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

Paper Review

Miguel A. Veganzones

Grupo de Inteligencia Computacional
Universidad del País Vasco

2012-01-27

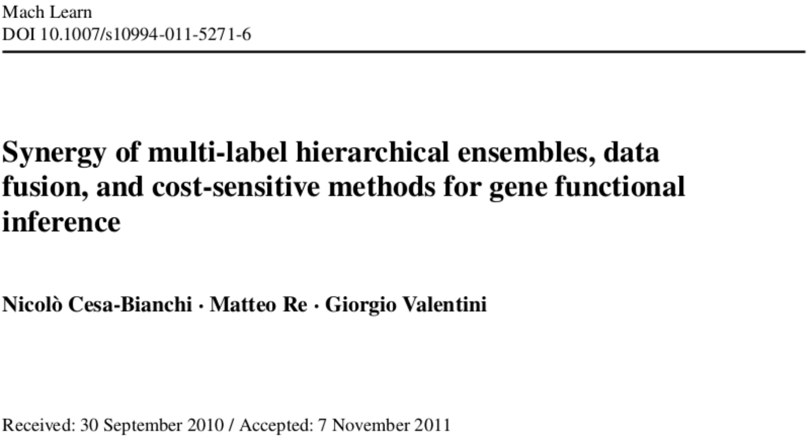
Outline

1. Introduction
2. Related work
3. Methods
4. Experimental set-up
5. Results and discussion
6. Conclusions

Outline

1. Introduction
2. Related work
3. Methods
4. Experimental set-up
5. Results and discussion
6. Conclusions

Paper



Mach Learn
DOI 10.1007/s10994-011-5271-6

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

Nicolò Cesa-Bianchi · Matteo Re · Giorgio Valentini

Received: 30 September 2010 / Accepted: 7 November 2011

Motivation

- The applications of multilabel classification span a large range of real-world applications: music categorization, web search and mining, semantic scene classification, directed marketing and *functional genomics*.
- Constraints between labels and, more in general, the issue of label dependence have been recognized to play a central role in multilabel learning.
- 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.
    - A forest of trees in the Functional Catalogue.

GFP challenges

- Large number of functional classes: hundreds for Functional Catalogue (FunCat) or thousands for the Gene Ontology (GO).
- Multiple annotations for each gene.
- Hierarchical relationships between functional classes: labels are not independent because functional classes are hierarchically organized.
- Multiple sources of data: high-throughput biotechnologies make available an increasing number of sources of genomic and proteomic data.

GFP challenges (II)

- Complex and noisy data.
- Unbalanced classes: 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.
- Different reliability of functional labels: functional annotations have different degrees of evidence.

Contributions

- 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.
- 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 with hierarchical multilabel cost-sensitive algorithms.
  - The resulting system is tested on genome and ontology-wide classification of genes according to the FunCat taxonomy.

Outline

1. Introduction
2. Related work
3. Methods
4. Experimental set-up
5. Results and discussion
6. Conclusions

Overview

- 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.

Label propagation methods

- 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.
- These methods are based on transductive label propagation algorithms: they predict the labels of unannotated examples without using a global predictive model.
- A network-based approach, alternative to label propagation and exhibiting strong theoretical predictive guarantees in the so-called *mistake bound model*.
    - This alternative method is extremely efficient: in most cases training and prediction take both time sublinear in the network size.

Decision tree-based methods

- 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.
- 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.
- In particular, these methods treat structured prediction as a maximum a-posteriori prediction problem.

Hierarchical ensemble methods

- 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.
- **The multilabel hierarchical approaches studied in this paper belong to this research line.**

Overwiew

- The integration of multiple sources of heterogeneous biomolecular data is the key to the prediction of gene function at genome-wide level.
- The main approaches proposed in the literature can be schematically grouped in four categories:
    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.
- This network is then processed by a transduction algorithm that assigns all missing labels.

Vector space integration

- In vector space integration vectorial data are concatenated to combine different data sources.
- 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, one may also use semidefinite programming to learn the coefficients.
    - 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.
- Kernel fusion methods, both with and without weighting the data sources, have been successfully applied to the classification of gene functions.

Ensemble methods

- Recently, Re and Valentini (2010c) showed that simple ensemble methods, such as weighted voting or Decision Templates 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 re-training the entire ensemble.
- Compared to kernel fusion methods, ensemble methods are also more robust to noisy data.

Outline

1. Introduction
2. Related work
3. Methods
4. Experimental set-up
5. Results and discussion
6. Conclusions

Gens representation

- We represent a gene g with a vector $\mathbf{x} \in \mathbb{R}^d$ 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 $\Omega = \{\omega_1, \ldots, \omega_m\}$ structured according to a FunCat tree T.
- The assignments are coded through a vector of multilabels $v = (v_1, \ldots, v_m) \in \{0, 1\}^m$, where g belongs to class ω_i if and only if $v_i = 1$.

FunCat tree

- 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.
    - The root of T is a dummy class ω_0, which every gene belongs to, that we added to facilitate the processing.
- We represent by $\mathrm{child}(i)$ the set of nodes that are children of i and by $\mathrm{par}(i)$ the parent of i.
    - $v_{\mathrm{par}(i)} = 1$ means that the gene under consideration belongs to the parent class of i.

FunCat tree (II)

- 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 $\mathrm{par}(i)$.

Hierarchical ensemble methods

- 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 $\hat{p}_i(g)$ of the probabilities $p_i(g) = \mathbb{P}\left(V_i = 1 | V_{\mathrm{par}(i)} = 1, g\right)$ 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.

Data fusion

- Data integration is performed locally at each node/class of the FunCat taxonomy.
- We consider two techniques: ensemble (weighted voting) and kernel fusion.
- Given L different sources $D_1, \dots, D_L$ of biomolecular data, we train node classifiers $c_{t,i}$ on the data set D_t, one for each class ω_i, $i = 1, \dots, m$.
- Let $\hat{p}_{t,i}(g)$ be the estimate of the probability $\mathbb{P}\left(V_i = 1 | V_{\mathrm{par}(i)} = 1, g\right)$ computed by the classifier $c_{t,i}$.

Ensemble weighted voting

- 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:

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

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.

Ensemble weighted voting (II)

- Given a gene g, the decision $\hat{y}_i$ of the ensemble about the class ω_i is taken by:

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

where output 1 corresponds to assigning class ω_i to g.

Kernel fusion

- Given a pair of genes g, g', and their corresponding pairs of feature vectors $\mathbf{x}_t, \mathbf{x}'_t \in D_t$, we implement a kernel averaging function $K_{\text{ave}}(g, g')$ by simply averaging the output of kernel functions $K_1, \ldots, K_L$ specific to each data set:

$$K_{\text{ave}}(g, g') = \frac{1}{L} \sum_{t=1}^{L} K_t(\boldsymbol{x}_t, \boldsymbol{x}'_t)$$

HTD

- The hierarchical Top-Down ensemble method (HTD) computes predictions in a top-down fashion (i.e., assigning y_i before assigning the label of any j in the subtree rooted at i).
- 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.

HTD implementation

- 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 $\hat{p}_i(g)$ of $\mathbb{P}\left(V_i = 1 | V_{\mathrm{par}(i)} = 1, g\right)$ 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 $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_m)$ are consistent with the hierarchy.

HBAYES

- The ensemble method HBAYES provides an approximation of the optimal Bayesian classifier w.r.t. the H-loss.
- 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 $l_H(\hat{\mathbf{y}}, \mathbf{v})$ is computed as follows:
  - All paths in the taxonomy T from the root down to each leaf are examined.
  - Whenever a node $i \in \{1, \ldots, m\}$ is encountered such that $\hat{y}_i \neq v_i$, then c_i is added to the loss.
  - All the other loss contributions from the subtree rooted at i are discarded.

HBAYES evaluation

- The optimal multilabel for g is the one that minimizes the loss when the true multilabel $\mathbf{V}$ is drawn from the joint distribution computed from the estimated conditionals $p_i(g)$. That is,

$$\widehat{\boldsymbol{y}} = \underset{\boldsymbol{y}\in\{0,1\}^m}{\operatorname{argmin}} \mathbb{E}\left[\ell_H(\boldsymbol{y}, \boldsymbol{V}) \mid g\right]$$

HBAYES prediction

- Initially, set the labels of each node *i* to

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

  where

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

  is recursively defined over the nodes *j* in the subtree rooted at *i*.

- Then, if y_i is set to zero, set all nodes in the subtree rooted at *i* to zero as well.

HBAYES VS HTD

- Unlike standard top-down hierarchical methods (HTD) each $\hat{y}_i$ also depends on the classification of its child nodes.
- If all child nodes k of i have $\hat{p}_k$ close to a half, then the Bayes-optimal label of i tends to be 0 irrespective of the value of p_i.
- If i's children all have $\hat{p}_k$ close to either 0 or 1, then the Bayes-optimal label of i is based on $\hat{p}_i$ only, ignoring the children.

True Path Rule

- “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.”
- Considering the parents of a given node i, a classifier that respects the true path rule needs to obey the following rules:

$$\begin{cases} y_i = 1 \Rightarrow y_{\mathrm{par}(i)} = 1, \\ y_i = 0 \not\Rightarrow y_{\mathrm{par}(i)} = 0. \end{cases}$$

- Considering the children of a given node i, a classifier that respects the true path rule needs to obey the following rules:

$$\begin{cases} y_i = 1 \not\Rightarrow y_{\mathrm{child}(i)} = 1, \\ y_i = 0 \Rightarrow y_{\mathrm{child}(i)} = 0. \end{cases}$$

True Path Rule for ensembles

- According to True Path Rules, in TPR ensembles positive predictions for a node influence in a recursive way their ancestors, while negative predictions influence their offsprings.
- 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 $p_i(g)$.

Global consensus probability

- 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:

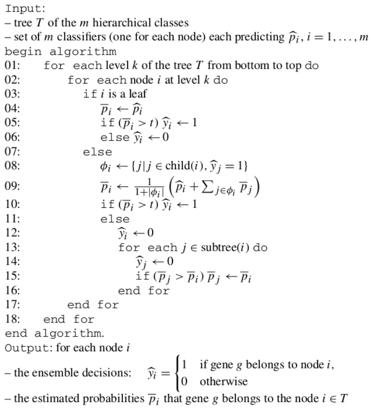
$$\phi_i(g) = \left\{ j \, : \, j \in \text{child}(i), \, \widehat{y}_j = 1 \right\}$$

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

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

- The decision $\hat{y}_i(g)$ 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).

Pseudocode



```
Input:
– tree $T$ of the $m$ hierarchical classes
– set of $m$ classifiers (one for each node) each predicting $\widehat{p}_i$, $i = 1, \ldots, m$
begin algorithm
    for each level $k$ of the tree $T$ from bottom to top do
        for each node $i$ at level $k$ do
            if $i$ is a leaf
                $\overline{p}_i \leftarrow \widehat{p}_i$
                if $(\overline{p}_i > t)$ $\widehat{y}_i \leftarrow 1$
                else $\widehat{y}_i \leftarrow 0$
            else
                $\phi_i \leftarrow \{j | j \in \mathrm{child}(i), \widehat{y}_j = 1\}$
                $\overline{p}_i \leftarrow \frac{1}{1+|\phi_i|}\left(\widehat{p}_i + \sum_{j\in\phi_i} \overline{p}_j\right)$
                if $(\overline{p}_i > t)$ $\widehat{y}_i \leftarrow 1$
                else
                    $\widehat{y}_i \leftarrow 0$
                    for each $j \in \mathrm{subtree}(i)$ do
                        $\widehat{y}_j \leftarrow 0$
                        if $(\overline{p}_j > \overline{p}_i)$ $\overline{p}_j \leftarrow \overline{p}_i$
                    end for
        end for
    end for
end algorithm.
Output: for each node $i$
– the ensemble decisions: $\widehat{y}_i = \begin{cases} 1 & \text{if gene } g \text{ belongs to node } i, \\ 0 & \text{otherwise} \end{cases}$
– the estimated probabilities $\overline{p}_i$ that gene $g$ belongs to the node $i \in T$
```

Fig. 1 True Path Rule multilabel hierarchical algorithm

Overview

- 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 $\hat{p}_i$ 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

- 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:

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

- By tuning τ we may obtain ensembles with different precision/recall characteristics.

HBAYES-CS

- HBAYES - CS distinguishes the cost c_i^- of a false negative (FN) mistake from the cost c_i^+ of a false positive (FP) mistake. Then:

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

- 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 $c_i^+ + c_i^- = 2c_i$. Rewriting:

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

HBAYES-CS (II)

- 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.

HBAYES-CS (III)

- We can set the α parameter experimentally (e.g., by cross-validation on the training data), or
- Choose a cost factor α_i for each node i to explicitly take into account the unbalance between the number of positive and negative examples, estimated from the training data:

$$\alpha_i = \frac{n_i^-}{n_i^+} \quad \Rightarrow \quad c_i^+ = \frac{2}{\frac{n_i^-}{n_i^+} + 1} c_i = \frac{2n_i^+}{n_i^- + n_i^+} c_i$$

  - The decision rule at each node then becomes:

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

TPR-W (weighted)

- In the TPR algorithm there is no way to explicitly balance the local prediction $\hat{p}_i(\mathbf{x})$ at node i 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.

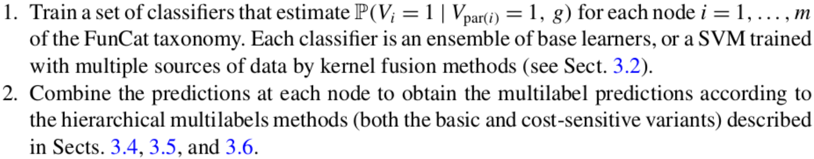
TPR-W (weighted) (II)

- 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 between the local parent predictor and the set of its children:

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

- By tuning the w parameter we can modulate the precision/recall characteristics of the resulting ensemble.

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 $i = 1, \dots, m$ 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 Sect. 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 Sects. 3.4, 3.5, and 3.6.

Outline

1. Introduction
2. Related work
3. Methods
4. Experimental set-up
5. Results and discussion
6. Conclusions

Data sources

- We integrated six different sources of yeast biomolecular data, previously used for single-source ontology-wide gene function prediction:
  - 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).
  - Pairwise sequence similarity data (SEQ.SIM.).

Gen selection

- We considered only yeast genes common to all data sets.
- In order to get a not too small set of positive examples for training, for each data set we selected only the FunCat-annotated genes and the classes with at least 20 positive examples.
- This selection process yielded 1901 yeast genes annotated to 168 FunCat classes distributed across 16 trees and 5 hierarchical levels.

Classification tasks

1. Comparison of "single-source" and data fusion techniques (kernel fusion and weighted voting) using both FLAT and hierarchical methods (HTD, HBAYES and TPR)
2. Assessment of the improvements achievable by:
    1. Multilabel hierarchical methods vs. flat methods.
    2. Cost-sensitive vs cost-insensitive strategies.
    3. Synergic en hancements due to the concurrent application of multilabel hierarchical methods, cost-sensitive, and data fusion techniques.
3. Analysis of the precision-recall characteristics of the compared methods.
4. Impact of the choice strategy for selecting negative examples.

Baseline method

- 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.

Task 4: selecting negative examples

- We tested whether training base learners with different strategies for choosing negative examples may have an impact on the generalization capabilities of mul- tilabel hierarchical methods.
- 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.
- 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.

SVM parameters

- 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.
- 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.

Resampling

- In order to assess the generalization capabilities of the ensembles, we adopted “external” 5-fold cross validation techniques.
- 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.
- F-score is the 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.

Hierarchical F-measure

- 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.
- 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).

Hierarchical F-measure (II)

- 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:

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

where $l(\cdot)$ is the set of leaves of a tree.

Outline

1. Introduction
2. Related work
3. Methods
4. Experimental set-up
5. Results and discussion
6. Conclusions

Synergy

- 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.

Average F-scores

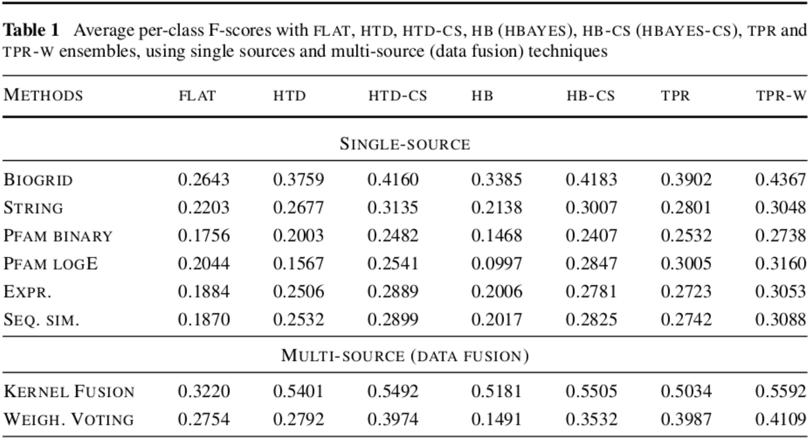


Table 1 Average per-class F-scores with FLAT, HTD, HTD-CS, HB (HBAYES), HB-CS (HBAYES-CS), TPR and TPR-W ensembles, using single sources and multi-source (data fusion) techniques

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

Statistical significance

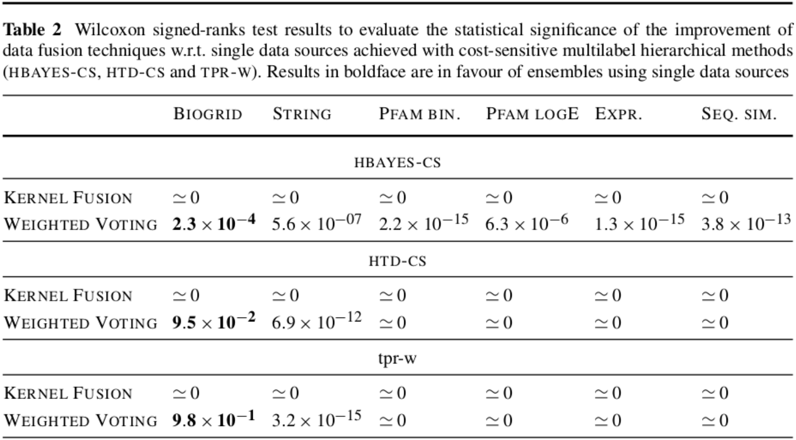
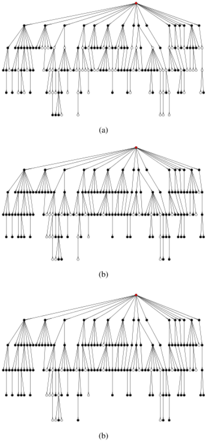


Table 2 Wilcoxon signed-ranks test results to evaluate the statistical significance of the improvement of data fusion techniques w.r.t. single data sources achieved with cost-sensitive multilabel hierarchical methods (HBAYES-CS, HTD-CS and TPR-W). Results in boldface are in favour of ensembles using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

	BIOGRID	STRING	PFAM BIN.	PFAM LOGE	EXPR.	SEQ. SIM.
			HBAYES-CS			
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			
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			
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}	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$

FunCat trees

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



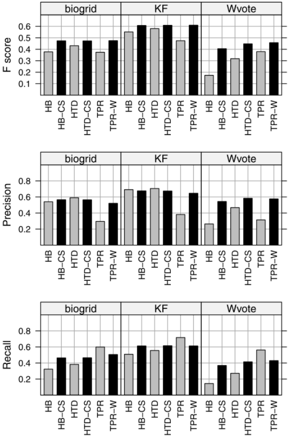
(a)

(b)

(b)

Hierarchical F-score

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



Statistical significance

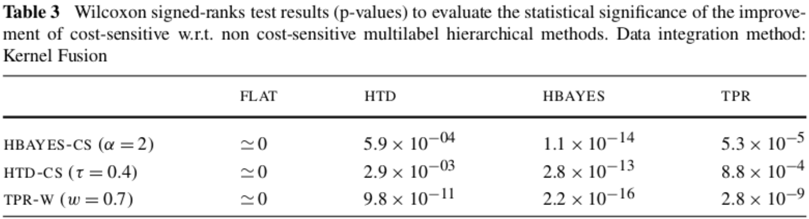


Table 3 Wilcoxon signed-ranks test results (p-values) to evaluate the statistical significance of the improvement of cost-sensitive w.r.t. non cost-sensitive multilabel hierarchical methods. Data integration method: Kernel Fusion

	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}

Per level average performance

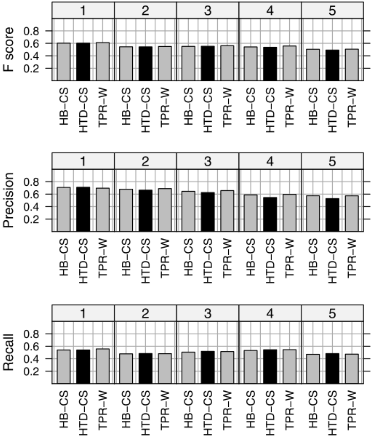


Fig. 4 Per level average F-score, precision and recall across the five levels of the FunCat taxonomy in HBAYES-CS, HTD-CS and TPR-W ensembles using Kernel Fusion data integration. Number 1 to 5 refer to levels: level 1 is the top level, level 5 the bottom

Per level average performance (II)

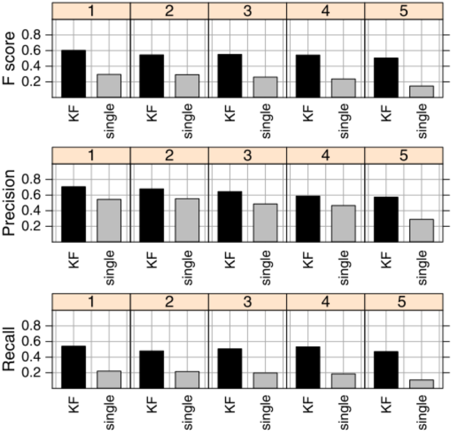


Fig. 5 Comparison of per level average F-score, precision and recall across the five levels of the FunCat taxonomy in HBAYES-CS using single data sets (single) and kernel fusion techniques (KF). Performance of "single" are computed by averaging across all the single data sources

Average per class recall

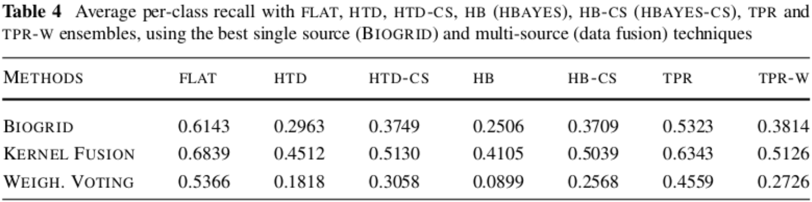


Table 4 Average per-class recall with FLAT, HTD, HTD-CS, HB (HBAYES), HB-CS (HBAYES-CS), TPR and TPR-W ensembles, using the best single source (Biogrid) and multi-source (data fusion) techniques

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

Average per class precision

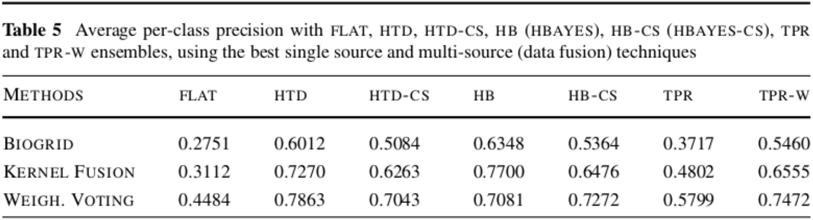
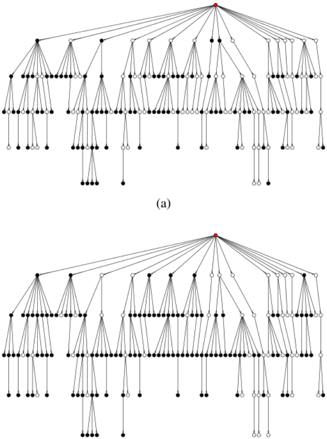


Table 5 Average per-class precision with FLAT, HTD, HTD-CS, HB (HBAYES), HB-CS (HBAYES-CS), TPR and TPR-W ensembles, using the best single source and multi-source (data fusion) techniques

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

Ontology-wide FunCat tree

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)

Hierarchical measures

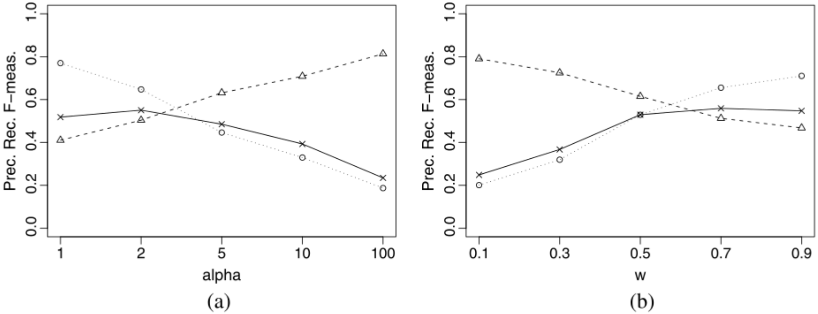


Fig. 7 Hierarchical F-score, precision and recall as functions of global cost sensitive parameters. (**a**) HBAYES-CS, (**b**) TPR-W

Average per class F-scores

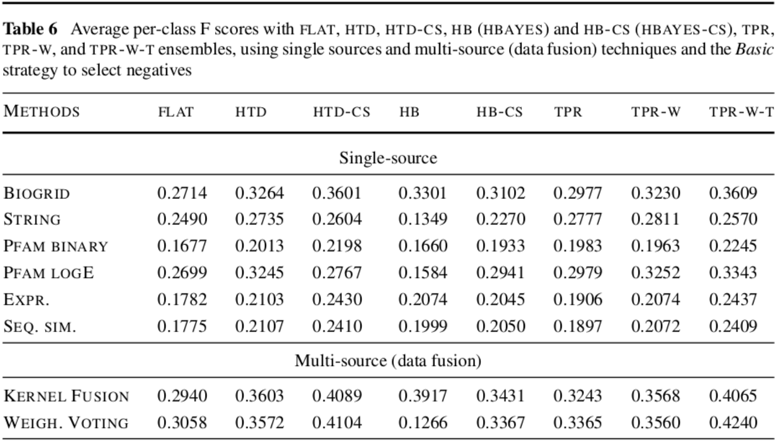


Table 6 Average per-class F scores with FLAT, HTD, HTD-CS, HB (HBAYES) and HB-CS (HBAYES-CS), TPR, TPR-W, and TPR-W-T ensembles, using single sources and multi-source (data fusion) techniques and the *Basic* strategy to select negatives

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

Average per class F-score (II)

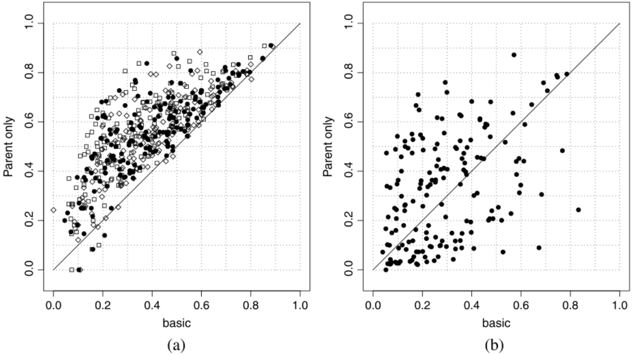
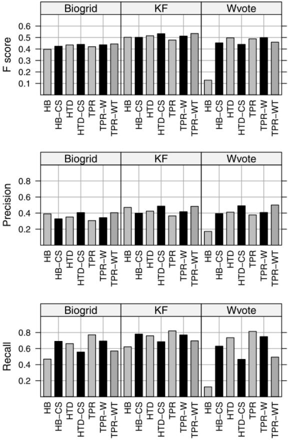


Fig. 8 Comparison of average per-class F-score between *Basic* and *PO* strategies. (**a**) Hierarchical cost-sensitive strategies: HTD-CS (*squares*), TPR-W (*triangles*), HBAYES-CS (*filled circles*). (**b**) FLAT. Abscissa: per–class F-score with base learners trained according to the *Basic* strategy;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

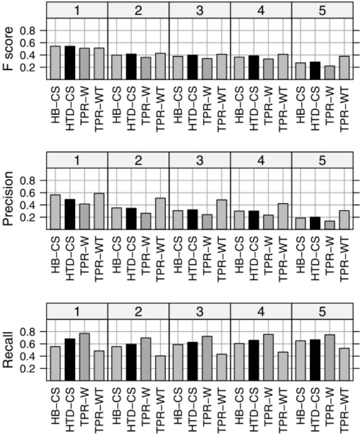
Hierarchical measures

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



Per level average measures

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



Outline

1. Introduction
2. Related work
3. Methods
4. Experimental set-up
5. Results and discussion
6. Conclusions

Summary

- 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.
    - 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.

Conclusions I

- 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.

Conclusions II

  - 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).

Conclusions III

  - 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.
- 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.
- 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.

Conclusions IV

  - This is of paramount importance when we need to reduce the costs of the biological validation of new gene functions discovered through computational methods.
- 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.