilabel, data fusion, and cost-sensitive techniques.

3.1 Basic notation

We represent a gene g with a vector $x \in \mathbb{R}^d$ having d different features (e.g., presence or absence of interactions with other d genes, or gene expression levels in d different conditions). A gene g is assigned to one or more functional classes in the set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$ structured according to a FunCat tree T [1]. The assignments are coded through a vector of multilabels $v = (v_1, v_2, \ldots, v_m) \in \{0,1\}^m$, where g belongs to class ω_i if and only if $v_i = 1$.

In the FunCat tree T, nodes correspond to classes, and edges to relationships between classes. We denote with i the node corresponding to class ω_i. We represent by child(i) the set of nodes that are children of i and by par(i) the parent of i, so that $v_{\text{par}(i)} = 1$ means that the gene under consideration belongs to the parent class of i. The multilabel of a gene g is built starting from the set of the most specific classes occurring in the gene's FunCat annotation; 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*, according to which if g belongs to a class/node i, then it also belongs to par(i).

The hierarchical ensemble methods proposed in this paper train a set of calibrated classifiers, one for each node of the taxonomy T. These classifiers are used to derive estimates $\widehat{p}_i(g)$ of the probabilities $p_i(g) = \mathbb{P}\big(V_i = 1 \mid V_{\text{par}(i)} = 1, g\big)$ for all g and i, where $(V_1, \ldots, V_m) \in \{0,1\}^m$ is the vector random variable modeling the unknown multilabel of a gene g.

Next we introduce: (1) data fusion techniques; (2) ensemble methods that infer a multilabel assignment $\widehat{y} = (\widehat{y}_1, \ldots, \widehat{y}_m) \in \{0,1\}^m$ based on estimates $\widehat{p}_1(g), \ldots, \widehat{p}_m(g)$; (3) their cost-sensitive variants.

3.2 Data fusion techniques

Data integration is performed locally at each node/class of the FunCat taxonomy. We consider two techniques: ensemble (weighted voting) and kernel fusion.

[1] The root of T is a dummy class ω_0, which every gene belongs to, that we added to facilitate the processing

Given L different sources $D_1,\ldots,D_L$ of biomolecular data, we train node classifiers $c_{t,i}$ on the data set D_t, one for each class $\omega_i, i = 1,\ldots,m$. Let $\widehat{p}_{t,i}(g)$ be the estimate of the probability $\mathbb{P}\big(V_i = 1 \mid V_{\mathrm{par}(i)} = 1, g\big)$ computed by the classifier $c_{t,i}$.

A simple way to integrate L different data sources is via the weighed linear combination rule (Kittler *et al.* 1998). The resulting ensemble estimates the probability that a given gene g belongs to class ω_i by a convex combination of the probabilities estimated by each base learner trained on a different "view" of the data:

$$\widehat{p}_i(g) = \frac{1}{\sum_{s=1}^{L} F_s} \sum_{t=1}^{L} F_t \, \widehat{p}_{t,i}(g) \tag{1}$$

where F_t is the F-measure assessed on the training data for the t-th base learner. The choice of the F-measure instead of the accuracy is motivated by the fact that gene classes are largely unbalanced (there are fewer positive examples than negative ones). Given a gene g, the decision $\widehat{y}_i$ of the ensemble about the class ω_i is taken using estimates (1),

$$\widehat{y}_i = \begin{cases} 1, & \text{if } \widehat{p}_i(g) > \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where output 1 corresponds to assigning class ω_i to g.

Another popular method to combine different sources of data is kernel fusion (Lanckriet *et al.* 2004b). Kernel fusion for data integration is based on the closure property of kernels with respect to the sum and other algebraic operators. Given a pair of genes g, g', and their corresponding pairs of feature vectors $x_t, x_t' \in D_t$, we implement a kernel averaging function $K_{\mathrm{ave}}(g,g')$ by simply averaging the output of kernel functions $K_1,\ldots,K_L$ specific to each data set,

$$K_{\mathrm{ave}}(g,g') = \frac{1}{L} \sum_{t=1}^{L} K_t(x_t, x_t') \; . \tag{3}$$

In our experiments we integrated the different data sets by simply summing their normalized kernel matrices. Then we trained the SVM using the resulting matrix. In this case we also use probabilistic SVMs (Lin *et al.* 2007) in order to obtain estimates of the posterior probabilities $\mathbb{P}(V_i = 1 \mid V_{\mathrm{par}(i)} = 1, g)$ for $i = 1,\ldots,m$.

3.3 Hierarchical Top-Down ensembles

The hierarchical Top-Down ensemble method (HTD) computes predictions in a top-down fashion (i.e., assigning $\widehat{y}_i$ before assigning the label of any j in the subtree rooted at i). The algorithm is straightforward: for each gene g, starting from the set of nodes at the first level of the tree T (denoted by $\mathrm{root}(T)$), the classifier associated to the node $i \in T$ computes whether the gene belongs to the class ω_i. If yes, the classification process continues recursively on the nodes $j \in \mathrm{child}(i)$; otherwise, it stops at node i, and the nodes belonging to the subtree rooted at i are all set to 0. In our setting we applied probabilistic classifiers as base learners trained to predict class ω_i associated to the node i of the hierarchical taxonomy. Their estimates $\widehat{p}_i(g)$ of $\mathbb{P}\big(V_i = 1 \mid V_{\mathrm{par}(i)} = 1, g\big)$ are used by the HTD ensemble to classify a gene g as follows

$$\widehat{y}_i = \begin{cases} \{\widehat{p}_i(g) > \frac{1}{2}\} & \text{if } i \in \mathrm{root}(T) \\ \{\widehat{p}_i(g) > \frac{1}{2}\} & \text{if } i \notin \mathrm{root}(T) \wedge \{\widehat{p}_{\mathrm{par}(i)}(g) > \frac{1}{2}\} \\ 0 & \text{if } i \notin \mathrm{root}(T) \wedge \{\widehat{p}_{\mathrm{par}(i)}(g) \leq \frac{1}{2}\} \end{cases}$$

where $\{x\} = 1$ if $x > 0$ otherwise $\{x\} = 0$. It is easy to see that this procedure ensures that the predicted multilabels $\widehat{y} = (\widehat{y}_1, \ldots, \widehat{y}_m)$ are consistent with the hierarchy.

3.4 Hierarchical Bayesian ensembles

The ensemble method HBAYES provides an approximation of the optimal Bayesian classifier w.r.t. the H-loss (Cesa-Bianchi *et al.* 2005) —see also (Cesa-Bianchi and Valentini 2010). H-loss is a measure of discrepancy between multilabels based on a simple intuition: *if a parent class has been predicted wrongly, then errors in its descendants should not be taken into account.* Given fixed *cost coefficients* $c_1, \ldots, c_m > 0$, the H-loss $\ell_H(\widehat{y}, v)$ between multilabels $\widehat{y}$ and v is computed as follows: all paths in the taxonomy T from the root down to each leaf are examined and, whenever a node $i \in \{1, \ldots, m\}$ 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.

In the evaluation phase, HBAYES predicts the Bayes-optimal multilabel $\widehat{y} \in \{0,1\}^m$ for a gene g based on the estimates $\widehat{p}_i(g)$ for $i = 1, \ldots, m$. By definition of Bayes-optimality, the optimal multilabel for g is the one that minimizes the loss when the true multilabel V is drawn from the joint distribution computed from the estimated conditionals $\widehat{p}_i(g)$. That is,

$$\widehat{y} = \operatorname*{argmin}_{y \in \{0,1\}^m} \mathbb{E}\big[\ell_H(y, V) \mid g\big] \ . \tag{4}$$

The calculation of the empirical performances reported in Section 4 has been performed using the uniform cost coefficients $c_i = 1$, for $i = 1, \ldots, m$. However, since with uniform coefficients the H-loss can be made small simply by predicting sparse multilabels (i.e., multilabels $\widehat{y}$ such that $\sum_i \widehat{y}_i$ is small), in the training phase we set the cost coefficients to $c_i = 1/|\text{root}(T)|$, if $i \in \text{root}(T)$, and to $c_i = c_j/|\text{child}(j)|$ with $j = \text{par}(i)$ otherwise. This normalizes the H-loss, in the sense that the maximal H-loss contribution of all nodes in a subtree excluding its root equals that of its root.

Let $\{A\}$ be the indicator function of event A. Given g and the estimates $\widehat{p}_i = \widehat{p}_i(g)$ for $i = 1, \ldots, m$, the HBAYES prediction (4) can be equivalently rewritten as follows —see (Cesa-Bianchi *et al.* 2005) for details.

HBAYES **prediction rule**

Initially, set the labels of each node i to

$$\widehat{y}_i = \operatorname*{argmin}_{y \in \{0,1\}} \Big(c_i \widehat{p}_i (1-y) + c_i (1-\widehat{p}_i) y + \widehat{p}_i \{y = 1\} \sum_{j \in \text{child}(i)} H_j(\widehat{y}) \Big) \tag{5}$$

where

$$H_j(\widehat{y}) = c_j \widehat{p}_j (1-\widehat{y}_j) + c_j (1-\widehat{p}_j) \widehat{y}_j + \widehat{p}_j \{\widehat{y}_j = 1\} \sum_{k \in \text{child}(j)} H_k(\widehat{y})$$

is recursively defined over the nodes j in the subtree rooted at i with each $\widehat{y}_j$ set according to (5).

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), $\widehat{y}$ can be computed for a given g via a simple bottom-up message-passing procedure whose only parameters are the estimates $\widehat{p}_i$. Unlike standard

top-down hierarchical methods —see Section 3.3, each $\widehat{y}_i$ also depends on the classification of its child nodes. In particular, if all child nodes k of i have $\widehat{p}_k$ close to a half, then the Bayes-optimal label of i tends to be 0 irrespective of the value of $\widehat{p}_i$. Vice versa, if i's children all have $\widehat{p}_k$ close to either 0 or 1, then the Bayes-optimal label of i is based on $\widehat{p}_i$ only, ignoring the children. The intuition behind this behavior is the following: the estimate $\widehat{p}_k$ is built based only on the examples on which the parent i of k is positive. Hence, a "neutral" estimate $\widehat{p}_k = \frac{1}{2}$ signals that the current instance is a negative example for the parent i.

3.5 Hierarchical True Path Rule ensembles

The True Path Rule (TPR) ensemble method (Valentini and Re 2009; Valentini 2011) is directly inspired by the *true path rule* that governs both GO and FunCat taxonomies. Citing the Gene Ontology Consortium (2010): "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". 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:

$$\begin{cases} y_i = 1 \Rightarrow y_{\text{par}(i)} = 1 \\ y_i = 0 \not\Rightarrow y_{\text{par}(i)} = 0. \end{cases} \tag{6}$$

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:

$$\begin{cases} y_i = 1 \not\Rightarrow y_{\text{child}(i)} = 1 \\ y_i = 0 \Rightarrow y_{\text{child}(i)} = 0. \end{cases} \tag{7}$$

From (6) and (7) we observe an asymmetry in the rules that govern the assignments of positive and negative labels. Indeed, we have a propagation of positive predictions from bottom to top of the hierarchy in (6), and a propagation of negative labels from top to bottom in (7). On the contrary negative labels cannot propagate from bottom to top, and positive predictions cannot propagate from top to bottom.

According to these rules, in TPR ensembles positive predictions for a node influence in a recursive way their ancestors, while negative predictions influence their offsprings. The ensemble embeds the functional relationships between functional classes that characterize the hierarchical taxonomy: in a first step base learners are independently trained to learn each specific class of the taxonomy. Then, their predictions are combined according to the true path rule. More precisely, the base classifiers estimate local probabilities $\overline{p}_i(g)$ that a given gene g belongs to class ω_i, and in a second step the ensemble provides an estimate $\overline{p}_i(g)$ of the "consensus" global probability $p_i(g)$. Let us consider the set $\phi_i(g)$ of the children of node i for which we have a positive prediction for a given gene g:

$$\phi_i(g) = \left\{ j : j \in \text{child}(i), \widehat{y}_j = 1 \right\}. \tag{8}$$

The global consensus probability $\overline{p}_i(g)$ of the ensemble depends both on the local prediction $\widehat{p}_i(g)$ and on the prediction of the nodes belonging to $\phi_i(g)$:

$$\overline{p}_i(g) = \frac{1}{1 + |\phi_i(g)|} \left(\widehat{p}_i(g) + \sum_{j \in \phi_i(g)} \overline{p}_j(g) \right). \tag{9}$$

The decision $\widehat{y}_i(g)$ at node/class i is set to 1 if $\overline{p}_i(g) > t$, and to 0 otherwise (a natural choice for t is 0.5). Note that the restriction to nodes belonging to $\phi_i(g)$ in the summation of (9) depends on the true path rule: indeed only children nodes for which we have a positive prediction can influence their parent. In the leaf nodes the sum disappears and (9) reduces to $\overline{p}_i(g) = \widehat{p}_i(g)$. On the contrary, if for a given node $\widehat{y}_i = 0$, then the algorithm propagates this decision to the corresponding subtree.

The high-level pseudo-code of the TPR ensemble algorithm to predict the hierarchical multilabel $\hat{y}$ for a generic unknown gene g is given in Fig. 1. To simplify the notation, $\overline{p}_i(g), \widehat{p}_i(g), \widehat{y}_i(g)$ are denoted, respectively, with $\overline{p}_i, \widehat{p}_i, \widehat{y}_i$, since in any case we refer to the same gene g whose labels $\widehat{y}$ need to be predicted.

Fig. 1: True Path Rule multilabel hierarchical algorithm

```
Input:
- tree T of the m hierarchical classes
- set of m classifiers (one for each node) each predicting p̂_i, i = 1,...,m
begin algorithm
01:    for each level k of the tree T from bottom to top do
02:        for each node i at level k do
03:           if i is a leaf
04:              p̄_i ← p̂_i
05:              if (p̄_i > t) ŷ_i ← 1
06:              else ŷ_i ← 0
07:           else
08:              φ_i ← {j | j ∈ child(i), ŷ_j = 1}
09:              p̄_i ← 1/(1+|φ_i|) (p̂_i + Σ_{j∈φ_i} p̄_j)
10:              if (p̄_i > t) ŷ_i ← 1
11:              else
12:                 ŷ_i ← 0
13:                 for each j ∈ subtree(i) do
14:                    ŷ_j ← 0
15:                    if (p̄_j > p̄_i) p̄_j ← p̄_i
16:                 end for
17:       end for
18:    end for
end algorithm.
Output: for each node i
- the ensemble decisions:   ŷ_i = { 1  if gene g belongs to node i
                                  { 0  otherwise
- the estimated probabilities p̄_i that gene g belongs to the node i ∈ T
```

The main external loop (rows $1-18$) performs a bottom-up traversal of the tree, thus assuring that all the offsprings of a given node i for which we have a positive prediction can influence its prediction (row 9). The internal loop (rows $2-17$) scans all the nodes at a given depth. Note that if a node is a leaf (row 3), then the consensus probability $\overline{p}_i$ is equal to the local probability $\widehat{p}_i$, while if a node is internal (rows $7-16$), the set ϕ_i of the "positive" children of i is determined (row 8) and then used to compute the consensus probability $\overline{p}_i$ according to (9). According to the true path rule, the algorithm sets the classes belonging to the subtree rooted at i to negative, when $\widehat{y}_i$ is set to 0 (rows 13-16). The algorithm provides both the multilabels $\widehat{y}_i$ and an estimate of the probabilities $\overline{p}_i$ that a given example g belongs to the class $i = 1,\ldots,m$.

3.6 Cost-sensitive methods

Functional classes are unbalanced, with negative examples typically outnumbering positives, and for this reason we need cost-sensitive techniques. Here we introduce cost-sensitive variants of HTD, HBAYES and TPR hierarchical ensemble methods, which are suitable for learning datasets whose multilabels are sparse (i.e., datasets whose classes are unbalanced). It is worth noting that all the cost-sensitive methods use the same estimates $\widehat{p}_i$ of the "a posteriori" probabilities: the only difference is in the way the cost-sensitive ensemble classifiers are defined in terms of these estimates.

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 in the subtree rooted at i) using the rule $\widehat{y}_i = \left\{\widehat{p}_i \geq \frac{1}{2}\right\} \times \{\widehat{y}_{\text{par}(i)} = 1\}$ for $i = 1, \ldots, m$ (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 $\widehat{y}_i = \{\widehat{p}_i \geq \tau\} \times \{\widehat{y}_{\text{par}(i)} = 1\}$. By tuning τ we may obtain ensembles with different precision/recall characteristics.

HBAYES-CS. The cost-sensitive variant of HBAYES, that we named HBAYES-CS, distinguishes the cost c_i^- of a false negative (FN) mistake from the cost c_i^+ of a false positive (FP) mistake. Using this distinction, (5) can be rewritten as

$$\widehat{y}_i = \operatorname*{argmin}_{y \in \{0,1\}} \left(c_i^- \widehat{p}_i (1-y) + c_i^+ (1-\widehat{p}_i) y + \widehat{p}_i \{y = 1\} \sum_{j \in \text{child}(i)} H_j(\widehat{y}) \right) \tag{10}$$

where the expression for $H_j(\widehat{y})$ gets changed correspondingly. We now parametrize the relative costs of FP and FN by introducing a factor $\alpha \geq 0$ such that $c_i^- = \alpha c_i^+$ while keeping $c_i^+ + c_i^- = 2c_i$. This allows to further rewrite (10) as

$$\widehat{y}_i = 1 \Longleftrightarrow \widehat{p}_i \left(2c_i - \sum_{j \in \text{child(i)}} H_j \right) \geq \frac{2c_i}{1+\alpha} \, . \tag{11}$$

It is easy to see that 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. Hence, by incrementing the cost factor, we could expect that the recall of the ensemble tends to increase, eventually at the expenses of the precision.

A global α parameter can be experimentally selected (e.g., by cross-validation on the training data), but considering that α represents a factor to balance the misclassification cost between positive and negative examples, we could also 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 n_i^- examples, estimated from the training data:

$$\alpha_i = \frac{n_i^-}{n_i^+} \Rightarrow c_i^+ = \frac{2}{\frac{n_i^-}{n_i^+} + 1} c_i = \frac{2n_i^+}{n_i^- + n_i^+} c_i \, . \tag{12}$$

The decision rule (11) at each node then becomes:

$$\widehat{y}_i = 1 \Longleftrightarrow p_i \left(2c_i - \sum_{j \in \text{child(i)}} H_j \right) \geq \frac{2c_i}{1+\alpha_i} = \frac{2c_i n_i^+}{n_i^- + n_i^+} \, . \tag{13}$$

TPR-W. In the TPR algorithm there is no way to explicitly balance the local prediction $\widehat{p}_i(x)$ at node i (9) with the positive predictions coming from the offsprings. 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, $0 \leq w \leq 1$, such that if $w = 1$ the decision at node i depends only by the local predictor, otherwise the prediction is shared proportionally to w and $1-w$ between respectively the local parent predictor and the set of its children:

$$\overline{p}_i = w\,\widehat{p}_i + \frac{1-w}{|\phi_i|}\sum_{j\in\phi_i}\overline{p}_j\ . \tag{14}$$

We thus obtain a variant of the *TPR* algorithm, that we name *weighted True Path Rule (TPR-w)* hierarchical ensemble algorithm by substituting rows 8 and 9 of the basic algorithm (Fig. 1) with the following pseudocode:

$\phi_i \leftarrow \{j | j \in \text{child}(i), \widehat{y}_j = 1\}$
`if` $(|\phi_i| > 0)$
  $\overline{p}_i \leftarrow w\,\widehat{p}_i + \frac{1-w}{|\phi_i|}\sum_{j\in\phi_i}\overline{p}_j$
`else`
  $\overline{p}_i(x) \leftarrow \widehat{p}_i$

By tuning the w parameter we can modulate the precision/recall characteristics of the resulting ensemble. In this sense, TPR-W can be considered a cost-sensitive version of the TPR ensemble. More precisely, for $w \to 0$ the weight of the parent local predictor is small, and the ensemble decision mainly depends on 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, $w \to 1$ corresponds to a higher weight of the parent predictor; then less weight is given to possible positive predictions of the children, and the decision depends mainly on the local/parent base classifier. In case of a negative decision all the subtree is set to zero, causing the precision to increase. Note that for $w \to 1$ the behaviour of TPR-W becomes similar to that of HTD.

3.7 Integration of hierarchical multilabel, data fusion, and cost-sensitive techniques

The hierarchical ensemble methods combine the probabilistic output of the classifiers associated to each node of the tree. Hence, by replacing the classifiers trained on single sources of data with classifiers trained on multiple sources of data, we immediately obtain an integration of hierarchical multilabel algorithms with data fusion techniques. The only requirement is that the base classifiers at each node provide an estimate $\widehat{p}_i(g)$ of $\mathbb{P}(V_i = 1 \mid V_{\text{par}(i)} = 1, g)$. For instance, we can supply as input to the hierarchical ensembles the $\widehat{p}_i$ estimated through ensembles of classifiers trained on multiple sources of data, or with SVMs trained on matrices obtained by summing kernel matrices specific for each data set. This is summarized with the following two-step strategy:

1. Train a set of classifiers that estimate $\mathbb{P}(V_i = 1 \mid V_{\text{par}(i)} = 1, g)$ for each node $i = 1, \ldots, m$ of the FunCat taxonomy. Each classifier is an ensemble of base learners, or a SVM trained with multiple sources of data by kernel fusion methods (see Section 3.2).

2. Combine the predictions at each node to obtain the multilabel predictions according to the hierarchical multilabels methods (both the basic and cost-sensitive variants) described in Sections 3.4, 3.5, and 3.6.

The resulting hierarchical multilabel predictions respect the "true path rule" and implement a local combination of multiple sources of biomolecular data at each node of the FunCat tree, while possibly using a cost-sensitive approach.

It is easy to see that the computational cost of the combination step of HTD, HBAYES and TPR is linear w.r.t. the number of classes included in the hierarchy.

4 Experimental set-up

4.1 Data

We integrated six different sources of yeast biomolecular data, previously used for single-source ontology-wide gene function prediction (Cesa-Bianchi and Valentini 2010).

The data sets include two types of protein domain data (PFAM BINARY and PFAM LOGE), gene expression measures (EXPR), predicted and experimentally supported protein-protein interaction data (STRING and BioGRID) and pairwise sequence similarity data (SEQ. SIM.).

PFAM BINARY data are coded as binary vectors representing the presence or absence of 4950 protein domains obtained from the *Pfam* (Protein families) database (Finn *et al.* 2008). An alternative enriched representation of the same data (PFAM LOGE) has been obtained by replacing the binary scoring with log E-values computed by the HMMER software toolkit (Eddy 1998). We merged the experiments of Spellman *et al.* (1998) (gene expression measures relative to 77 conditions) with the transcriptional responses of yeast to environmental stress (173 conditions) by Gasch *et al.* (2000) to obtain the gene expression (EXPR) data set. Protein-protein interaction (PPI) data (BioGRID) 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 and represent the presence or absence of protein-protein interactions. Other binary protein-protein interactions, representing interaction data from yeast two-hybrid assay, mass-spectrometry of purified complexes, correlated mRNA expression and genetic interactions, have been collected in the STRING data set (von Mering *et al.* 2002). Pairwise sequence similarity data (SEQ. SIM.) have been computed using Log-E values obtained by Smith and Waterman local pairwise alignments between all pairs of yeast sequences.

We considered only yeast genes common to all data sets, and 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 [2], and the classes with at least 20 positive examples, using the *HCgene* R package (Valentini and Cesa-Bianchi 2008). This selection process yielded 1901 yeast genes annotated to 168 FunCat classes distributed across 16 trees and 5 hierarchical levels. We added a "dummy" root node to obtain a tree from the overall FunCat forest (Fig. 2).

[2] Our experiments build on annotations coded in the funcat-2.1 scheme, and funcat-2.1_data_20070316 data, available from the MIPS web site (http://mips.gsf.de/projects/funcat).

4.2 Experimental tasks

In order to understand the potentially different impact of hierarchical strategies, data fusion and cost-sensitive methods on the *GFP* problem, we performed several experimental classification tasks at genome and ontology-wide level (i.e., we considered all genes and all the 168 classes of the hierarchically structured multilabel classification problem):

(a) Comparison of "single-source" and data fusion techniques (kernel fusion and weighted voting) using both FLAT and hierarchical methods (HTD, HBAYES and TPR);
(b) Assessment of the improvements achievable by: (i) multilabel hierarchical methods vs. flat methods; (ii) cost-sensitive vs cost-insensitive strategies; (iii) synergic enhancements due to the concurrent application of multilabel hierarchical methods, cost-sensitive, and data fusion techniques;
(c) Analysis of the precision-recall characteristics of the compared methods;
(d) Impact of the choice strategy for selecting negative examples.

As baseline method we adopted the annotation transfer method based on the best BLAST hit (Altschul *et al.* 1990) of each query protein against the database of the available yeast proteins.

Note that by FLAT ensembles we mean a set of base learners each one predicting a single functional class, without any combination of the predictions that takes into account the hierarchical structure of the classes. For both FLAT and hierarchical ensemble methods we used linear SVMs with probabilistic output (Lin *et al.* 2007) as base learners.

About task (d), we tested whether training base learners with different strategies for choosing negative examples may have an impact on the generalization capabilities of multilabel hierarchical methods. More precisely, in Section 5.1, 5.2 and 5.3 we adopted the following strategy to select negative examples for training:

Parent Only (PO) strategy. At each FunCat node the negatives are the genes that are not annotated at the corresponding class, but are annotated at the parent class/node.

Then in Section 5.4 the same whole-ontology tasks have been performed using a strategy that does not take into account the hierarchical structure of classes:

Basic (B) strategy. Negatives for a given class are simply examples not annotated for that class.

4.3 Performance assessment

Following the experimental set-up proposed by Lewis *et al.* (2006), we did not perform model selection to select the best values for the parameters of the SVM base learners: we simply set the regularization parameter C to 10. By performing model selection we could of course expect better results. However, our aim is not to achieve the best possible results, but rather to analyze the impact and the synergy of different learning strategies for the *GFP* problem.

In order to assess the generalization capabilities of the ensembles, we adopted "external" 5-fold cross validation techniques, while to select the threshold value τ for HTD-CS ensembles, the values of α and w parameters for respectively HBAYES-CS and TPR-W ensembles, we applied "internal" 3-fold cross-validation, using the F-score as evaluation criterion.

In the context of ontology-wide gene function prediction problems, where negative examples are usually a lot more than positives, accuracy is not a reliable measure to assess

the classification performance. For this reason we adopted the classical F-score to take into account the unbalance of FunCat classes.

In order to better capture the hierarchical and sparse nature of the gene function prediction problem, we also need specific measures that estimate how far a predicted structured annotation is from the correct one. For instance, correctly predicting a parent or ancestor annotation, while failing to predict the most specific available annotation should be "partially correct", in the sense that we can gain information about the more general functional characteristics of a gene, missing only its most specific functions. For the purpose of capturing these specificities of functional annotations, we should consider how much the entire path from the most specific up to the more general annotation is correctly predicted or not. To this end, we specialized to trees a hierarchical version of the F-measure (*hierarchical F-measure*) originally proposed for graph-structured classes by Verspoor *et al.* (2006).

More precisely, for a given gene or gene product g consider the subtree $G \subset T$ of the predicted classes and the subtree C of the correct classes associated to g. For a leaf $f \in G$ and $c \in C$, let be $\uparrow f$ and $\uparrow c$ the set of their ancestors that belong, respectively, to G and C. The hierarchical precision (HP) and hierarchical recall (HR) are defined as follows:

$$HP = \frac{1}{|\ell(G)|} \sum_{f \in \ell(G)} \frac{|C \cap \uparrow f|}{|\uparrow f|} \qquad \text{and} \qquad HR = \frac{1}{|\ell(C)|} \sum_{c \in \ell(C)} \frac{|\uparrow c \cap G|}{|\uparrow c|}$$

where $\ell(\cdot)$ is the set of leaves of a tree. The *hierarchical F-measure* (HF) is the harmonic mean of the hierarchical precision and recall. It is easy to verify that HP, HR and HF have values between 0 and 1. Note that these measures show how much each single example is correctly predicted w.r.t. the hierarchy of the classes. By averaging across examples we can obtain average HP, HR and HF.

A high average 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. The hierarchical F-measure expresses the correctness of the structured prediction of the functional classes, taking into account also partially correct paths in the overall hierarchical taxonomy, thus providing in a synthetic way the goodness of the structured hierarchical prediction.

As a final remark, we would like to outline that FunCat and GO ontologies can be trusted, since they represent the classification of known functions of genes according to the results of the scientific community at a given time, but at the same time they keep on evolving, due to the new knowledge coming from the ongoing new studies in functional genomics, where also the computational prediction of gene functions plays a central role. From this standpoint, false positive predictions provided by computational methods can change, for instance, in true positive predictions in future releases of both FunCat and GO ontologies.

5 Results and discussion

In this section we analyze and try to quantify the synergy between the different learning issues involved in *GFP*. In this context, by "synergy" we mean the improvement with respect to a given performance metric (e.g., the F-score) due to the concurrent effect of two learning strategies. In particular, we detect a synergy whenever the combined action of the two strategies causes the performance, under the considered metric, to be larger than the average of the performances of the two strategies in isolation.

5.1 Impact of data fusion on flat and hierarchical methods

As a baseline for our functional prediction experiments, we performed a sequence homology-based functional annotation transfer, using *blastp* (protein-protein BLAST) (Altschul *et al.* 1990). For each queried protein we sorted the collected hits according to the blast score normalized by the length of the alignment. Then we transferred to the query sequence the functional annotations of the best scoring hit found in the database of the considered set of proteins. The F-score averaged across all the considered functional terms is 0.2224. It is worth noting that the transfer of the entire set of known functional annotations from a protein to another introduces a bias in favour of BLAST because it prevents the introduction of hierarchical inconsistencies, since the set of transferred functional annotations are, by definition, hierarchically consistent w.r.t. the FunCat functional ontology.

Table 1: *Average per-class F-scores with* FLAT, HTD, HTD-CS, HB *(*HBAYES*),* HB-CS *(*HBAYES-CS*),* TPR *and* TPR-W *ensembles, using single sources and multi-source (data fusion) techniques.*

METHODS	FLAT	HTD	HTD-CS	HB	HB-CS	TPR	TPR-W
SINGLE-SOURCE							
BIOGRID	0.2643	0.3759	0.4160	0.3385	0.4183	0.3902	0.4367
STRING	0.2203	0.2677	0.3135	0.2138	0.3007	0.2801	0.3048
PFAM BINARY	0.1756	0.2003	0.2482	0.1468	0.2407	0.2532	0.2738
PFAM LOGE	0.2044	0.1567	0.2541	0.0997	0.2847	0.3005	0.3160
EXPR.	0.1884	0.2506	0.2889	0.2006	0.2781	0.2723	0.3053
SEQ. SIM.	0.1870	0.2532	0.2899	0.2017	0.2825	0.2742	0.3088
MULTI-SOURCE (DATA FUSION)							
KERNEL FUSION	0.3220	0.5401	0.5492	0.5181	0.5505	0.5034	0.5592
WEIGH. VOTING	0.2754	0.2792	0.3974	0.1491	0.3532	0.3987	0.4109

Table 2: *Wilcoxon signed-ranks test results to evaluate the statistical significance of the improvement of data fusion techniques w.r.t. single data sources achieved with cost-sensitive multilabel hierarchical methods (*HBAYES-CS, HTD-CS *and* TPR-W*). Results in boldface are in favour of ensembles using single data sources.*

HBAYES-CS						
	BIOGRID	STRING	PFAM BIN.	PFAM LOGE	EXPR.	SEQ. SIM.
KERNEL FUSION	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$
WEIGHTED VOTING	$\mathbf{2.3 \times 10^{-4}}$	5.6×10^{-07}	2.2×10^{-15}	6.3×10^{-6}	1.3×10^{-15}	3.8×10^{-13}
HTD-CS						
	BIOGRID	STRING	PFAM BIN.	PFAM LOGE	EXPR.	SEQ. SIM.
KERNEL FUSION	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$
WEIGHTED VOTING	$\mathbf{9.5 \times 10^{-2}}$	6.9×10^{-12}	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$
TPR-W						
	BIOGRID	STRING	PFAM BIN.	PFAM LOGE	EXPR.	SEQ. SIM.
KERNEL FUSION	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$
WEIGHTED VOTING	$\mathbf{9.8 \times 10^{-1}}$	3.2×10^{-15}	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$

Table 1 summarizes the results of the comparison including single-source and data integration approaches together with both flat and hierarchical ensembles. As a first observation, we note that the baseline BLAST-based method achieves results comparable with FLAT SVM ensembles, but worse than FLAT SVM with data fusion methods, and significantly

worse than hierarchical ensemble methods with and without data fusion and cost-sensitive techniques (Table 1).

Data fusion techniques improve average per class F-score across classes in FLAT ensembles (first column of Table 1), and significantly boost multilabel hierarchical methods (columns HTD, HTD-CS, HB, HB-CS, TPR, TPR-W of Table 1). Note that Kernel Fusion largely improves on results achieved with any "single-source" ensemble methods, while Weighted Voting results are sometimes worse than those of the best single-source (BIOGRID) when hierarchical ensemble methods are applied (with FLAT and TPR ensembles Weighted Voting improves on BIOGRID). These results seem to partially contradict previous ones published in Re and Valentini (2010c), but note that in that work only the most general classes at the first level of the FunCat hierarchy were classified, and no hierarchical methods were applied.

The improvements achieved by data integration techniques are statistically significant according to the Wilcoxon test (Table 2). With all cost-sensitive hierarchical ensembles, Kernel Fusion performances are significantly better than any single-source approach (*p-value* = 2.2×10^{-16}). This is true also for Weighted Voting except for the BIOGRID data, where results are in favour of the single-source data against all the cost-sensitive hierarchical ensembles, even if the difference is significant only for HBAYES-CS ensembles (*p-value* = 2.3×10^{-4}, Table 2).

Focusing on Kernel Fusion, Fig. 2 depicts the classes (black nodes) where Kernel Fusion achieves better results than the best single-source data set (BIOGRID). It is worth noting that the number of black nodes is significantly larger in HBAYES-CS ensembles (Fig. 2 b) and TPR-W (Fig. 2 c) w.r.t. FLAT methods (Fig. 2 a). Moreover, considering the average F-score across classes (Table 1), the relative improvement due to the application of Kernel Fusion w.r.t. the best results achieved with a single source (namely, BIOGRID), even if statistically relevant for FLAT (about 20%), is significantly larger for hierarchical ensemble methods (between 30% and 50%).

It is well known that hierarchical multilabel ensembles largely outperform FLAT approaches (Guan *et al.* 2008; Obozinski *et al.* 2008), but these results also reveal a synergy between hierarchical ensemble methods and data fusion techniques.

5.2 Analysis of the synergy between hierarchical multilabel methods, cost sensitive, and data fusion techniques

Table 3: *Wilcoxon signed-ranks test results (p-values) to evaluate the statistical significance of the improvement of cost-sensitive w.r.t. non cost-sensitive multilabel hierarchical methods. Data integration method: Kernel Fusion.*

	FLAT	HTD	HBAYES	TPR
HBAYES-CS ($\alpha = 2$)	$\simeq 0$	5.9×10^{-04}	1.1×10^{-14}	5.3×10^{-5}
HTD-CS ($\tau = 0.4$)	$\simeq 0$	2.9×10^{-03}	2.8×10^{-13}	8.8×10^{-4}
TPR-W ($w = 0.7$)	$\simeq 0$	9.8×10^{-11}	2.2×10^{-16}	2.8×10^{-9}

According to previous works (Valentini and Re 2009; Cesa-Bianchi and Valentini 2010), cost-sensitive approaches boost predictions of hierarchical methods when single-sources of data are used to train the base learners. These results are confirmed when cost-sensitive

methods (HBAYES-CS, HTD-CS and TPR-W) are integrated with data fusion techniques, showing a synergy between multilabel hierarchical, data fusion (in particular kernel fusion), and cost-sensitive approaches (Fig. 3).

The improvements of per-class F-scores achieved by HBAYES-CS, HTD-CS and TPR-W are statistically significant at 0.005 significance level (Wilcoxon test) w.r.t. their "vanilla" counterparts and FLAT methods (Table 3). No significant difference can be detected between HBAYES-CS, HTD-CS and TPR-W. These results show that the adoption of hierarchical strategies with embedded global cost-sensitive strategies is a key to improving *GFP* performances.

It is worth noting that other approaches for learning unbalanced classes, i.e., undersampling techniques or cost-sensitive SVMs (Morik *et al.* 1999), can be applied to predict gene functions. These local methods could in principle be combined with the global cost-sensitive approach of HTD-CS, HBAYES-CS and TPR-W to further improve prediction performances.

Per-level analysis of the F-score in HBAYES-CS, HTD-CS, and TPR-W ensembles shows a certain degradation of performance w.r.t. the depth of nodes (Fig. 4), but this degradation is significantly lower when data fusion is applied. Indeed, the per-level F-score achieved by HBAYES-CS and HTD-CS when a single source is used consistently decreases from the top to the bottom level, and it is halved at level 5 w.r.t. to the first level. On the other hand, in our experiments with Kernel Fusion the average F-score at level 2, 3 and 4 is comparable, and the decrement at level 5 w.r.t. level 1 is only about 15% (Fig. 5). Similar results are reported also with TPR-W ensembles.

In conclusion, the synergic effects of hierarchical multilabel ensembles, cost-sensitive, and data fusion techniques significantly improve the performance of *GFP*. Moreover, these enhancements allow to obtain better and more homogeneous results at each level of the hierarchy. This is of paramount importance, because more specific annotations are more informative, and can get more biological insights about the functions of genes.

5.3 Analysis of the precision/recall characteristics of hierarchical multilabel methods

Since functional classes are unbalanced, precision/recall analysis plays a central role in *GFP* problems, and often drives "in vitro" experiments that provide biological insights about specific functional genomics problems (Friedberg 2006).

While FLAT ensembles achieve the overall best average per-class recall, among hierarchical ensemble methods TPR obtains the best results in terms of average recall at the expenses of a certain decrement in average precision (see Table 4). Nevertheless, the average precision in hierarchical methods is twice (and in several cases more than twice) that of FLAT methods (see Table 5). Moreover, we can observe a synergy between hierarchical methods and data fusion. For instance, HBAYES-CS with a Kernel Fusion strategy increases precision from 0.275 to 0.770 w.r.t. FLAT methods trained with the best single source data (BIOGRID). Likewise, HTD with a WEIGTHED VOTING fusion strategy increases precision from 0.275 to 0.786 w.r.t. FLAT methods trained with BIOGRID (see Table 5). Note that the precision of FLAT methods is too low to be practically relevant. These results clearly show that FLAT methods are not suitable for such a complex multilabel classification task.

Note that HP and HR measures are not applicable to FLAT methods, since their predictions can be inconsistent with the class hierarchy. In any case, per-class average F-score and precision show that hierarchical ensembles significantly outperform FLAT methods (Table 1 and 5).

Table 4: *Average per-class recall with* FLAT, HTD, HTD-CS, HB *(*HBAYES*)*, HB-CS *(*HBAYES-CS*)*, TPR *and* TPR-W *ensembles, using the best single source (*BIOGRID*) and multi-source (data fusion) techniques.*

METHODS	FLAT	HTD	HTD-CS	HB	HB-CS	TPR	TPR-W
BIOGRID	0.6143	0.2963	0.3749	0.2506	0.3709	0.5323	0.3814
KERNEL FUSION	0.6839	0.4512	0.5130	0.4105	0.5039	0.6343	0.5126
WEIGH. VOTING	0.5366	0.1818	0.3058	0.0899	0.2568	0.4559	0.2726

Table 5: *Average per-class precision with* FLAT, HTD, HTD-CS, HB *(*HBAYES*)*, HB-CS *(*HBAYES-CS*)*, TPR *and* TPR-W *ensembles, using the best single source and multi-source (data fusion) techniques.*

METHODS	FLAT	HTD	HTD-CS	HB	HB-CS	TPR	TPR-W
BIOGRID	0.2751	0.6012	0.5084	0.6348	0.5364	0.3717	0.5460
KERNEL FUSION	0.3112	0.7270	0.6263	0.7700	0.6476	0.4802	0.6555
WEIGH. VOTING	0.4484	0.7863	0.7043	0.7081	0.7272	0.5799	0.7472

Considering hierarchical recall, TPR achieves the best results (Fig. 3). We believe that this is possibly due to the bottom-up propagation of positive predictions in TPR (Sect. 3.5): sensitivity (recall) is improved, but at the expenses of a certain decay of hierarchical precision (Fig. 3).

HTD ensembles show the best hierarchical precision except with Weighted Voting, where hierarchical cost-sensitive methods perform better (Fig. 3). Here the propagation of negative predictions from top to bottom ensures that only "safe" positive predictions (according to the hierarchical structure of the classes) are maintained. Hierarchical cost-sensitive ensembles, which address the unbalance between positive and negative examples, show quite comparable results in terms of precision and a recall significantly higher than HTD. As a result, HBAYES-CS, HTD-CS, and TPR-W achieve good "intermediate" results for both precision and recall, leading to the best results in terms of the hierarchical F-score (Fig. 3).

Note also that while HTD-CS uses a top-down strategy, HBAYES-CS and TPR-W work bottom-up. Moreover, while HBAYES-CS is theoretically well-founded (Sect. 3.4), HTD-CS and TPR-W (Sect. 3.5) are heuristic methods. Despite these differences, there is no significant discrepancy between their overall performance in terms of average per-class F-score and hierarchical F-score. We believe that these results can be explained considering that the key to improve prediction performance in this task is not the choice of a specific hierarchical multilabel method, but rather the synergy between hierarchical multilabel, data fusion and cost-sensitive strategies.

A more refined analysis in terms of precision/recall and per-level results reveals differences between methods that are relevant to this specific application context. For instance, while the overall hierarchical precision and recall between HBAYES-CS and HTD-CS is quite similar, TPR-W achieves a slightly higher recall and a slightly lower precision (Fig. 3). These results can be explained through the bottom-up propagation of positive predictions that characterizes both TPR and TPR-W, as outlined above in this section.

The scenario is different if we analyze the average precision across levels of the FunCat taxonomy. Indeed, precision of HBAYES-CS and TPR-W at lower levels is higher than that of HTD-CS (Fig. 4). Fig. 6 shows that the black nodes representing FunCat classes for which HBAYES-CS and TPR-W improves precision on HTD-CS are concentrated on the middle and lower levels of the hierarchy. 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 and TPR-W is the fact that their precision/recall characteristics can be tuned via a single global parameter. In HBAYES-CS, by incrementing the cost factor $\alpha = c_i^- / c_i^+$ we introduce progressively lower costs for positive predictions, thus resulting in an increment of the recall (at the expenses of a possibly lower precision), In TPR-W, by incrementing w we reduce the recall and enhance the precision (Fig. 7).

As for HBAYES-CS, observe that by setting the α parameter at each node to the ratio of negative to positive examples for the corresponding class (Sect. 3.6), we attain results comparable to those obtained by internal cross-validation of the global α parameter, thus avoiding the corresponding computational overhead (results not shown).

5.4 Impact of the choice of different strategies for selecting negatives

In both GO and FunCat negative annotations are not typically available [3]. Moreover, some seminal works in functional genomics pointed out that the strategy of choosing negative training examples does affect the classifier performance (Ben-Hur and Noble 2006; Lewis *et al.* 2006).

In our experiments we used a strategy according to which negative examples for a class must be annotated for the parent class (*Parent Only* or *PO* strategy). More precisely, for a given class ω_i corresponding to node i in the taxonomy, the set of negative examples is $N_i = \{g : g \notin \omega_i, g \in \text{par}(i)\}$. Hence, this strategy selects negative examples for training that are in a certain sense "close" to positives.

To check whether an alternative strategy could significantly influence the performance of flat and hierarchical methods, we repeated the same whole-ontology and genome-wide experiments performed in the previous section, comprising the tuning of w, α and τ parameters, but this time choosing the set of negative examples simply as those genes g that are not annotated for class ω_i (*Basic* or *B* strategy), that is $N_i' = \{g : g \notin \omega_i\}$. It is easy to see that $N_i \subseteq N_i'$, hence this strategy selects for training a large set of generic negative examples, possibly annotated with classes that are associated with faraway nodes in the taxonomy. Of course, the set of positive examples is the same for both strategies.

If we compare results about average per-class F-score obtained with the *B* strategy (Table 6) to those obtained with the *PO* strategy (Table 1), we observe that the *B* strategy worsens the performance of hierarchical multilabel methods, while for FLAT ensembles there is no clear trend. This is more apparent in Fig. 8, comparing the F-scores obtained with *B* to those obtained with *PO*, using both hierarchical cost-sensitive (Fig. 8 (a)) and FLAT (Fig. 8 (b)) methods. Each point represents the F-score for a specific FunCat class achieved by a specific method with *B* (abscissa) and *PO* (ordinate) strategy for the selection of negative examples. For each method we have 168 points corresponding to the 168 different FunCat classes considered in the experiments. In Figure 8 (a) most points lie above the bisector independently of the hierarchical cost-sensitive method being used. This shows that hierarchical methods gain in performance when using the *PO* strategy as opposed to the *B* strategy (p-value = 2.2×10^{-16} according to the Wilcoxon signed-ranks test). This is not the case for FLAT methods (Fig. 8 (b)).

These results can be explained by considering that the *PO* strategy takes into account the hierarchy to select negatives, while the *B* strategy does not. More precisely, the *PO*

[3] More precisely, for some functional classes in both GO and FunCat we have a few negative annotations, but not so many to be practically relevant.

Table 6: *Average per-class F scores with* FLAT, HTD, HTD-CS, HB *(*HBAYES*) and* HB-CS *(*HBAYES-CS*),* TPR, TPR-W, *and* TPR-W-T *ensembles, using single sources and multi-source (data fusion) techniques and the* Basic *strategy to select negatives.*

METHODS	FLAT	HTD	HTD-CS	HB	HB-CS	TPR	TPR-W	TPR-W-T
SINGLE-SOURCE								
BIOGRID	0.2714	0.3264	0.3601	0.3301	0.3102	0.2977	0.3230	0.3609
STRING	0.2490	0.2735	0.2604	0.1349	0.2270	0.2777	0.2811	0.2570
PFAM BINARY	0.1677	0.2013	0.2198	0.1660	0.1933	0.1983	0.1963	0.2245
PFAM LOGE	0.2699	0.3245	0.2767	0.1584	0.2941	0.2979	0.3252	0.3343
EXPR.	0.1782	0.2103	0.2430	0.2074	0.2045	0.1906	0.2074	0.2437
SEQ. SIM.	0.1775	0.2107	0.2410	0.1999	0.2050	0.1897	0.2072	0.2409
MULTI-SOURCE (DATA FUSION)								
KERNEL FUSION	0.2940	0.3603	0.4089	0.3917	0.3431	0.3243	0.3568	0.4065
WEIGH. VOTING	0.3058	0.3572	0.4104	0.1266	0.3367	0.3365	0.3560	0.4240

strategy trains base classifiers to distinguish local differences (i.e., examples that are negative for a class and positive for the parent class), and hierarchical methods, which know the taxonomy, can use the information coming from other base classifiers to prevent a local base learner from incorrectly classifying "distant" negative examples. On the contrary, FLAT methods have no information about the hierarchical structure of classes and cannot correct local predictions, thus suffering from significantly higher false positive rates.

It is worth noting that even if we observe a degradation of performance in hierarchical methods with the *B* strategy, their results are still better than FLAT, and a synergy between hierarchical, cost-sensitive and data fusion approaches can be always observed (Table 6 and Fig. 9).

Looking at the behaviour of hierarchical cost-sensitive methods trained with *B* strategy, we noted that the best results of TPR-W have been obtained with relatively large values of w ($w > 0.7$, but sometimes also with $w = 0.9$). In these conditions TPR-W tends to become similar to HTD (apart from the bottom-up strategy), since decisions at each node mainly depend on the local predictor associated to that node. Hence, observing that HTD-CS performs significantly better than HTD (Table 6), we introduced a thresholded version of TPR-W, that we named TPR-W-T (T stands for threshold). Analogously to HTD-CS, we optimized by cross validation the best global threshold t applied to predict the class according to the rule $\widehat{p}_i > t \iff \widehat{y}_i = 1$ (Fig. 1). Results in the last column of Table 6 and in Fig. 9 show that TPR-W-T significantly improves on TPR-W, achieving the best results among hierarchical cost-sensitive methods when the *B* strategy is applied.

Regarding HBAYES-CS, its performance is slightly lower than the other cost-sensitive hierarchical methods (Table 6 and Fig. 9) when using the *B* strategy for selecting negatives, while with the *PO* strategy no significant differences can be detected between HBAYES-CS and the other cost-sensitive hierarchical methods. These results are not surprising, since the probabilistic model underlying HBAYES assumes that data are distributed according to the *PO* strategy, while the other methods make no explicit assumptions, even if they take advantage of this selection strategy.

The per-level precision/recall analysis in cost-sensitive hierarchical ensembles show that the *Basic* strategy introduces a significant decrement of the F-score, and in particular of the precision (Fig. 10), as we move down in the levels of the FunCat hierarchy. With the *PO* strategy (Fig. 4), on the contrary, precision is reasonably sustained across levels (e.g., we can observe a 17% reduction using *PO*, but a 68% reduction, moving down from level 1 to level 5 and using the *Basic* strategy with TPR-W ensembles combined with kernel fusion data

integration). We need high precision, especially at the lower levels of the hierarchy, since they correspond to the most specific and hence most informative classes from a functional genomics standpoint. These results confirm that the correct choice of the strategy to select negative examples for training is as important as the choice of the correct methods, and that with hierarchical methods *PO* significantly improves on the *Basic* strategy.

6 Conclusions

In this work we investigated the relationships between different learning strategies involved in *GFP*, a challenging multi-label classification problem characterized by constraints and dependencies between labels, unbalance of classes, and by the availability of multiple sources of data.

Our analysis shows and quantifies the synergy among heterogeneous data integration, hierarchical multi-label, and cost-sensitive approaches. This synergy is the key to drive biomolecular experiments aimed at discovering previously unannotated gene functions.

In particular, the main findings of our work can be summarized as follows:

- *There does exist a synergy between data integration and hierarchical multi-label methods*. Confirming previous results, data integration improves upon single-source approaches, and hierarchical ensembles enhance multi-label FLAT methods. Nevertheless, the combination of data integration and multi-label hierarchical methods achieves a significant performance increment over both hierarchical and data fusion techniques alone, confirming a synergy between them.
- *There does exist a synergy between hierarchical multi-label and cost-sensitive approaches*. According to previous works, cost-sensitive approaches boost predictions of hierarchical methods when individual data sources are used to train the base learners. With or without data fusion, hierarchical methods that take into account the unbalance between classes significantly improve their "vanilla" counterparts, and multi-view approaches yield further enhancements.
- *The combination of different learning strategies is more effective than the choice of a specific learning method*. Despite the fact that HBAYES-CS is theoretically well founded, while HTD-CS and TPR-W are heuristic methods, there is no significant difference between their overall results (in terms of average per-class F-score and hierarchical F-score). The key to improve prediction performance is not the choice of a specific hierarchical multi-label method, but the synergy between hierarchical multi-label, data fusion, and cost-sensitive strategies.
- *Synergic effects spread out across the levels of the hierarchy*. The performance decrease exhibited by HBAYES-CS, HTD-CS, and TPR-W as we move down the levels of the hierarchy is significantly reduced when data fusion is applied, thus resulting in better and more homogeneous results at each level of the hierarchy.
- FLAT *methods should not be applied to GFP*. The overall F-score achieved by hierarchical multi-label methods is always significantly higher than FLAT methods. In particular, the precision of FLAT methods is too low to be useful in practice, especially with lower level classes. As a consequence, such methods should not be applied to this task.
- *Combining different learning strategies preserves precision across the levels of the hierarchy*. If we combine hierarchical multi-label learning strategies, data fusion and cost-sensitive techniques, the decrease in precision at the low-level classes of the hierarchy is significantly limited. This is of paramount importance when we need to reduce the

costs of the biological validation of new gene functions discovered through computational methods. This synergy is clear in HBAYES-CS and TPR-W, while in HTD-CS we observe a less pronounced preservation of the precision across the levels of the hierarchy. Another advantage of HBAYES-CS and TPR-W is the possibility of tuning their precision/recall characteristics through a single global parameter.

– *The strategy of choosing negative examples influences performance*. The *Parent Only* (*PO*) strategy to select negative examples in the training phase significantly improves the performance of hierarchical multi-label methods, while the choice of the *PO* or *Basic* seems to be not so influent when using FLAT methods.

Summarizing, our analysis suggests that multi-label methods for *GFP* should combine: (a) hierarchical strategies to take into account the relationships between classes; (b) data integration approaches to capture different functional characteristics of genes; (c) cost-sensitive methods to address the unbalance between positive and negative examples for each functional class.

According to these findings, we proposed a general methodology to integrate hierarchical multi-label algorithms, data fusion, and cost-sensitive methods, that could be applied to design new integrated approaches to the *GFP* problem.

The strategy of choosing negative examples for training also seems to play a central role to improve the performance of *GFP* methods. Nevertheless, we need new theoretical and experimental studies to investigate the impact of this issue on *GFP*.

Other important issues listed in the introduction of this paper are left for future investigations. A possible research topic regards methods sensitive to the reliability of labels. They could address the different evidence of association between genes and functional classes, and their synergy with other learning issues involved in the *GFP* problem.

In conclusion, we believe that the analysis of the relationships and the quantification of the synergy between these different items is the key to design new algorithms for combining multiple learning strategies, and to solve a multilabel problem of great importance in molecular biology.

Acknowledgements We would like to thank the anonymous reviewers for their comments and suggestions. 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

Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, **215**.

Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, **25**(17), 3389–3402.

Amit, Y., Dekel, O., and Singer, Y. (2007). A boosting algorithm for label covering in multilabel problems. *Journal of Machine Learning Research, W&C Proceedings*, **2**, 27–34.

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

Bakir, G., Hoffman, T., Scholkopf, B., Smola, A.J. andTaskar, B., and Vishwanathan, S. (2007). *Predicting structured data*. MIT Press, Cambridge, MA.

Barutcuoglu, Z., Schapire, R., and Troyanskaya, O. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, **22**(7), 830–836.

Ben-Hur, A. and Noble, W. (2006). Choosing negative examples for the prediction of protein-protein interactions. *BMC Bioinformatics*, **7**(Suppl 1/S2).

Bengio, Y., Delalleau, O., and Le Roux, N. (2006). Label Propagation and Quadratic Criterion. In O. Chapelle, B. Scholkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press.

Blockeel, H., Bruynooghe, M., Dzeroski, S., Ramon, J., and Struyf, J. (1998). Top-down induction of clustering trees. In *Proc. of the 15th Int. Conf. on Machine Learning*, pages 55–63.

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

Cai, L. and Hofmann, T. (2004). Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 78–87, New York, NY, USA.

Cesa-Bianchi, N. and Valentini, G. (2010). Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. *Journal of Machine Learning Research, W&C Proceedings, Machine Learning in Systems Biology*, **8**, 14–29.

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

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

Cesa-Bianchi, N., Re, M., and Valentini, G. (2010a). Functional inference in FunCat through the combination of hierarchical ensembles with data fusion methods. In *ICML-MLD 2nd International Workshop on learning from Multi-Label Data*, pages 13–20, Haifa, Israel.

Cesa-Bianchi, N., Gentile, C., Vitale, F., and Zappella, G. (2010b). Random spanning trees and the prediction of weighted graphs. In *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel.

Chua, H., Sung, W., and Wong, L. (2007). An efficient strategy for extensive integration of diverse biological data for protein function prediction. *Bioinformatics*, **23**(24), 3364–3373.

Clare, A. and King, R. (2003). Predicting gene function in *saccharomices cerevisiae*. *Bioinformatics*, **19**(Supp.2), II42–II49.

Dembczynski, K., Cheng, W., and Hullermeier, E. (2010a). Bayes optimal multilabel classification via probabilistic classifier chains. In *Proc. of ICML 2010*, pages 1–10.

Dembczynski, K., Waegeman, W., Cheng, W., and Hullermeier, E. (2010b). On label dependence in multi-label classification. In *ICML-MLD: 2nd International Workshop on learning from Multi-Label Data*, pages 5–12, Haifa, Israel.

Deng, M., Chen, T., and Sun, F. (2004). An integrated probabilistic model for functional prediction of proteins. *J. Comput. Biol.*, **11**, 463–475.

desJardins, M., Karp, P., Krummenacker, M., Lee, T., and Ouzounis, C. (1997). Prediction of enzyme classification from protein sequence without the use of sequence similarity. In *Proc. of the 5th ISMB*, pages 92–99. AAAI Press.

Dimou, A., Tsoumakas, G., Mezaris, V., Kompatsiaris, I., and Vlahavas, I. (2009). An empirical study of multi-label methods for video annotation. In *Proc. 7th International Workshop on Content-Based Multimedia Indexing, CBMI 09*, Chania, Greece.

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

Eisner, R., Poulin, B., Szafron, D., and Lu, P. (2005). Improving protein prediction using the hierarchical structure of the Gene Oontology. In *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*.

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

Friedberg, I. (2006). Automated protein function prediction-the genomic challenge. *Brief. Bioinformatics*, **7**, 225–242.

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

Gene Ontology Consortium (2010). True path rule. http://www.geneontology.org/GO.usage.shtml#truePathRule.

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

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

Juncker, A., Jensen, L., Perleoni, A., Bernsel, A., Tress, M., Bork, P., von Heijne, G., Valencia, A., Ouzounis, A., Casadio, R., and Brunak, S. (2009). Sequence-based feature prediction and annotation of proteins. *Genome Biology*, **10:206**.

Karaoz, U. *et al.* (2004). Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc. Natl Acad. Sci. USA*, **101**, 2888–2893.

Kittler, J., Hatef, M., Duin, R., and Matas, J. (1998). On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **20**(3), 226–239.

Kuncheva, L., Bezdek, J., and Duin, R. (2001). Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, **34**(2), 299–314.

Lampert, C. and Blaschko, M. (2009). Structured prediction by joint kerenl support estimation. *Machine Learning*, **77**, 249–269.

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

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

Lewis, D., Jebara, T., and Noble, W. (2006). Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. *Bioinformatics*, **22**(22), 2753–2760.

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

Loewenstein, Y., Raimondo, D., Redfern, O., Watson, J., Frishman, D., Linial, M., Orengo, C., Thornton, J., and Tramontano, A. (2009). Protein function annotation by homology-based inference. *Genome Biology*, **10:207**.

Marcotte, E., Pellegrini, M., Thompson, M., Yeates, T., and Eisenberg, D. (1999). A combined algorithm for genome-wide prediction of protein function. *Nature*, **402**, 83–86.

McDermott, J.and Bumgarner, R. and Samudrala, R. (2005). Functional annotation from predicted protein interaction networks. *Bioinformatics*, **21**(15), 3217–3226.

Morik, K., Brockhausen, P., and Joachims, T. (1999). Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proceedings of 16th International Conference on Machine Learning (ICML)*, Bled (Slovenia). Morgan Kaufmann.

Mostafavi, S. and Morris, Q. (2009). Using the gene ontology hierarchy when predicting gene function. In *Proceedings of the Twenty-Fifth Annual Conference on Uncertainty in Artificial Intelligence (UAI-09)*, pages 419–427, Corvallis, Oregon. AUAI Press.

Mostafavi, S. and Morris, Q. (2010). Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics*, **26**(14), 1759–1765.

Mostafavi, S., Ray, D., Warde-Farley, D., Grouios, C., and Morris, Q. (2008). GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, **9**(S4).

Myers, C. and Troyanskaya, O. (2007). Context-sensitive data inegration and prediction of biological networks. *Bioinformatics*, **23**, 2322–2330.

Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., and Singh, M. (2005). Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, **21**(S1), 302–310.

Noble, W. and Ben-Hur, A. (2007). Integrating information for protein function prediction. In T. Lengauer, editor, *Bioinformatics - From Genomes to Therapies*, volume 3, pages 1297–1314. Wiley-VCH.

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

Oliver, S. (2000). Guilt-by-association goes global. *Nature*, **403**, 601–603.

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

Prlic, A., Down, T., Kulesha, E., Finn, R., Kahari, A., and Hubbard, T. (2007). Integrating sequence and structural biology with DAS. *BMC Bioinformatics*, **8**(233).

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, (1), 81–106.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2007). More efficiency in multiple kernel learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 775–782, New York, NY, USA. ACM.

Re, M. and Valentini, G. (2010a). Integration of heterogeneous data sources for gene function prediction using Decision Templates and ensembles of learning machines. *Neurocomputing*, **73**(7-9), 1533–37.

Re, M. and Valentini, G. (2010b). Noise tolerance of Multiple Classifier Systems in data integration-based gene function prediction. *Journal of Integrative Bioinformatics*, **7**(3:139).

Re, M. and Valentini, G. (2010c). 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, Machine Learning in Systems Biology*, **8**, 98–111.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. (2006). Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, **7**, 1601–1626.

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

Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, MA.

Schietgat, L., Vens, C., Struyf, J., Blockeel, H., and Dzeroski, S. (2010). Predicting gene function using hierarchical multilabel decision tree ensembles. *BMC Bioinformatics*, **11**(2).

Shahbaba, B. and Neal, M. (2006). Gene function classification using Bayesian models with hierarchy-based priors. *BMC Bioinformatics*, **7**(448).

Sokolov, A. and Ben-Hur, A. (2010). Hierarchical classification of Gene Ontology terms using the GOstruct method. *Journal of Bioinformatics and Computational Biology*, **8**(2), 357–376.

Sonnenburg, S., Ratsch, G., Schafer, C., and Scholkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, **7**, 1531–1565.

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

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

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

Trohidis, K., Tsoumahas, G., Kalliris, G., and Vlahavas, I. (2008). Multilabel classification of music into emotions. In *Proc. of the 9th International Conference on Music Information Retrieval*, pages 325–330.

Troyanskaya, O. *et al.* (2003). A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *saccharomices cerevisiae*). *Proc. Natl Acad. Sci. USA*, **100**, 8348–8353.

Tsochantaridis, I., Joachims, T., Hoffman, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, **6**, 1453–1484.

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

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). Random k-Labelsets for Multi-Label Classification. *IEEE Transactions on Knowledge and Data Engineering*.

Tsuda, K., Shin, H., and Scholkopf, B. (2005). Fast protein classification with multiple networks. *Bioinformatics*, **21**(Suppl 2), ii59–ii65.

Valentini, G. (2011). True Path Rule hierarchical ensembles for genome-wide gene function prediction. *IEEE ACM Transactions on Computational Biology and Bioinformatics*, **8**(3), 832–847.

Valentini, G. and Cesa-Bianchi, N. (2008). Hcgene: a software tool to support the hierarchical classification of genes. *Bioinformatics*, **24**(5), 729–731.

Valentini, G. and Re, M. (2009). Weighted True Path Rule: a multilabel hierarchical algorithm for gene function prediction. In *MLD-ECML 2009, 1st International Workshop on learning from Multi-Label Data*, pages 133–146, Bled, Slovenia.

Vazquez, A., Flammini, A., Maritan, A., and Vespignani, A. (2003). Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, **21**, 697–700.

Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., and Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, **73**(2), 185–214.

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

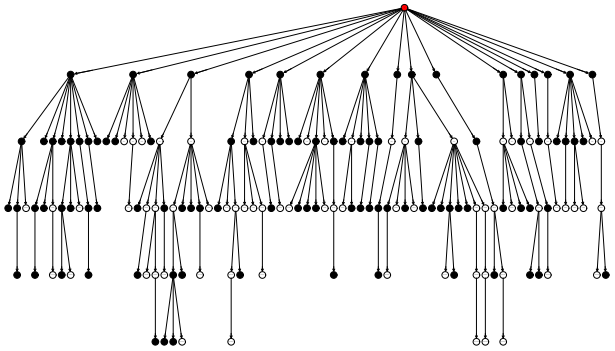
von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S., Fields, S., and Bork, P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403.

Xiong, J. *et al.* (2006). Genome wide prediction of gene function via a generic knowledge discovery approach based on evidence integration. *BMC Bioinformatics*, **7**(268).

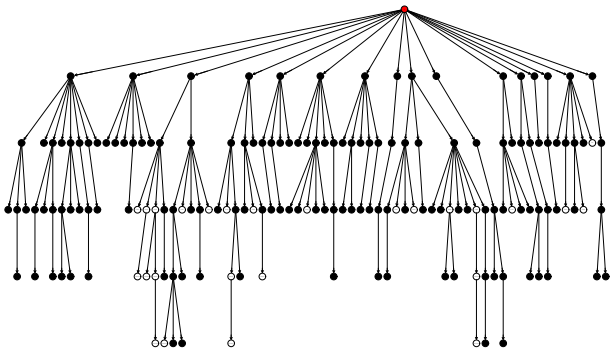
Zhang, M. and Zhou, Z. (2006). Multi-label neural network with applications to functional genomics and text categorization. *IEEE Trans. on Knowledge and Data Engineering*, **18**(10), 1338–1351.

Zhang, M. and Zhou, Z. (2007). ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, **40**(7), 2038–2048.

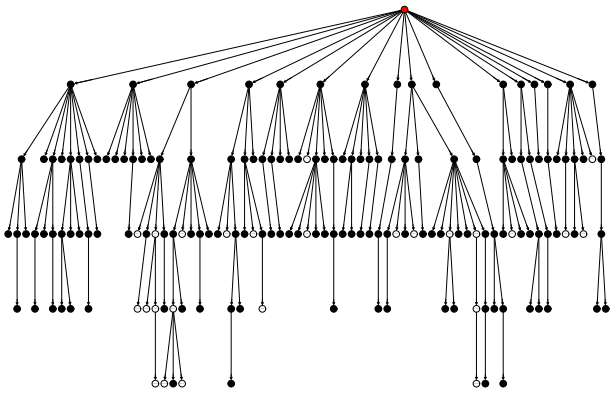
Zhang, M., Tsoumakas, G., and Zhou, Z. (2010). *2nd International Workshop on Learning from Multi-Label Data (MLD'10) - Working notes*. Haifa, Israel.



(a)



(b)



(b)

Fig. 2: *FunCat trees to compare F-scores achieved with data integration (*KF*) to the best single-source classifiers trained on* BIOGRID *data. Black nodes depict functional classes for which* KF *achieves better F-scores. (a)* FLAT*, (b)* HBAYES-CS*, (c)* TPR-W *ensembles.*

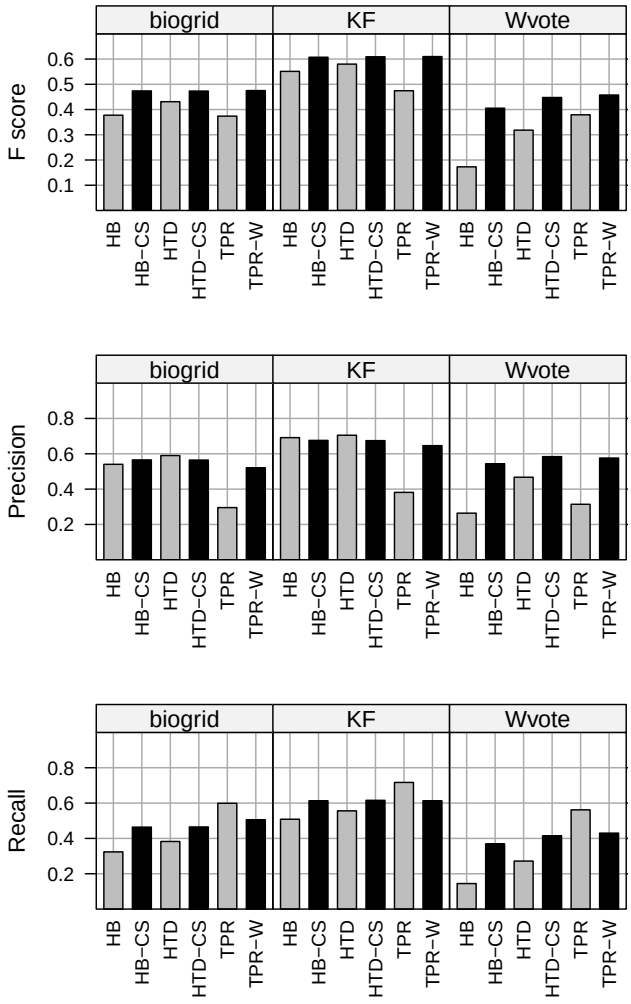


Fig. 3: *Comparison of hierarchical F-score, precision, and recall among different ensemble methods using the best source of biomolecular data (*BIOGRID*), Kernel Fusion (*KF*), and Weighted Voting (*WVOTE*) data integration techniques.* HB *stands for* HBAYES.

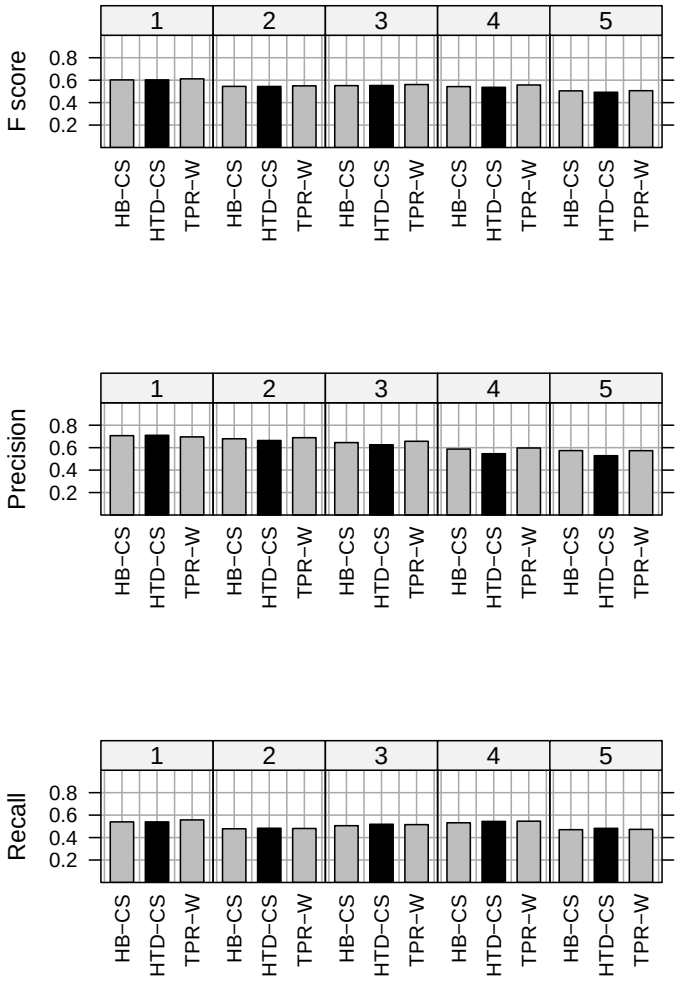


Fig. 4: *Per level average F-score, precision and recall across the five levels of the FunCat taxonomy in* HBAYES-CS, HTD-CS *and* TPR-W *ensembles using Kernel Fusion data integration. Number 1 to 5 refer to levels: level 1 is the top level, level 5 the bottom.*

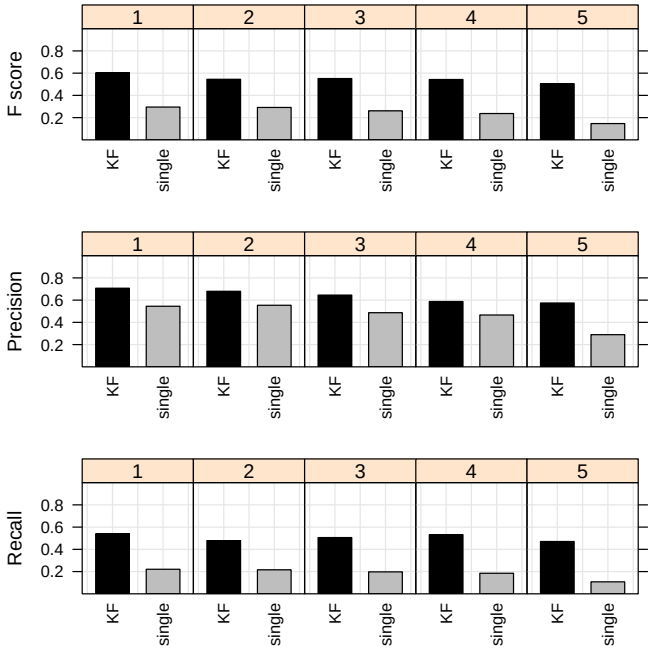
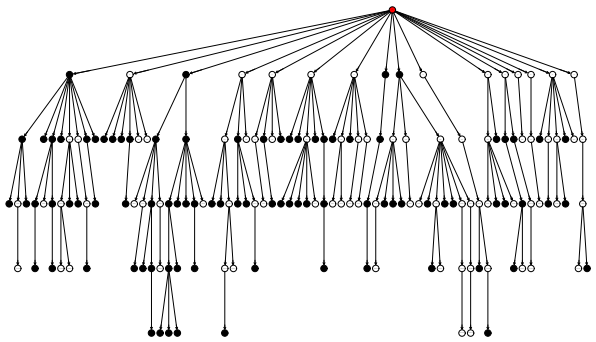
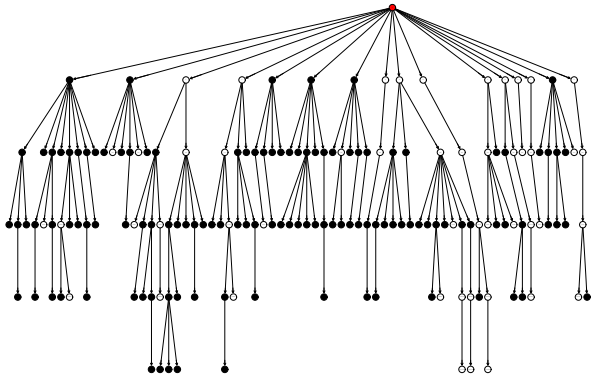


Fig. 5: *Comparison of per level average F-score, precision and recall across the five levels of the FunCat taxonomy in* HBAYES-CS *using single data sets (single) and kernel fusion techniques (KF). Performance of "single" are computed by averaging across all the single data sources.*

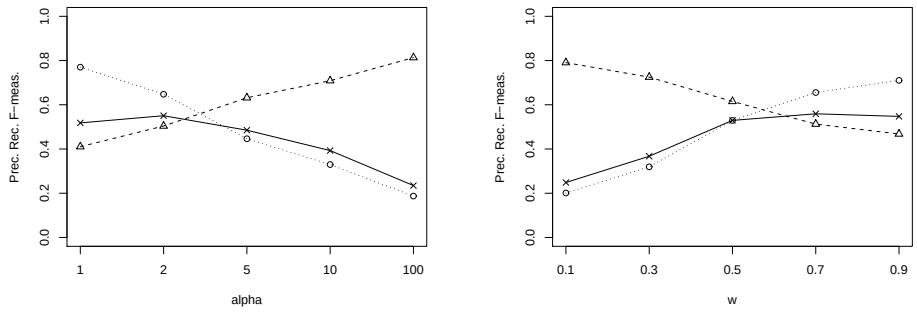


(a)



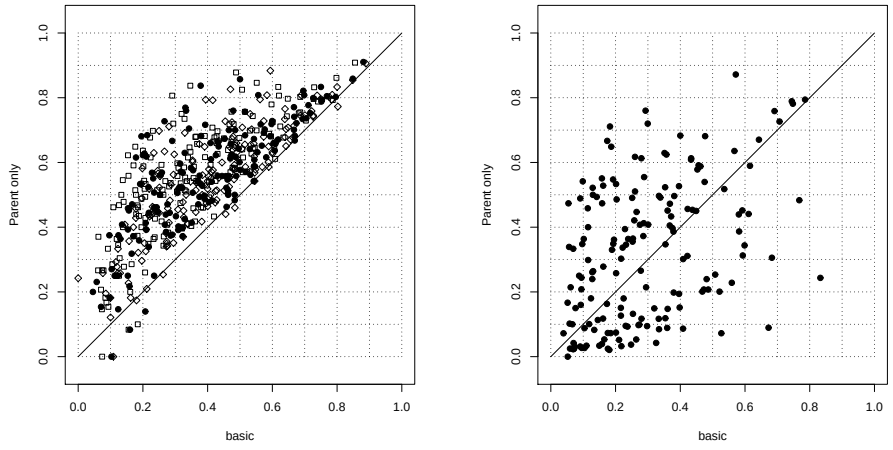
(b)

Fig. 6: *Ontology-wide FunCat tree plot highlighting the nodes at which the precision of the cost-sensitive hierarchical methods* HBAYES-CS *and* TPR-W *is larger than the one obtained by* HTD-CS *using Kernel Fusion to integrate multiple sources of data. (a)* HBAYES-CS *vs.* HTD-CS*; (b)* TPR-W *vs.* HTD-CS



(a) (b)

Fig. 7: *Hierarchical F-score, precision and recall as functions of global cost sensitive parameters. (a)* HBAYES-CS*, (b)* TPR-W



(a) (b)

Fig. 8: *Comparison of average per-class F-score between* Basic *and* PO *strategies. (a) Hierarchical cost-sensitive strategies:* HTD-CS *(squares),* TPR-W *(triangles),* HBAYES-CS *(filled circles). (b)* FLAT*. Abscissa: per-class F-score with base learners trained according to the* Basic *strategy; ordinate: per-class F-score with base learners trained according to the* PO *strategy.*

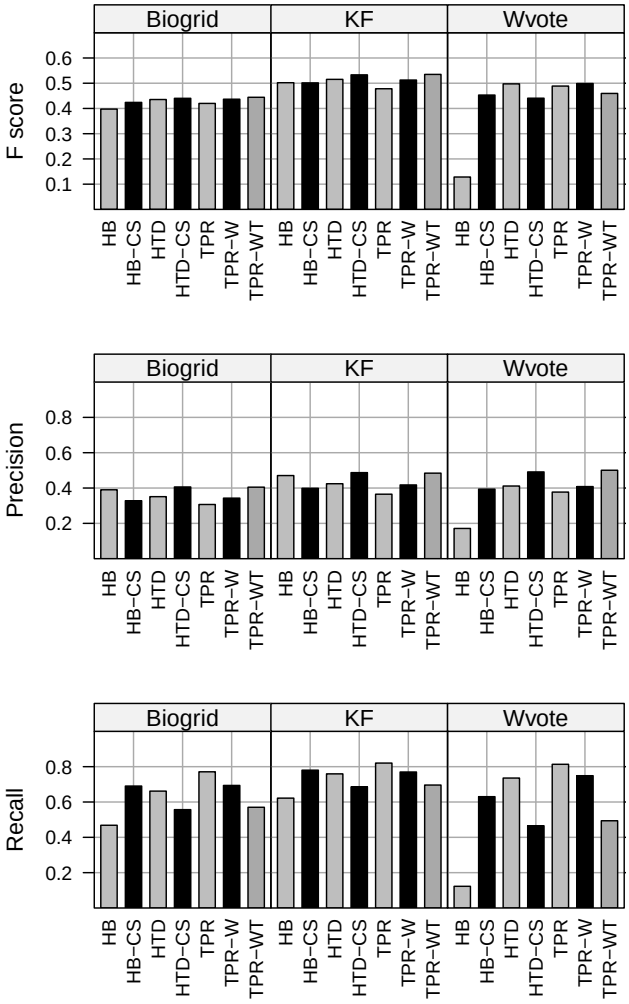


Fig. 9: *Comparison of hierarchical F-score, precision and recall, among different ensemble methods using the best source of biomolecular data (*BIOGRID*), Kernel Fusion (*KF*), and Weighted Voting (*WVOTE*) data integration techniques, with the* Basic *strategy to select negatives.*

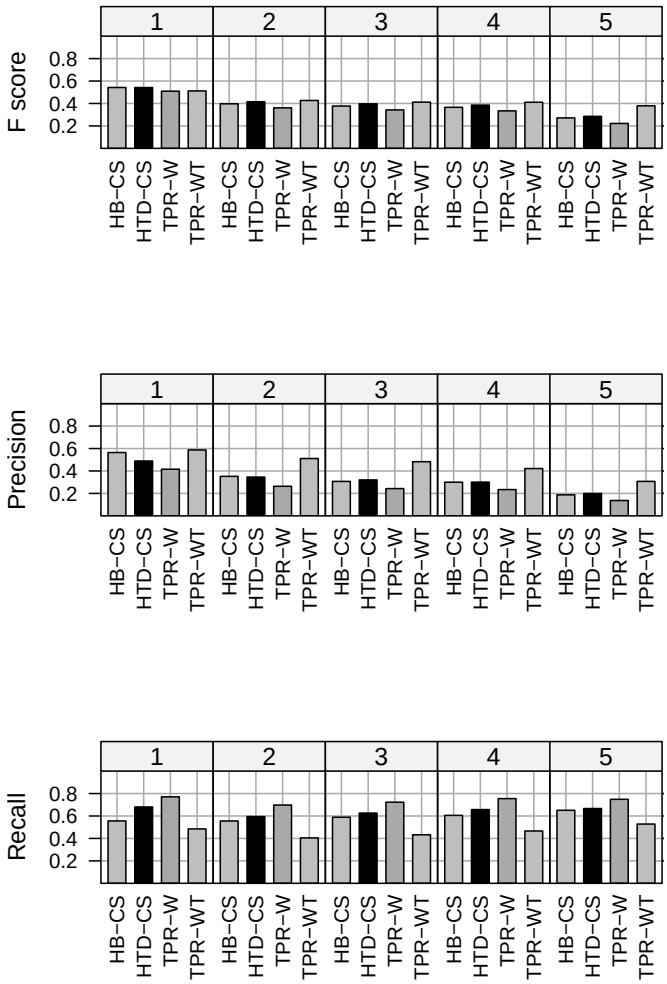


Fig. 10: *Per level average F-score, precision and recall across the five levels of the FunCat taxonomy in* HBAYES-CS,, HTD-CS, TPR-W *and* TPR-W-T *ensembles using Kernel Fusion data integration, with the* Basic *strategy to select negatives.*