Multitask Protein Function Prediction Through Task Dissimilarity

Marco Frasca, Nicolò Cesa Bianchi

Abstract—Automated protein function prediction is a challenging problem with distinctive features, such as the hierarchical organization of protein functions and the scarcity of annotated proteins for most biological functions. We propose a multitask learning algorithm addressing both issues. Unlike standard multitask algorithms, which use task (protein functions) similarity information as a bias to speed up learning, we show that dissimilarity information enforces separation of rare class labels from frequent class labels, and for this reason is better suited for solving unbalanced protein function prediction problems. We support our claim by showing that a multitask extension of the label propagation algorithm empirically works best when the task relatedness information is represented using a dissimilarity matrix as opposed to a similarity matrix. Moreover, the experimental comparison carried out on three model organism shows that our method has a more stable performance in both "protein-centric" and "function-centric" evaluation settings.

Index Terms—Multitask learning; protein function prediction; label propagation algorithm; Gene Ontology; task dissimilarity.

1 Introduction

The constant increase in the volume and variety of publicly available genomic and proteomic data is a characteristic trait of modern biomedical sciences. A fundamental problem in this area is the assignment of functions to biological macromolecules, especially proteins. Indeed, the accurate annotation of protein function would also have great biomedical and pharmaceutical implications, since several human diseases have genetic causes. While molecular experiments provide the most reliable annotation of proteins, their relatively low throughput and restricted scope have led to an increasing role for automated function prediction (AFP). AFP is characterized by unbalanced functional classes with rare positive instances. Moreover, since only positive membership to functional classes is usually assessed, negative instances are not uniquely defined, and different approaches to choose them have been proposed [1], [2], [3]. Other peculiarities of AFP include: (1) the need of integrating several heterogeneous sources of genomic, proteomic, and transcriptomic data in order to achieve more accurate predictions [4], [5]; (2) the presence of multiple labels and dependencies among class labels; (3) the hierarchical structure of functional classes (a direct acyclic graph for the *Gene Ontology* GO [6], a forest of trees for the *FunCat* taxonomy [7]) with different levels of specificity.

Recently, two international challenges for Critical Assessment of Functional Annotation, (CAFA [8] and CAFA2 [9]) were organized to evaluate computational methods that automatically assign protein functions. In particular, CAFA2 emphasized the need for multilabel or structured-output learning algorithms for predicting a set of terms, or a subgraph of the GO ontology for a given protein. In this work we mainly focus on this problem, whose solution however requires paying attention also to the other aspects of AFP.

Several approaches to the predicton of protein functions were proposed in the literature, including sequence-based [10], [11], [12] and network-based methods [13], [14], [15], structured output algorithms based on kernels [3], [16], [17] and hierarchical ensemble methods [18], [19], [20]. In particular, the availability of large-scale networks, in which nodes are genes/proteins and edges their functional pairwise relationships, has promoted the development of several machine learning methods where novel annotations are inferred by exploiting the topology of the resulting biomolecular network. Initially, network-based approaches relied on the so called *guilt-by-association* (GBA) rule, which makes predictions assuming that interacting proteins are likely to share similar functions [21], [22], [23]. Indirect neighbours were also exploited to modify the notion of pairwise-similarities among nodes by accounting for pairs of nodes connected through intermediate ones [24], [25]. Protein functions can be also propagated through the network with an iterative process until convergence [26], [27], by tuning the amount of propagation allowed in the graph through Markov random walks [28], [29], by evaluating the functional flow through the nodes [30], by exploiting kernelized score functions [31], and by modelling protein memberships through Markov Random Fields [32] and Gaussian Random Fields [33], [34]. Furthermore, methods based on the convergence of classical [35], [36] and multicategory Hopfield networks [37] were recently proposed to specifically tackle the class imbalance.

Although protein functions are clearly dependent (see, e.g., the GO functions, where parent terms include all the proteins of their children) most AFP methods described above predict biological functions independently from each other. Multitask methods, on the other hand, take advantage of existing dependencies by transferring information between related tasks, which typically leads to learning faster than algorithms trained independently on each task.

- *M. Frasca and N. Cesa Bianchi are with the Dipartimento di Informatica, Università degli Studi di Milano, Via Comelico 39 Milano, 20137, Italy. E-mail: {marco.frasca, nicolo.cesa-bianchi}@unimi.it*

Preprint submitted to the IEEE Transactions on Computational Biology and Bioinformatics, 2017; Accepted March 17, 2017.

In this paper we investigate an alternative approach to multitask learning based on exploiting task dissimilarities rather than similarities. In particular, we consider two multitask extensions of a known label propagation algorithm [26]: the first extension follows a standard multitask approach based on task similarities; the second extension learns instead from task dissimilarities. Both approaches can be naturally applied to the multilabel prediction of proteins. The prediction tasks we consider are the GO protein functions of *fly*, *human*, and *bacteria* model organisms. We compute different measures of similarity/dissimilarity between GO terms, taking into account both GO structure and protein annotations. We show that the approach learning from task dissimilarities greatly helps in unbalanced tasks (by helping instances labeled with the rare class labels to be correctly classified), and does not hurt in the more balanced cases. This is a crucial point in protein function prediction, since terms better describing protein functions —i.e., the most specific ones— are the most unbalanced (proteins annotated with these terms are very rare). On the other hand, learning from similar tasks tends to be more effective on balanced settings. Note that the proposed multitask extensions of label propagation do not increase the overall running time of the algorithm, allowing its application on large-sized datasets. Finally, we compare our methods with the state-of-the-art methodologies for AFP by considering both "term-centric" and "protein-centric" evaluation settings.

The paper is organized as follows. In Section 2 we formally introduce the problem and in Section 3 we describe the proposed multitask label propagation methodology. Section 4 is dedicated to the experimental validation of the method on a real-world application.

2 Automated Protein Function Prediction

The Automated protein Function Prediction (AFP) problem can be formalized as semi-supervised learning problem on a weighted and undirected graph $G = (V, E, \boldsymbol{W})$, where $V = \{1, \dots, n\}$ is the set of vertices, $E \subset V \times V$ is the set of edges, and $\boldsymbol{W} = [w_{ij}]_{n \times n}$ is the symmetric weight matrix, where w_{ij} is the weight on the edge between vertices i and j (we assume $w_{ii} = 0$ and $w_{ij} = 0$ for all $(i,j) \notin E$).

A set of m binary classification tasks on G is defined by m labelings $\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(m)} \in \{-1, 1\}^n$ of the nodes in V, where $y_i^{(k)}$ is the label of node i for task k. For any subset $T \subseteq \{1, \dots, n\}$ and any vector $\boldsymbol{y} = (y_1, \dots, y_n)$, we use $\boldsymbol{y}_T$ to denote the vector obtained from $\boldsymbol{y}$ by retaining only the coordinates in T.

The multitask prediction problem on the graph G is then defined as follows. Given a set $S \subset V$ of training vertices and the complement set $U \equiv V \setminus S$ of test vertices, the learner must predict the test labels $\boldsymbol{y}_U^{(1)}, \dots, \boldsymbol{y}_U^{(m)}$ for each task given the training labels $\boldsymbol{y}_S^{(1)}, \dots, \boldsymbol{y}_S^{(m)}$ for the same tasks.

3 Methods

We first describe the standard label propagation algorithm [26], [38], [39] for single-task classification on graphs. This will be later extended to the multitask setting.

3.1 Label Propagation (LP)

In the single-task setting, a standard notion of *regularity* of a labeling $\boldsymbol{f} \in \{-1, 1\}^n$ on a graph G is the *weighted cutsize* induced by $\boldsymbol{f}$ and defined as follows:

$$\Gamma_G^W(\boldsymbol{f}) = \sum_{\substack{(i,j) \in E \\ f_i \neq f_j}} w_{ij} \ . \tag{1}$$

The weighted cutsize can be also expressed as a quadratic form

$$\Gamma_G^W(\boldsymbol{f}) = \frac{1}{4} \boldsymbol{f}^\top L \boldsymbol{f} = \frac{1}{4} \sum_{(i,j) \in E} w_{ij} (f_i - f_j)^2 \ .$$

The matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the *Laplacian* of G, where $\boldsymbol{D}$ is the diagonal matrix with entries $D_{ii} = d_i = \sum_j w_{ij}$. The Label Propagation algorithm minimizes the above quadratic form over real-valued (rather than binary) labels. More precisely, LP finds the unique solution of

$$\begin{aligned} &\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^\top L \boldsymbol{f} \\ &\text{s.t.} \quad f_i = y_i \quad i \in S \ . \end{aligned} \tag{2}$$

The solution $\boldsymbol{f}_U^*$ of (2) is smooth on G. Namely, if two vertices $i, j \in U$ are connected with a large weight w_{ij}, then f_i^* is close to f_j^*. Indeed, the components $i \in U$ of f^* satisfy the harmonic property [26]

$$f_i^* = \frac{1}{d_i} \sum_j w_{ij} f_j^* \ .$$

The vector $\boldsymbol{f}_U^*$ can be also written in closed form as

$$\boldsymbol{f}_U^* = (\boldsymbol{D}_{UU} - \boldsymbol{W}_{UU})^{-1} \boldsymbol{W}_{US} \, \boldsymbol{f}_S^* \tag{3}$$

where

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}_{UU} & \boldsymbol{W}_{US} \\ \boldsymbol{W}_{US}^T & \boldsymbol{W}_{SS} \end{pmatrix}$$

is the weight matrix partitioned in blocks to emphasize the labeled and unlabeled part of the graph (similarly for the matrix $\boldsymbol{D}$). As the components of $\boldsymbol{f}_U^*$ given by (3) are not in $\{-1, 1\}$, the final labeling produced by LP is obtained by thresholding each component f_i^* for $i \in U$.

3.2 Multitask label propagation (MTLP)

It is fairly easy to use similarity or dissimilarity information between tasks in order to generalize LP to multitask learning, while preserving the regularity of every task in the sense of (1).

We start by considering multitask LP based on similarity information. Suppose a $m \times m$ symmetric matrix $\boldsymbol{\mathcal{C}}$ is given, where each entry $\mathcal{C}_{kr} \in [0, 1]$ quantifies the relatedness between tasks k and r. Let $\boldsymbol{\mathcal{A}} = \gamma \boldsymbol{\mathcal{I}}_m + \boldsymbol{\mathcal{L}}$ be the matrix where $\gamma > 0$, $\boldsymbol{\mathcal{I}}_m$ is the $m \times m$ identity matrix, and $\boldsymbol{\mathcal{L}}$ is the Laplacian of $\boldsymbol{\mathcal{C}}$. The matrix $\boldsymbol{\mathcal{A}}$ is symmetric and positive definite, since $\boldsymbol{\mathcal{A}}$ is diagonally dominant with positive diagonals, and thus invertible. Denote by $\boldsymbol{Y}$ the $n \times m$ label matrix whose k-th column is the vector $\boldsymbol{y}^{(k)}$, and by $\boldsymbol{F}$ the $n \times m$ matrix whose k-th column is the vector $\boldsymbol{f}^{(k)}$.

When learning multiple related tasks, a widely used approach is requiring that similar tasks be assigned similar labelings. To this end, we introduce the linear map $\psi_{\boldsymbol{\mathcal{A}}^{-1}} : \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times m}$, defined as follows:

$$\psi_{\boldsymbol{\mathcal{A}}^{-1}}(\boldsymbol{Y}) = \boldsymbol{Y}\boldsymbol{\mathcal{A}}^{-1} \,. \tag{4}$$

It can be shown that the map $\psi_{\boldsymbol{\mathcal{A}}^{-1}}$ acts on a multitask labeling matrix $\boldsymbol{Y}$ by getting closer (in Euclidean distance) the labelings (columns of $\boldsymbol{Y}$) corresponding to tasks that are similar according to $\boldsymbol{\mathcal{C}}$.

By means of $\psi_{\boldsymbol{\mathcal{A}}^{-1}}$, the exploitation of task similarities can be encoded into the learning problem (2) as follows:

$$\begin{aligned} &\min_{\boldsymbol{F}} \operatorname{trace}(\boldsymbol{F}^\top \boldsymbol{L}\boldsymbol{F}) \\ &\text{s.t.} \quad F_{ik} = \widetilde{Y}_{ik} \quad i \in S,\, k = 1, \ldots, m \end{aligned} \tag{5}$$

where $\widetilde{\boldsymbol{Y}} = \psi_{\boldsymbol{\mathcal{A}}^{-1}}(\boldsymbol{Y}) = \boldsymbol{Y}\boldsymbol{\mathcal{A}}^{-1}$. The solution to (5) is

$$\widetilde{\boldsymbol{F}}_U = (\boldsymbol{D}_{UU} - \boldsymbol{W}_{UU})^{-1}\boldsymbol{W}_{US}\widetilde{\boldsymbol{Y}}_S$$

where $\widetilde{\boldsymbol{F}}_U$ is the submatrix of $\boldsymbol{F}$ including only the rows indexed by U, and $\widetilde{\boldsymbol{Y}}_S$ is the submatrix of $\widetilde{\boldsymbol{Y}}$ including only the rows indexed by S. By observing that $\widetilde{\boldsymbol{Y}}_S = \boldsymbol{Y}_S\boldsymbol{\mathcal{A}}^{-1}$, we have

$$\widetilde{\boldsymbol{F}}_U = (\boldsymbol{D}_{UU} - \boldsymbol{W}_{UU})^{-1}\boldsymbol{W}_{US}\boldsymbol{Y}_S\boldsymbol{\mathcal{A}}^{-1} = \boldsymbol{F}_U^*\boldsymbol{\mathcal{A}}^{-1}$$

where $\boldsymbol{F}_U^*$ is the solution of (5) with constraints $F_{ik} = Y_{ik}$ for $i \in S$ and $k = 1, \ldots, m$. The equality $\widetilde{\boldsymbol{F}}_U = \boldsymbol{F}_U^*\boldsymbol{\mathcal{A}}^{-1}$ shows that it is equivalent to apply the task feature map (4) before or after performing label propagation. This ensures that the multitask mapping does not increase the label propagation complexity.

As we show next, this solution does not perform well on unbalanced classification problems, where some class (typically the positive class) is largely underrepresented. We propose here an alternative approach, which exploits the prior information about task relatedness in an "inverse" manner. Specifically, we propose a multitask label propagation algorithm which learns multiple tasks by requiring that dissimilar tasks be assigned dissimilar labelings. As we see in the experiments, this approach turns out to work particularly well on unbalanced classification problems.

The first component of our method is a dissimilarity matrix $\overline{\boldsymbol{\mathcal{C}}}$, where $\overline{\mathcal{C}}_{kr} \in [0, 1]$ is measure of dissimilarity between tasks k and r (we discuss in Section 3.2.2 possible choices for the matrices $\boldsymbol{\mathcal{C}}$ and $\overline{\boldsymbol{\mathcal{C}}}$).

Given the matrix $\overline{\boldsymbol{\mathcal{C}}}$, we consider the multitask map $\psi_{\overline{\boldsymbol{\mathcal{A}}}} : \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times m}$, defined as

$$\psi_{\overline{\boldsymbol{\mathcal{A}}}}(\boldsymbol{Y}) = \boldsymbol{Y}\overline{\boldsymbol{\mathcal{A}}} \tag{6}$$

where $\overline{\boldsymbol{\mathcal{A}}} = \overline{\gamma}\boldsymbol{\mathcal{I}}_m + \overline{\boldsymbol{\mathcal{L}}}$, $\overline{\gamma} > 0$, and $\overline{\boldsymbol{\mathcal{L}}}$ is the Laplacian matrix of $\overline{\boldsymbol{\mathcal{C}}}$. Unlike the inverse transformation (4), the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ moves the columns of matrix $\boldsymbol{M}$ farther away from each other, in the sense of the Euclidean distance, in the corresponding n-dimensional space. We formally show that in Section 3.2.1. Using $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ instead of $\psi_{\boldsymbol{\mathcal{A}}^{-1}}$ in (5), we obtain the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{F}} \operatorname{trace}\left(\boldsymbol{F}^\top \boldsymbol{L}\boldsymbol{F}\right) \\ &\text{s.t.} \quad F_{ik} = \widehat{Y}_{ik} \quad i \in S,\, k = 1, \ldots, m \end{aligned} \tag{7}$$

with $\widehat{\boldsymbol{Y}} = \psi_{\overline{\boldsymbol{\mathcal{A}}}}(\boldsymbol{Y})$. Similarly to (5), the solution to (7) is

$$\widehat{\boldsymbol{F}}_U = (\boldsymbol{D}_{UU} - \boldsymbol{W}_{UU})^{-1}\boldsymbol{W}_{US}\boldsymbol{Y}_S\overline{\boldsymbol{\mathcal{A}}} = \boldsymbol{F}_U^*\overline{\boldsymbol{\mathcal{A}}}$$

where $\boldsymbol{F}_U^*$ is the solution of (7) with constraints $F_{ik} = Y_{ik}$ for $i \in S$ and $k = 1, \ldots, m$. Just like in the previous case, the equality $\widehat{\boldsymbol{F}}_U = \boldsymbol{F}_U^*\overline{\boldsymbol{\mathcal{A}}}$ shows that it is equivalent to apply the task feature map (6) before or after performing label propagation.

We call MTLP-inv the similarity-based method (5) and MTLP the dissimilarity-base method (7). In the next section we show some interesting properties of the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ which make MTLP suitable for unbalanced classification problems.

3.2.1 Analysis of the multitask map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$

Given $\boldsymbol{M} \in \mathbb{R}^{n \times m}$, let $\boldsymbol{M}_{i\cdot}$ and $\boldsymbol{M}_{\cdot k}$ be, respectively, the i-th row and the k-th column of the matrix $\boldsymbol{M}$. Let also $\mathcal{P}_i = \{1 \leq k \leq m : Y_{ik} = 1\}$ be the set of tasks for which the instance i is positive, and $\mathcal{N}_i$ the set of tasks for which the instance i is negative. We introduce the following notation: for each $k = 1, \ldots, m$

$$\mathfrak{d}_{k,i}^+ = \sum_{r \in \mathcal{P}_i} \overline{\mathcal{C}}_{rk} \qquad \mathfrak{d}_{k,i}^- = \sum_{r \in \mathcal{N}_i} \overline{\mathcal{C}}_{rk} \qquad \mathfrak{d}_k = \sum_{r=1}^{m} \overline{\mathcal{C}}_{rk}$$

and

$$\mathfrak{a}_{k,i}^+ = \sum_{r \in \mathcal{P}_i} \overline{\mathcal{A}}_{rk} \qquad \mathfrak{a}_{k,i}^- = \sum_{r \in \mathcal{N}_i} \overline{\mathcal{A}}_{rk} \qquad \mathfrak{a}_k = \sum_{r=1}^{m} \overline{\mathcal{A}}_{rk} \,.$$

The next result shows that the action of the linear map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ on the label matrix $\boldsymbol{Y}$ is to change the value of each label without altering the sign. The label of an instance i in task k is made roughly proportional to the weighted sum of tasks in $\overline{\boldsymbol{\mathcal{C}}}$ that are dissimilar to k and have a different label for instance i —see also Corollary 1.

Fact 1. Given $\boldsymbol{Y} \in \{-1, 1\}^{n \times m}$, the task interaction matrix $\overline{\boldsymbol{\mathcal{C}}} \in \mathbb{R}^{m \times m}$, and the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}} : \mathbb{R}^{n \times m} \longrightarrow \mathbb{R}^{n \times m}$ such that $\widehat{\boldsymbol{Y}} = \psi_{\overline{\boldsymbol{\mathcal{A}}}}(\boldsymbol{Y}) = \boldsymbol{Y}\overline{\boldsymbol{\mathcal{A}}}$, where $\overline{\boldsymbol{\mathcal{A}}} = \overline{\gamma}\boldsymbol{\mathcal{I}}_m + \overline{\boldsymbol{\mathcal{L}}}$, then for all $i = 1, \ldots, n$ it holds

$$\widehat{Y}_{ik} = \begin{cases} \overline{\gamma} + 2\mathfrak{d}_{k,i}^- & \text{if } Y_{ik} = +1 \\ -\overline{\gamma} - 2\mathfrak{d}_{k,i}^+ & \text{if } Y_{ik} = -1 \end{cases}$$

Proof: By definition, $\widehat{Y}_{ik} = \sum_{r=1}^{m} Y_{ir}\overline{\mathcal{A}}_{rk} = \mathfrak{a}_{k,i}^+ - \mathfrak{a}_{k,i}^-$. We distinguish the following two cases.

Case 1. $k \in \mathcal{P}_i$. In this case we have $\mathfrak{a}_{k,i}^+ = \overline{\mathcal{A}}_{kk} - \mathfrak{d}_{k,i}^+ = \overline{\gamma} + \mathfrak{d}_k - \mathfrak{d}_{k,i}^+ = \overline{\gamma} + \mathfrak{d}_{k,i}^+ + \mathfrak{d}_{k,i}^- - \mathfrak{d}_{k,i}^+ = \overline{\gamma} + \mathfrak{d}_{k,i}^-$, since by definition $\mathfrak{d}_k = \mathfrak{d}_{k,i}^+ + \mathfrak{d}_{k,i}^-$ for any $i \in \{1, 2, \ldots, n\}$. Moreover, since $k \in \mathcal{P}_i$, we have $\mathfrak{a}_{k,i}^- = -\mathfrak{d}_{k,i}^-$ (by the definition of Laplacian), and accordingly

$$\widehat{Y}_{ik} \;=\; \overline{\gamma} + \mathfrak{d}_{k,i}^- - (-\mathfrak{d}_{k,i}^-) = \overline{\gamma} + 2\mathfrak{d}_{k,i}^-$$

Case 2. $k \in \mathcal{N}_i$. In this case, it holds $\mathfrak{a}_{k,i}^+ = -\mathfrak{d}_{k,i}^+$, whereas $\mathfrak{a}_{k,i}^- = \overline{\mathcal{A}}_{kk} - \mathfrak{d}_{k,i}^- = \overline{\gamma} + \mathfrak{d}_k - \mathfrak{d}_{k,i}^- = \overline{\gamma} + \mathfrak{d}_{k,i}^+$. It follows

$$\widehat{Y}_{ik} \;=\; -\mathfrak{d}_{k,i}^+ - \overline{\gamma} - \mathfrak{d}_{k,i}^+ = -\overline{\gamma} - 2\mathfrak{d}_{k,i}^+$$

The property is proven by observing that $k \in \mathcal{P}_i$ implies $Y_{ik} = +1$ and $k \in \mathcal{N}_i$ implies $Y_{ik} = -1$. □

Using Fact 1 we can show that the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ tends to increase the distance between the labelings $\boldsymbol{Y}_{\cdot r}$ and $\boldsymbol{Y}_{\cdot s}$, for any pair of distinct tasks $r, s \in \{1, 2, \ldots, m\}$. Indeed, we can prove the following.

Fact 2. Given $\boldsymbol{Y} \in \{-1,1\}^{n\times m}$, the task interaction matrix $\overline{\boldsymbol{\mathcal{C}}} \in \mathbb{R}^{m\times m}$, and the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}} : \mathbb{R}^{n\times m} \longrightarrow \mathbb{R}^{n\times m}$ such that $\widehat{\boldsymbol{Y}} = \psi_{\overline{\boldsymbol{\mathcal{A}}}}(\boldsymbol{Y}) = \boldsymbol{Y}\overline{\boldsymbol{\mathcal{A}}}$, where $\overline{\boldsymbol{\mathcal{A}}} = \overline{\gamma}\boldsymbol{\mathcal{I}}_m + \overline{\boldsymbol{\mathcal{L}}}$. Then for every $r, s \in \{1,2,\ldots,m\}$ it holds

$$\|\boldsymbol{Y}_{\cdot r} - \boldsymbol{Y}_{\cdot s}\|^2 \leq \|\widehat{\boldsymbol{Y}}_{\cdot r} - \widehat{\boldsymbol{Y}}_{\cdot s}\|^2$$

for every $\overline{\gamma} \geq 1$, where $\|\cdot\|$ is the Euclidean norm.

Proof: We prove this property by showing that $(Y_{ir} - Y_{is})^2 \leq (\widehat{Y}_{ir} - \widehat{Y}_{is})^2$ for all $i \in \{1,2,\ldots,n\}$. We distinguish the following four cases:

Case 1: $Y_{ir} = Y_{is} = 1$. In this case $(Y_{ir} - Y_{is})^2 = 0$, and by Fact 1, $(\widehat{Y}_{ir} - \widehat{Y}_{is})^2 = (\overline{\gamma} + 2\mathfrak{d}^-_{r,i} - \overline{\gamma} - 2\mathfrak{d}^-_{s,i})^2 = 4(\mathfrak{d}^-_{r,i} - \mathfrak{d}^-_{s,i})^2 \geq 0$.

Case 2: $Y_{ir} = Y_{is} = -1$. Even in this case $(Y_{ir} - Y_{is})^2 = 0$, whereas $(\widehat{Y}_{ir} - \widehat{Y}_{is})^2 = (-\overline{\gamma} - 2\mathfrak{d}^+_{r,i} + \overline{\gamma} + 2\mathfrak{d}^+_{s,i})^2 = 4(\mathfrak{d}^+_{s,i} - \mathfrak{d}^+_{r,i})^2 \geq 0$.

Case 3: $Y_{ir} = 1 \ \wedge\ Y_{is} = -1$. In this case, $(Y_{ir} - Y_{is})^2 = 4$, and $(\widehat{Y}_{ir} - \widehat{Y}_{is})^2 = (\overline{\gamma} + 2\mathfrak{d}^-_{r,i} + \overline{\gamma} + 2\mathfrak{d}^+_{s,i})^2 = 4(\overline{\gamma} + \mathfrak{d}^-_{r,i} + \mathfrak{d}^+_{s,i})^2$. Since both $\mathfrak{d}^+_{s,i}, \mathfrak{d}^-_{r,i} \geq 0$ and $\overline{\gamma} \geq 1$, it follows $(Y_{ir} - Y_{is})^2 \leq (\widehat{Y}_{ir} - \widehat{Y}_{is})^2$.

Case 4: $Y_{ir} = -1 \ \wedge\ Y_{is} = 1$. Again $(Y_{ir} - Y_{is})^2 = 4$, and $(\widehat{Y}_{ir} - \widehat{Y}_{is})^2 = (-\overline{\gamma} - 2\mathfrak{d}^+_{r,i} - \overline{\gamma} - 2\mathfrak{d}^-_{s,i})^2 = 4(-(\overline{\gamma} + \mathfrak{d}^+_{r,i} + \mathfrak{d}^-_{s,i}))^2$. As $\mathfrak{d}^-_{s,i}, \mathfrak{d}^+_{r,i} \geq 0$ and $\overline{\gamma} \geq 1$ we have, like the previous case, $(Y_{ir} - Y_{is})^2 \leq (\widehat{Y}_{ir} - \widehat{Y}_{is})^2$.

□

The map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ not only increases the distance between the instance-indexed label vector for two distinct tasks (as we just showed), but it also increases the distance between the task-indexed label vector for two distinct instances. Indeed, since $\overline{\boldsymbol{\mathcal{L}}}$ is positive semidefinite, it is easy to show that when $\overline{\gamma} \geq 1$ the transformation $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ increases the distance between the labelings $\boldsymbol{Y}_{i\cdot}$ and $\boldsymbol{Y}_{j\cdot}$, for any pair of distinct instances $i, j \in \{1,2,\ldots,n\}$.

We now focus our discussion on another important feature of the algorithm, which makes our multitask label propagation appropriate for tasks with very unbalanced labelings. Specifically, when most entries of each column in the label matrix $\boldsymbol{Y}$ are -1. In this case, the rows of $\boldsymbol{Y}$ also contain mostly negative entries. Accordingly, by Fact 1, we can compensate the preponderance of negatives by applying the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$. We show that with an example.

Consider the task interaction matrix $\overline{\boldsymbol{\mathcal{C}}}$ such that $\mathcal{C}_{rs} = 1$ for all $r \neq s$. That is, all tasks are strongly dissimilar to each other. Then

$$\overline{\boldsymbol{\mathcal{A}}} = \begin{bmatrix} \overline{\gamma}+m-1 & -1 & \ldots & -1 \\ -1 & \overline{\gamma}+m-1 & \ldots & -1 \\ \vdots & \vdots & \vdots & \vdots \\ -1 & \ldots & \ldots & \overline{\gamma}+m-1 \end{bmatrix} \tag{8}$$

By Fact 1, it is straightforward to prove the following.

Corollary 1. Fix $\boldsymbol{Y} \in \{-1,1\}^{n\times m}$ and the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ such that $\widehat{\boldsymbol{Y}} = \psi_{\overline{\boldsymbol{\mathcal{A}}}}(\boldsymbol{Y}) = \boldsymbol{Y}\overline{\boldsymbol{\mathcal{A}}}$, where $\overline{\boldsymbol{\mathcal{A}}}$ is defined as in (8). Then, for all $i = 1,\ldots,n$ it holds that

$$\widehat{Y}_{ik} = \begin{cases} \overline{\gamma} + 2|\mathcal{N}_i| & \text{if } Y_{ik} = +1 \\ -\overline{\gamma} - 2|\mathcal{P}_i| & \text{if } Y_{ik} = -1. \end{cases}$$

Corollary 1 shows that, when $|\mathcal{P}_i| \ll |\mathcal{N}_i| = m - |\mathcal{P}_i|$ (that is, the multitask labeling for vertex i is unbalanced towards negatives), the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ assigns to positives ($Y_{ik} = +1$) an absolute value higher than that assigned to negatives ($Y_{ik} = -1$). An analogous behaviour characterizes our method when a generic matrix $\overline{\boldsymbol{\mathcal{C}}}$ is considered, as stated in Fact 1. This simple property allows the rare positive labels to propagate in the graph. This is unlike the standard LP algorithm, where negative vertices are easily overwhelmed by the positive vertices during the label propagation process. The toy example in Figure 1 shows that the application of the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$, where $\overline{\boldsymbol{\mathcal{A}}}$ is defined as in (8), allows to improve the final classification of vertices. These observations are empirically confirmed in Section 4.

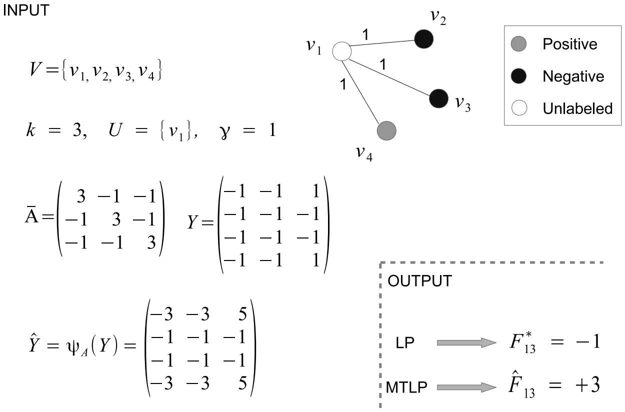

INPUT
$V = \{v_1, v_2, v_3, v_4\}$
$k = 3, \quad U = \{v_1\}, \quad \gamma = 1$
$\bar{\mathrm{A}} = \begin{pmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{pmatrix}$
$Y = \begin{pmatrix} -1 & -1 & 1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & 1 \end{pmatrix}$
$\hat{Y} = \psi_A(Y) = \begin{pmatrix} -3 & -3 & 5 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ -3 & -3 & 5 \end{pmatrix}$
v_1
v_2
v_3
v_4
1
1
1
● Positive
● Negative
○ Unlabeled
OUTPUT
LP
$F^*_{13} = -1$
MTLP
$\hat{F}_{13} = +3$

Fig. 1. Toy example with four vertices $v_1, \ldots v_4$, labeled for three tasks according to the matrix $\boldsymbol{Y}$. The test point is instance v_1 in all the tasks, and we apply LP and MTLP to predict it. For tasks 1 and 2 both methods correctly associate v_1 with a negative label. However, in the third task, only MTLP correctly predicts a positive label for v_1.

3.2.2 Task similarities

While MTLP and MTLP-inv are designed to work with any task matrix, similarity and dissimilarity measures are typically tailored to specific domains. Different tasks may share different types of similarities, or may be organized in a hierarchy with a specific structure —such as a tree or a directed acyclic graph— where the positive instances of the children tasks are subsets of the positive instances of their parent tasks. In the case of a hierarchy, different approaches for computing the task matrix are possible: considering only the structure of the hierarchy [40], [41], or combining the hierarchical information with the information content of the tasks [42].

In this work we consider two dissimilarity measures (diss_0 and diss_3) and three similarity measures ($\mathrm{sim}_1, \mathrm{sim}_2, \mathrm{sim}_3$). The similarity measures sim_1 and sim_2 were introduced by Jiang [43] and Lin [44], respectively. Both measures are derived from the dissimilarity measure diss_0, whose definition requires a hierarchy over the tasks. The dissimilarity diss_3 is computed directly from the similarity sim_3, which does not require any hierarchical information.

When tasks are organized in a hierarchy, we denote by $\mathrm{anc}(k) \subset \{1,\ldots,m\}$ the set of ancestor tasks of task k in the hierarchy. Moreover, we use $\nu(k)$ to denote the frequency of positive instances for task k. Since a positive instance for a task k is also positive for any $r \in \mathrm{anc}(k)$, it holds that $\nu(k) \leq \nu(r)$. Finally, we denote by $\mathrm{MA}(k,r)$ the common ancestor of tasks k and r whose frequency $\nu(\mathrm{MA}(k,r))$ is the lowest among all ancestors of k and r.

Let $-\log(\nu(k))$ be the information content of task k. We start by recalling the hierarchical dissimilarity measure introduced in [44],

$$\mathrm{diss}_0(k,r) = -\log\nu(k) - \log\nu(r) + 2\log\nu(\mathrm{MA}(k,r)) \;.$$

This is the sum of the information content of k and r minus the information content of their closest common ancestor $\mathrm{MA}(k,r)$. Note that diss_0 is always positive, as $\mathrm{MA}(k,r) \geq \max\{\nu(k),\nu(r)\}$. The two hierarchical similarity measures associated with diss_0 are defined as follows.

Jiang similarity measure:

$$\mathrm{sim}_1(k,r) = \frac{1}{1+\mathrm{diss}_0(k,r)} \;.$$

Lin similarity measure:

$$\mathrm{sim}_2(k,r) = \frac{2\log\nu(\mathrm{MA}(k,r))}{\log\nu(k)+\log\nu(r)}$$

Our third similarity measure does not rely on a hierarchy of tasks. Let $P^{(k)}$ the set of instances that are positive for the task k.

Information content measure:

$$\mathrm{sim}_3(k,r) = \begin{cases} \dfrac{\left|P^{(k)} \cap P^{(r)}\right|}{\left|P^{(k)} \cup P^{(r)}\right|} & \text{if } P^{(k)} \cup P^{(r)} \neq \emptyset \\ \\ 0 & \text{otherwise.} \end{cases}$$

This is the ratio between the number of examples that are positive for both tasks and the number of examples that are positive for at least one task. The higher the number of shared positive examples, the higher the similarity (up to 1). When two tasks do not share any positive example, their similarity is zero. In a hierarchy of tasks, tasks with many positive examples are usually closer to the root (less specific). In this case the denominator of sim_3 tends to reduce the similarity between the two tasks as opposed to the case in which the task have a small number of positive annotations. Indeed, sharing annotations between two specific tasks (closer to leaves) is more informative than sharing annotations between two more general tasks (closer to the root).

In the experiments, we compare learning with similarities $\mathrm{sim}_1(k,r)$ and $\mathrm{sim}_2(k,r)$ against learning with the dissimilarity $\mathrm{diss}_0(k,r)$. We also compare learning with $\mathrm{sim}_3(k,r)$ against $\mathrm{diss}_3(k,r) = 1 - \mathrm{sim}_3(k,r)$. For each one of the similarity/dissimilarity measures defined above, we set $\mathcal{C}_{kr} = \mathrm{sim}(k,r)$ and $\overline{\mathcal{C}}_{kr} = \mathrm{diss}(k,r)$ (where necessary, values are normalized so that all matrix entries lie in the range $[0,1]$).

4 Results and Discussion

In this section we evaluate our multitask algorithm on the prediction of the bio-molecular functions of proteins belonging to some considered model organisms. We start by describing the experimental setting. Then we compare the performance of our algorithm against that of state-of-the-art methods.

4.1 Experimental setting

4.1.1 Data

We considered three different experiments to predict the protein functions of three model organisms: *Drosophila melanogaster* (fly), *Homo sapiens* (human) and *Escherichia coli* (bacteria). Gene networks for model organisms have been downloaded from the GeneMANIA website (`www.genemania.org`), and selected in order to cover different types of data, including co-expression, genetic interactions, shared domains, and physical interactions. The selected networks are described in Tables 1, 2 and 3. For every organism, networks were integrated through unweighted sum on the union of genes in the individual networks. No preprocessing was applied to the individual networks, whereas each network, denoted by the corresponding connection matrix $\boldsymbol{W}$, was normalized as follows:

$$\hat{\boldsymbol{W}} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$$

Type	Source	Nodes
Co-expression	Baradaran-Heravi et al. [45]	8857
Co-expression	Busser et al. [46]	8857
Co-expression	Colombani et al. [47]	8857
Co-expression	Lundberg et al. [48]	8857
Genetic interactions	BioGRID [49]	929
Genetic interactions	Yu et al. [50]	1414
Physical interactions	Guruharsha et al. A [51]	1866
Physical interactions	Guruharsha et al. B [51]	3833
Physical interactions	BioGRID [49]	558
Shared protein domains	InterPro [52]	5627

TABLE 1
Fly networks.

Type	Source	Nodes
Co-expression	Bahr et al. [53]	7611
Co-expression	Balgobind et al. [54]	17522
Co-expression	Bigler et al. [55]	17522
Co-expression	Botling et al. [56]	17522
Co-expression	Clarke et al. [57]	17458
Co-expression	Vallat et al. [58]	17521
Common biological pathways	PATHWAYCOMMONS [59]	2133
Common biological pathways	Wu et al. [60]	5319
Physical interactions	BioGRID [49]	15800
Physical interactions	iRref-GRID [61]	9403
Physical interactions	iRref-HPRD [61]	9403
Physical interactions	iRref-OPHID [61]	9403
Physical interactions	IREF SMALL-SCALE-STUDIES [61]	9036
Shared protein domains	InterPro [52]	15800
Shared protein domains	Pfam [62]	15251

TABLE 2
Human networks.

Type	Source	Nodes
Co-expression	Graham et al. [63]	3959
Co-expression	Robbins-Manke et al. [64]	3912
Genetic interactions	Babu et al. [65]	715
Genetic interactions	Butland et al. [66]	3497
Physical interactions	Hu at al [67]	1537
Physical interactions	IREF-Dip [61]	633
Physical interactions	Y2H - PPI	1063
Shared protein domains	InterPro [52]	3005
Shared protein domains	Pfam [62]	2726

TABLE 3
Bacteria networks.

where $\boldsymbol{D}$ is the diagonal matrix with diagonal entries $d_{ii} = \sum_j W_{ij}$.

Protein functions were downloaded from the Gene Ontology. This ontology is structured as a directed acyclic graph with different levels of specificity and contains three branches: *Biological Process* (BP), *Molecular Functions* (MF), and *Cellular Components* (CC). We considered the experimental annotations in the releases 07.03.16, 16.03.16, and 17.10.16 respectively for fly, human and bacteria organisms. We performed a dedicated experiment for every branch.

For predicting the most specific terms in the ontology (i.e., those best describing protein functions), and in order to consider terms with a minimum amount of prior information, we selected all the GO terms with $5 - 100$ positive annotated genes, obtaining 2657 (1742 BP, 539 MF, 376 CC), 5312 (3799 BP, 957 MF, 556 CC), and 1324 (653 BP, 610 MF, 61 CC) terms for fly, human, and bacteria, respectively. We considered two groups of GO terms according to their specificity: GO terms with 5-20 and 21-100 annotated proteins, for a total of 2 categories for every GO branch. In the end, we obtained a total of 10329 fly, 15262 human, and 4132 bacteria genes which have at least one GO positive annotation in the considered GO release. The obtained tasks are therefore severely unbalanced toward negatives.

4.1.2 Evaluation metrics

In order to evaluate the generalization performance of the compared methods, we applied a 3-fold cross-validation experimental setting and adopted the Area Under the Precision-Recall Curve (AUPRC) as "per term" ranking measure. AUPRC is indeed more informative on unbalanced settings than the classical area under the ROC curve [68]. Furthermore, following the recent CAFA2 international challenge, we also considered a "protein-centric evaluation" to assess performance accuracy in predicting all ontological terms associated with a given protein sequence [9]. In this scenario, the multiple-label F-score is used as performance measure. More precisely, if we indicate as $\mathrm{TP}_j(t)$, $\mathrm{TN}_j(t)$ and $\mathrm{FP}_j(t)$ respectively the number of true positives, true negatives, and false positives for the protein j at threshold t, we can define the "per-protein" multiple-label precision $\mathrm{Prec}(t)$ and recall $\mathrm{Rec}(t)$ at a given threshold t as:

$$\mathrm{Prec}(t) = \frac{1}{n}\sum_{j=1}^{n}\frac{\mathrm{TP}_j(t)}{\mathrm{TP}_j(t) + \mathrm{FP}_j(t)}$$

$$\mathrm{Rec}(t) = \frac{1}{n}\sum_{j=1}^{n}\frac{\mathrm{TP}_j(t)}{\mathrm{TP}_j(t) + \mathrm{FN}_j(t)}$$

where n is the number of proteins. In other words, $\mathrm{Prec}(t)$ (resp., $\mathrm{Rec}(t)$) is the average multilabel precision (resp., recall) across proteins. The multilabel F-measure depends on t and according to CAFA2 experimental setting, the maximum achievable F-score ($F_{\max}$) is adopted as the main multilabel "per-protein" metric:

$$F_{\max} = \max_t \frac{2\mathrm{Prec}(t)\mathrm{Rec}(t)}{\mathrm{Prec}(t) + \mathrm{Rec}(t)} \tag{9}$$

4.2 Results

4.2.1 Evaluating GO semantic similarities

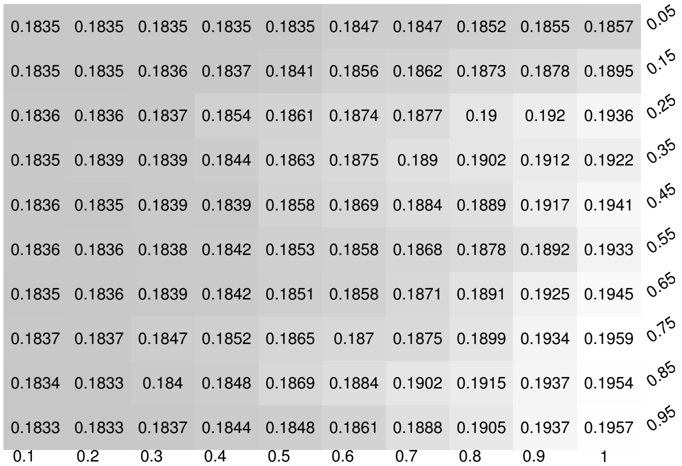
This section investigates the impact of the task similarity/dissimilarity measures described in Section 3.2.2 on the performance of the proposed multitask label propagation algorithms. Table 4 shows the obtained results. In this experiment we set $\gamma = \overline{\gamma} = 1$ (the choice of parameter $\overline{\gamma}$ is discussed in Section 4.2.5). When MTLP-inv uses the similarity measures $\mathrm{sim}_1, \mathrm{sim}_2$ and MTLP uses diss_0 for MTLP, MTLP outperforms MTLP-inv in both AUPRC and $F_{\max}$. Nevertheless, the GO term similarity sim_3 is much more informative for MTLP-inv, which achieves in this case results competitive with MTLP (whose performance instead is nearly indistinguishable when using diss_0 or diss_3), and in some cases even better. The difference in favor of MTLP seems to increase with the data imbalance: on *human* data set, the most unbalanced, we observe the highest gap in favor of MTLP; whereas on the *Bacteria* data set, the least unbalanced, the gap is reduced and —in some cases like for the MF terms— MTLP-inv significantly outperforms MTLP in terms of average AUPRC. In terms of $F_{\max}$, however, MTLP is always the top method.

Overall, these results suggests that MTLP should be preferred when the proportion of positives is drastically smaller than that of negatives. When data are more balanced, MTLP-inv better exploits the similarities among tasks and, at least in term of AUPRC, is a valid option. In terms of multilabel accuracy, is always better than MTLP-inv. Finally, it is worth noting that both methods outperforms LP in terms of AUPRC (see Section 4.2.3 for LP results), whereas in terms of $F_{\max}$ only MTLP achieves better results than LP. In order to investigate the reasons why, unlike MTLP-inv, MTLP performance slightly varies with the task dissimilarity measure, we run MTLP on the fly organism and CC tasks by randomly generating the matrix $\overline{\boldsymbol{\mathcal{C}}}$. We generated matrices with different sparsity (from 5% to 95%, with steps of 10%) and with different ranges of weight values. Specifically, we uniformly selected weights in the interval $[0, \tau]$, with τ ranging from 0.1 to 1, by steps of 0.1. In Figure 2, we show the heatmap of the average AUPRC obtained in each experiment. As expected, the results are considerably worse than those obtained when considering real dissimilarity matrices (see Table 4). There is a small

METHODS	BP				MF				CC			
	All	5-20	21-100	$F_{\max}$	All	5-20	21-100	$F_{\max}$	All	5-20	21-100	$F_{\max}$
FLY												
MTLP diss_0	**0.140**	**0.133**	**0.153**	**0.247**	**0.333**	**0.322**	**0.355**	**0.411**	**0.262**	**0.265**	**0.253**	0.354
MTLP diss_3	**0.140**	**0.133**	**0.153**	0.246	**0.333**	**0.322**	**0.355**	0.410	**0.262**	**0.265**	**0.253**	**0.357**
MTLP-inv sim_1	0.020	0.013	0.031	0.183	0.198	0.179	0.238	0.374	0.150	0.138	0.181	0.306
MTLP-inv sim_2	0.020	0.014	0.031	0.170	0.192	0.172	0.235	0.351	0.101	0.082	0.147	0.259
MTLP-inv sim_3	0.135	0.129	0.146	0.244	0.328	0.318	0.352	0.381	0.261	0.265	0.251	0.333
HUMAN												
MTLP diss_0	0.144	0.133	**0.165**	0.273	0.248	0.247	**0.250**	0.383	**0.224**	**0.259**	0.156	0.317
MTLP diss_3	**0.145**	**0.134**	**0.165**	**0.275**	**0.249**	**0.248**	**0.250**	**0.385**	**0.224**	**0.259**	0.156	**0.318**
MTLP-inv sim_1	0.008	0.005	0.014	0.200	0.093	0.083	0.152	0.330	0.105	0.113	0.090	0.274
MTLP-inv sim_2	0.008	0.005	0.012	0.182	0.059	0.050	0.079	0.294	0.066	0.064	0.068	0.223
MTLP-inv sim_3	0.139	0.129	0.159	0.244	0.243	0.241	0.244	0.355	0.220	0.256	**0.160**	0.299
BACTERIA												
MTLP diss_0	0.119	0.107	0.169	0.210	0.173	0.157	0.238	0.269	0.122	0.105	0.220	**0.348**
MTLP diss_3	0.119	0.107	0.168	**0.212**	0.173	0.157	0.238	**0.276**	0.122	0.105	**0.219**	**0.348**
MTLP-inv sim_1	0.069	0.056	0.123	0.181	0.106	0.092	0.165	0.235	0.101	0.086	0.187	0.246
MTLP-inv sim_2	0.053	0.043	0.094	0.109	0.057	0.045	0.107	0.117	0.106	0.089	0.207	0.281
MTLP-inv sim_3	**0.121**	**0.108**	**0.176**	0.189	**0.181**	**0.165**	**0.247**	0.247	**0.123**	**0.107**	0.212	0.289

TABLE 4

Comparison according to average AUPRC and multilabel F-measure ($F_{\max}$) between MTLP and MTLP-inv using the semantic similarity measures described in Section 3.2.2. Column All is the average across all GO terms, column 5-20 is the average across GO terms with at most 20 positive genes, and column 21-100 is the average across terms with more than 20 positives. Best results are in boldface. Results are underlined when the difference between MTLP and MTLP-inv is statistically significant (Wilcoxon signed rank test, $p\text{-}value < 0.05$).



0.1835	0.1835	0.1835	0.1835	0.1835	0.1847	0.1847	0.1852	0.1855	0.1857	0.05
0.1835	0.1835	0.1836	0.1837	0.1841	0.1856	0.1862	0.1873	0.1878	0.1895	0.15
0.1836	0.1836	0.1837	0.1854	0.1861	0.1874	0.1877	0.19	0.192	0.1936	0.25
0.1835	0.1839	0.1839	0.1844	0.1863	0.1875	0.189	0.1902	0.1912	0.1922	0.35
0.1836	0.1835	0.1839	0.1839	0.1858	0.1869	0.1884	0.1889	0.1917	0.1941	0.45
0.1836	0.1836	0.1838	0.1842	0.1853	0.1858	0.1868	0.1878	0.1892	0.1933	0.55
0.1835	0.1836	0.1839	0.1842	0.1851	0.1858	0.1871	0.1891	0.1925	0.1945	0.65
0.1837	0.1837	0.1847	0.1852	0.1865	0.187	0.1875	0.1899	0.1934	0.1959	0.75
0.1834	0.1833	0.184	0.1848	0.1869	0.1884	0.1902	0.1915	0.1937	0.1954	0.85
0.1833	0.1833	0.1837	0.1844	0.1848	0.1861	0.1888	0.1905	0.1937	0.1957	0.95
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	

Fig. 2. Average AUPRC values achieved by MTLP method of *fly* data and CC GO terms when the matrix $\overline{\mathcal{C}}$ is randomly generated. Values of τ are reported on the columns, whereas row labels show the proportion of nonzero entries in the generated matrix. The lighter the color, the larger the corresponding AUPRC value.

AUPRC variation from the different random data, with higher AUPRC when the dissimilarity matrix is denser and with larger entries (the former seems to affect the results more than the latter). This is consistent with Fact 1, since the lower the weight and/or the sparser the matrix, the closer MTLP is to LP. Finally, on randomly generated dissimilarity matrices MTLP performs even worse than LP, as we can see from Figure 4.

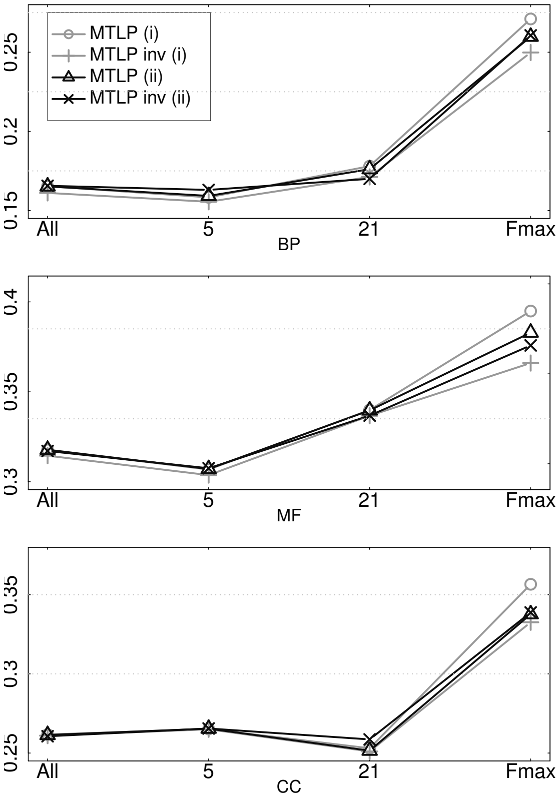


Fig. 3. Average AUPRC performance across all GO terms (All), across GO terms with at most 20 positive instances (5), and across terms with more than 20 positives (21).

4.2.2 *Grouping GO terms for multitask mapping*

Following the approach proposed in [4], in addition to the strategy grouping GO terms by branch (i) adopted in the previous section, we have examined an alternative way for grouping the terms to be considered in the multitask map (6) when running MTLP algorithm. Specifically, we grouped GO terms not just by GO branch (BP, MF, and CC), but also by taking into account the number of annotated proteins (ii), obtaining 6 groups: BP with 5-20 (1119 terms) and 21-100 (623 terms) annotations, MF 5-20 (362 terms), 21-100 (177 terms), and CC 5-20 (267 terms), 21-100 (109 terms). The corresponding results on *fly* data are reported in Figure 3. AUPRC results show negligible differences between strategies (i) and (ii), for both MTLP and MTLP-inv. More clear is the difference in terms of *Fmax*, with opposite behaviour between MTLP and MTLP-inv: MTLP has worse performance in all GO branches; MTLP-inv instead tends to perform better (see for instance MF results). Indeed, black lines (grouping strategy (ii)) in correspondence of *Fmax* are always between grey lines (grouping strategy (i)). However, the best results are still achieved by MTLP when grouping terms by GO branch, and accordingly we consider this strategy in the rest of the paper.

4.2.3 *Prediction of GO functions for fly, human, and bacteria organisms*

MTLP ($\overline{\gamma} = 1$) was compared with state-of-the-art graph-based methodologies applied to the prediction of protein functions. We considered: *LP*, the label propagation algorithm described in Section 3.1; *COSNetM* [15], an extension of a node classifier designed for unbalanced settings [36]; *RW*, the classical t-step random walk algorithm [69]; *GBA*, a method based on the *guilt-by-association* assumption [23]; *MS-kNN*, one of the best methods in the recent CAFA2 challenge applying the *kNN* algorithm to each network independently, and then combining the obtained predictions [70].

In order to deal with label imbalance in LP, we applied a label normalization step before running label propagation. This step normalizes the labels of each GO term so that positive and negative labels sum to 1. In our experiments, this variant of LP performs much better than the vanilla LP algorithm. For the RW algorithm we set the limit on the number of iterations to 100, since higher values did not improve the performance while increasing the computational burden. Finally, we set to 5 the parameter k for the kNN algorithm, as a result of a tuning process on training data.

In Figures 4 and 5 we show the obtained results in terms of AUPRC and $F_{\max}$, on BP and CC terms respectively (on MF terms the methods showed a similar behaviour). Interestingly, MTLP always achieves the highest AUPRC averaged over all tasks (*All*), with statistically significant improvements over the second top method (*p-value* < 0.001), except for *bacteria* data and for BP terms on *fly* data. When comparing with LP method, the improvement is always significant, except for CC (*bacteria* data). COSNetM is the second method on *human* and *fly* data sets, while on *bacteria* LP (CC) and RW (BP) rank as second method. Furthermore, and more importantly, MTLP improvements are more noticeable on the most unbalanced terms, which are those best characterizing the biological functions of genes. GBA, MS-kNN and RW methods seem suffer the strongly unbalanced setting, and perform worse than LP, with the exception of RW on *bacteria* data set. The good performance of COSNetM in this unbalanced setting is likely due to its cost-sensitive strategy, which requires learning two model parameters. This extra learning step increases its computation time. Indeed, COSNetM takes on average around 4 seconds on a Linux machine with Intel Xeon(R) CPU 3.60GHz and 32 Gb RAM to perform an entire cross validation cycle for one task on fly data, whereas both LP and MTLP take on average slightly less than one second. This confirms our observation that applying the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ after label propagation does not increase the algorithm complexity, and just slightly increases the execution time for computing $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$.

Even in terms of *Fmax* MTLP obtains the best results, with LP second-best method (except on BP —*fly* data). This shows that our method can achieve good predictive capabilities both when predicting single GO terms and when predicting a GO multilabel for single proteins. On the other side, the compared methods tend to have competitive performance in only one scenario. For instance, RW poorly performs in terms of $F_{\max}$, whereas, unlike AUPRC, MS-kNN achieves good *Fmax* results: on BP (*fly* data) it is the best method after MTLP. Even COSNetM, which is the second method in terms of AUPRC, achieves the third or the fourth best $F_{\max}$ rank.

4.2.4 *Evaluating different powers of the Laplacian matrix*

A further experiment was carried out to analyze how MTLP performance changes when using the map $\psi_{\overline{\boldsymbol{\mathcal{A}}},p}(\boldsymbol{Y}) = \boldsymbol{Y}\overline{\boldsymbol{\mathcal{A}}}^p$ for $p \geq \frac{1}{2}$, instead of $\psi_{\overline{\boldsymbol{\mathcal{A}}}}(\boldsymbol{Y}) = \boldsymbol{Y}\overline{\boldsymbol{\mathcal{A}}}$. We empirically tested on the *fly* organism different values of p, fixing the parameter $\overline{\gamma} = 1$ and using the diss_3 measure. The results are shown in Figure 6. We considered $p = \frac{1}{2}, 2, 3, 4, 5$. Except for BP terms, where the map $\psi_{\overline{\boldsymbol{\mathcal{A}}},1/2}$ performs slightly better than $\psi_{\overline{\boldsymbol{\mathcal{A}}},1}$, all choices of $p \neq 1$ lead to worse results. In particular, the performance strongly decays for $p > 2$.

4.2.5 *Impact of parameter* $\overline{\gamma}$

Large values of the $\overline{\gamma}$ parameter, introduced in Section 3.2, tend to reduce the multitask contribution encoded in $\overline{\boldsymbol{\mathcal{A}}}$, since $\overline{\boldsymbol{\mathcal{A}}}$ is diagonally dominant and absolute labels assigned to positives and negatives vertices by the map $\psi_{\overline{\boldsymbol{\mathcal{A}}}}$ tend to be almost the same (see Fact 1). Hence, this allows to "regulate" to some extent the method between multitask and singletask label propagation. We experimentally tuned $\overline{\gamma}$ on *fly* and *human* data from 0.25 to 1.5 with step size 0.25. It turns out there is a negligible difference, with results reported in Table 4 and corresponding to $\overline{\gamma} = 1$. This is expected, since m is much larger than 1 in the considered experiments. For this reason, we also performed another experiment in which we selected a smaller subset of terms in the BP branch (a similar trend is observed for the MF and CC branches). Specifically, we ran our algorithm on a subset of 42 terms for the *fly* organism, by varying $\overline{\gamma}$ in the specified range. The results are shown in Table 5. Confirming our observations, our method is more sensitive to $\overline{\gamma}$ values in this setting, and the overall trend is that the average AUPRC tends to decrease when $\overline{\gamma}$ becomes larger (similarly to $F_{\max}$). This not

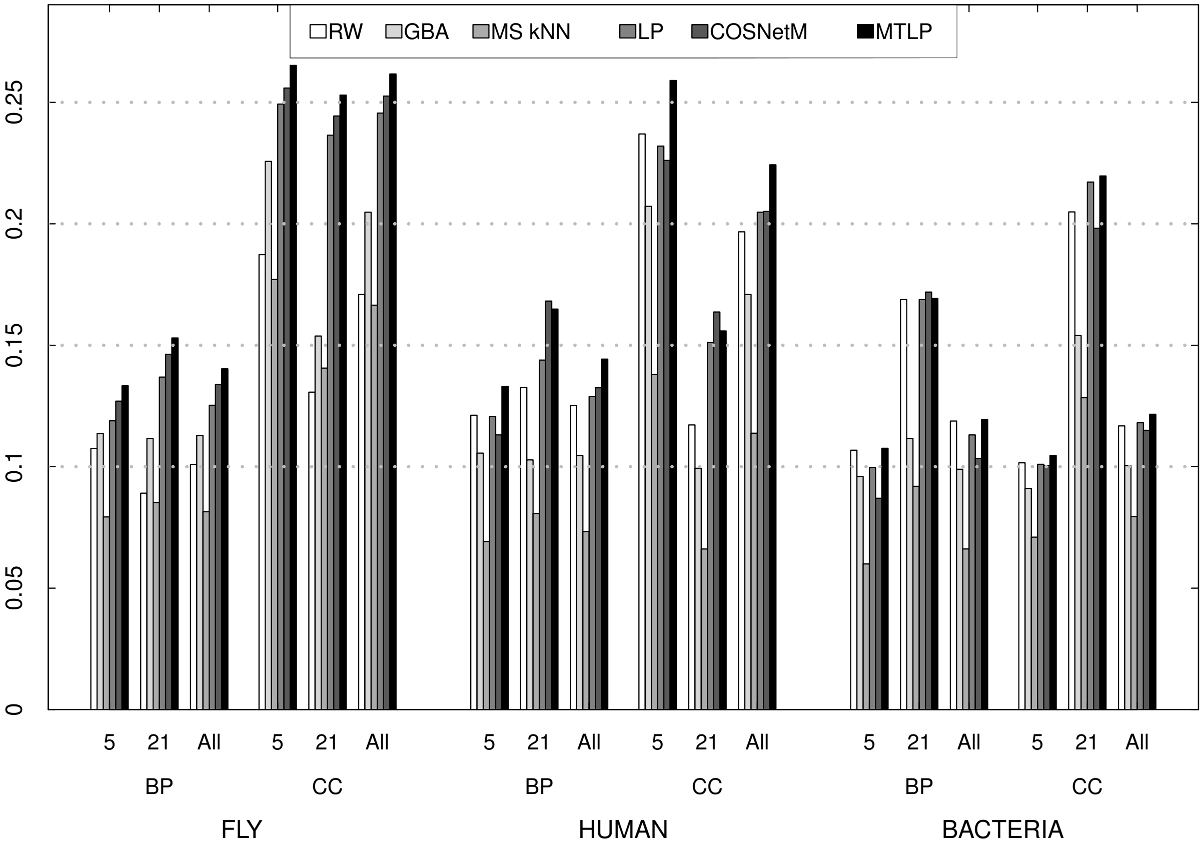


Fig. 4. Average AUPRC performance across all GO terms (All), across GO terms with at most 20 positive instances (5), and across terms with more than 20 positives (21).

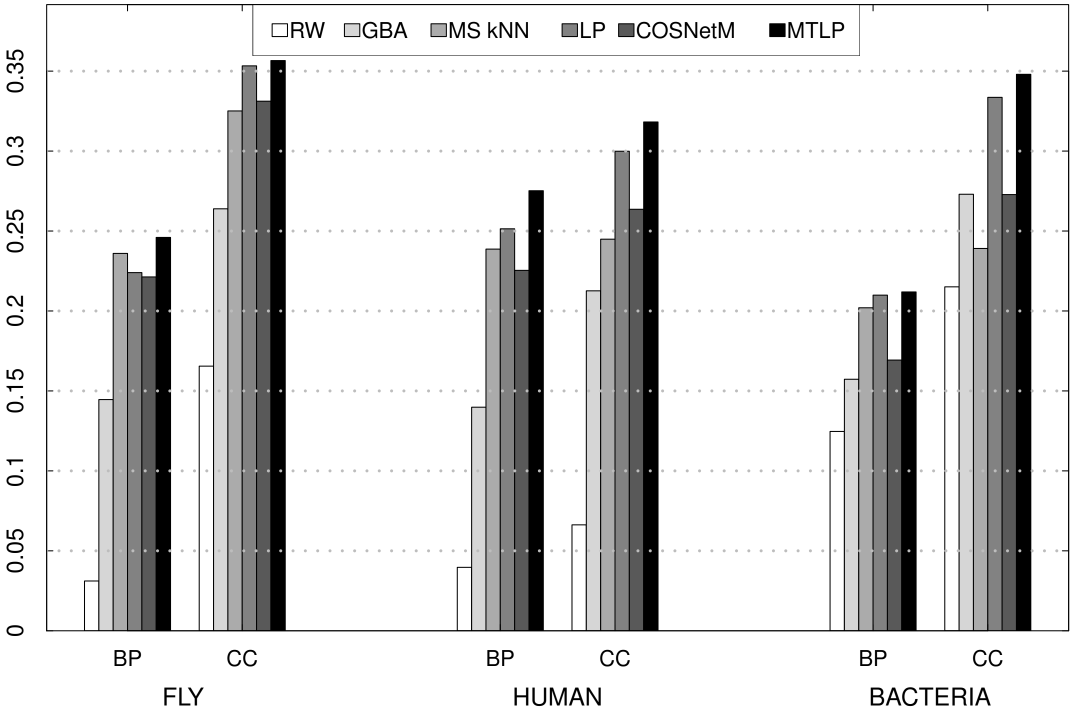


Fig. 5. Average multi-label F-measure performance across all GO terms.

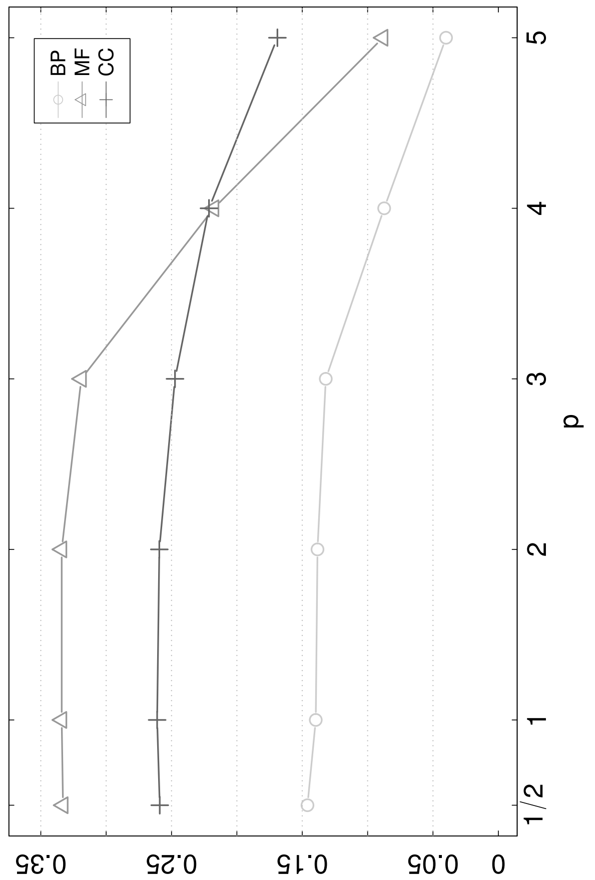


Fig. 6. Average AUPRC values achieved by MTLP of *fly* data with different values of the parameter p.

$\overline{\gamma}$	All	5−20	20−100
0.25	0.158	0.150	0.182
0.5	0.157	0.151	0.177
0.75	0.145	0.134	0.178
1	0.144	0.133	0.178
1.25	0.140	0.130	0.175
1.5	0.139	0.129	0.174

TABLE 5
AUPRC of the MTLP method ($p = 1$, task similarity measure diss_3 averaged across 42 selected MF GO terms for *human* data by varying the parameter $\overline{\gamma}$. Column All is the average across all tasks, column 5-20 is the average across terms with at most 20 annotations, and column 21-100 is the average across terms with more than 20 positives.

surprising: as we explained, with large values of $\overline{\gamma}$ MTLP behaves closer to LP, whose results are lower in this setting.

5 Conclusions

We have shown that task relatedness information represented through task dissimilarity is better suited for label propagation in unbalanced protein function prediction than task similarity. The proposed multitask label propagation algorithm compared favourably with the state-of-the-art methodologies for protein function prediction on three model organisms. Although we gained some intuition and collected empirical evidence, we are still invesigating the multitask problems where our approach is most effective. Specifically, it would be useful to investigate whether dissimilarity information helps when coupled with multitask algorithms different from label propagation. For example, linear learning algorithms such as SVM or Perceptron. Laplacian spectral theory is also likely to help us shed some further light on the properties of our method.

Acknowledgments

The authors would like to thank the reviewers of BIOKDD16 for useful comments on an earlier draft of this paper.

References

[1] N. Youngs, D. Penfold-Brown, K. Drew, D. Shasha, and R. Bonneau, “Parametric bayesian priors and better choice of negative examples improve protein function prediction,” *Bioinformatics*, vol. 29, no. 9, pp. 1190–1198, 2013.

[2] M. Frasca and D. Malchiodi, “Selection of negative examples for node label prediction through fuzzy clustering techniques,” in *Advances in Neural Networks: Computational Intelligence for ICT*, S. Bassis, A. Esposito, C. F. Morabito, and E. Pasero, Eds. Cham: Springer International Publishing, 2016, pp. 67–76. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-33747-0_7

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

[4] S. Mostafavi and Q. Morris, “Fast integration of heterogeneous data sources for predicting gene function with limited annotation,” *Bioinformatics*, vol. 26, no. 14, pp. 1759–1765, 2010.

[5] M. Frasca, A. Bertoni *et al.*, “UNIPred: unbalance-aware Network Integration and Prediction of protein functions,” *Journal of Computational Biology*, vol. 22, no. 12, pp. 1057–1074, 2015.

[6] The Gene Ontology Consortium, “Gene ontology: tool for the unification of biology,” *Nature Genet.*, vol. 25, pp. 25–29, 2000.

[7] A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and H. Mewes, “The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes,” *Nucleic Acids Research*, vol. 32, no. 18, pp. 5539–5545, 2004.

[8] P. Radivojac *et al.*, “A large-scale evaluation of computational protein function prediction,” *Nature Methods*, vol. 10, no. 3, pp. 221–227, 2013.

[9] Y. Jiang, T. R. Oron *et al.*, “An expanded evaluation of protein function prediction methods shows an improvement in accuracy,” *Genome Biology*, vol. 17, no. 1, p. 184, 2016. [Online]. Available: http://dx.doi.org/10.1186/s13059-016-1037-6

[10] D. Martin, M. Berriman, and G. Barton, “Gotcha: a new method for prediction of protein function assessed by the annotation of seven genomes.” *BMC Bioinformatics*, vol. 5, p. 178, 2004.

[11] T. Hawkins, M. Chitale, S. Luban *et al.*, “Pfp: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data.” *Proteins*, vol. 74, no. 3, pp. 566–82, 2009.

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

[13] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani, “Global protein function prediction from protein-protein interaction networks,” *Nature Biotechnology*, vol. 21, pp. 697–700, 2003.

[14] R. Sharan, I. Ulitsky, and R. Shamir, “Network-based prediction of protein function,” *Mol. Sys. Biol.*, vol. 8, no. 88, 2007.

[15] M. Frasca, “Automated gene function prediction through gene multifunctionality in biological networks,” *Neurocomputing*, vol. 162, pp. 48 – 56, 2015.

[16] A. Sokolov and A. Ben-Hur, “Hierarchical classification of Gene Ontology terms using the GOstruct method,” *Journal of Bioinformatics and Computational Biology*, vol. 8, no. 2, pp. 357–376, 2010.

[17] A. Sokolov, C. Funk, K. Graim, K. Verspoor, and A. Ben-Hur, "Combining heterogeneous data sources for accurate functional annotation of proteins," *BMC Bioinformatics*, vol. 14, no. Suppl 3:S10, 2013.

[18] G. Obozinski, G. Lanckriet, C. Grant, J. M., and W. Noble, "Consistent probabilistic output for protein function prediction," *Genome Biology*, vol. 9, no. S6, 2008.

[19] Y. Guan, C. Myers, D. Hess, Z. Barutcuoglu, A. Caudy, and O. Troyanskaya, "Predicting gene function in a hierarchical context with an ensemble of classifiers," *Genome Biology*, vol. 9, no. S2, 2008.

[20] G. Valentini, "Hierarchical Ensemble Methods for Protein Function Prediction," *ISRN Bioinformatics*, vol. 2014, no. Article ID 901419, pp. 1–34, 2014.

[21] E. Marcotte, M. Pellegrini, M. Thompson, T. Yeates, and D. Eisenberg, "A combined algorithm for genome-wide prediction of protein function," *Nature*, vol. 402, pp. 83–86, 1999.

[22] S. Oliver, "Guilt-by-association goes global," *Nature*, vol. 403, pp. 601–603, 2000.

[23] B. Schwikowski, P. Uetz, and S. Fields, "A network of protein-protein interactions in yeast." *Nature biotechnology*, vol. 18, no. 12, pp. 1257–1261, Dec. 2000.

[24] Y. Li and J. Patra, "Integration of multiple data sources to prioritize candidate genes using discounted rating systems," *BMC Bioinformatics*, vol. 11, no. Suppl I:S20, 2010.

[25] P. Bogdanov and A. Singh, "Molecular function prediction using neighborhood features," *IEEE ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 2, pp. 208–217, 2011.

[26] X. Zhu *et al.*, "Semi-supervised learning with gaussian fields and harmonic functions," in *Proc. of the 20th Int. Conf. on Machine Learning*, Washintgton DC, USA, 2003.

[27] H. Zhou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society B*, vol. 67, no. 2, pp. 301–320, 2007.

[28] M. Szummer and T. Jaakkola, "Partially labeled classification with markov random walks," in *NIPS 2001*, vol. 14, Whistler BC, Canada, 2001.

[29] A. Azran, "The rendezvous algorithm: Multi- class semi-supervised learning with Markov random walks," in *Proceedings of the 24th International Confer- ence on Machine Learning (ICML)*, 2007.

[30] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh, "Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps," *Bioinformatics*, vol. 21, no. S1, pp. 302–310, 2005.

[31] G. Valentini, G. Armano, M. Frasca, J. Lin, M. Mesiti, and M. Re, "RANKS: a flexible tool for node label ranking and classification in biological networks," *Bioinformatics*, 2016, in press. Accepted on 22 April 2016.

[32] M. Deng, T. Chen, and F. Sun, "An integrated probabilistic model for functional prediction of proteins," *J. Comput. Biol.*, vol. 11, pp. 463–475, 2004.

[33] K. Tsuda, H. Shin, and B. Scholkopf, "Fast protein classification with multiple networks," *Bioinformatics*, vol. 21, no. Suppl 2, pp. ii59–ii65, 2005.

[34] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris, "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function," *Genome Biology*, vol. 9, no. S4, 2008.

[35] U. Karaoz *et al.*, "Whole-genome annotation by using evidence integration in functional-linkage networks," *Proc. Natl Acad. Sci. USA*, vol. 101, pp. 2888–2893, 2004.

[36] A. Bertoni, M. Frasca, and G. Valentini, *COSNet: A Cost Sensitive Neural Network for Semi-supervised Learning in Graphs*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 219–234. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23780-5_24

[37] M. Frasca, S. Bassis, and G. Valentini, "Learning node labels with multi-category hopfield networks," *Neural Computing and Applications*, vol. 27, no. 6, pp. 1677–1692, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00521-015-1965-1

[38] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, pp. 1373–1396, 2002.

[39] B. Kveton, M. Valko, A. Rahimi, and L. Huang, "Semi-supervised learning with max-margin graph cuts." in *AISTATS*, ser. JMLR Proceedings, Y. W. Teh and D. M. Titterington, Eds., vol. 9. JMLR.org, 2010, pp. 421–428. [Online]. Available: http://dblp.uni-trier.de/db/journals/jmlr/jmlrp9.html#KvetonVRH10

[40] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 1994, pp. 133–138. [Online]. Available: http://portal.acm.org/citation.cfm?id=981751

[41] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," in *MIT Press*, C. Fellfaum, Ed., Cambridge, Massachusetts, 1998, pp. 265–283.

[42] L. Meng, R. Huang, and J. Gu, "A review of semantic similarity measures in wordnet," *International Journal of Hybrid Information Technology*, vol. 6, no. 1, pp. 1–12, 2013.

[43] J. J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," in *International Conference Research on Computational Linguistics (ROCLING X)*, Sep. 1997, pp. 9008+. [Online]. Available: http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=1997cmp.lg....9008J

[44] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ser. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 296–304. [Online]. Available: http://dl.acm.org/citation.cfm?id=645527.657297

[45] A. Baradaran-Heravi, K. S. Cho, B. Tolhuis *et al.*, "Penetrance of biallelic SMARCAL1 mutations is associated with environmental and genetic disturbances of gene expression," *Human Molecular Genetics*, vol. 21, no. 11, pp. 2572–2587, Jun. 2012.

[46] B. W. Busser, L. Shokri, S. A. Jeager *et al.*, "Molecular mechanism underlying the regulatory specificity of a Drosophila homeodomain protein that specifies myoblast identity." *Development (Cambridge, England)*, vol. 139, no. 6, pp. 1164–1174, Mar. 2012.

[47] J. Colombani, D. S. Andersen, and P. Lopold, "Secreted peptide dilp8 coordinates drosophila tissue growth with developmental timing," *Science*, vol. 336, no. 6081, pp. 582–585, 2012.

[48] L. E. Lundberg, M. Fiqueiredo, P. Stenberg *et al.*, "Buffering and proteolysis are induced by segmental monosomy in Drosophila melanogaster," *Nucleic Acids Research*, Mar. 2012.

[49] C. Stark, B. joe Breitkreutz, T. Reguly *et al.*, "Biogrid: a general repository for interaction datasets." *Nucleic Acids Research*, no. Database-Issue, pp. 535–539, 2006.

[50] J. Yu, S. Pacifico, G. Liu *et al.*, "DroID: the Drosophila Interactions Database, a comprehensive resource for annotated gene and protein interactions," *BMC Genomics*, vol. 9, no. 1, pp. 461+, Oct. 2008.

[51] K. G. Guruharsha, J. Rual, B. Zhai *et al.*, "A Protein Complex Network of Drosophila melanogaster," *Cell*, vol. 147, no. 3, pp. 690–703, Oct. 2011.

[52] R. Apweiler, T. K. Attwood, A. Bairoch *et al.*, "The InterPro database, an integrated documentation resource for protein families, domains and functional sites," *Nucleic Acids Research*, vol. 29, no. 1, pp. 37–40, Jan. 2001.

[53] T. M. Bahr, G. J. Hughes *et al.*, "Peripheral Blood Mononuclear Cell Gene Expression in Chronic Obstructive Pulmonary Disease," *American Journal of Respiratory Cell and Molecular Biology*, vol. 49, no. 2, pp. 316–323, 2013.

[54] B. V. Balgobind, M. M. Van den Heuvel-Eibrink *et al.*, "Evaluation of gene expression signatures predictive of cytogenetic and molecular subtypes of pediatric acute myeloid leukemia," *Haematologica*, vol. 96, no. 2, pp. 221–230, 2011.

[55] J. Bigler, H. A. Rand *et al.*, "Cross-study homogeneity of psoriasis gene expression in skin across a large expression range," *PLoS ONE*, vol. 8, no. 1, pp. 1–15, 01 2013.

[56] J. Botling, K. Edlund, M. Lohr, B. Hellwig, L. Holmberg, M. Lambe, A. Berglund, S. Ekman, M. Bergqvist, F. Pontn, A. Knig, O. Fernandes, M. Karlsson, G. Helenius, C. Karlsson, J. Rahnenfhrer, J. G. Hengstler, and P. Micke, "Biomarker discovery in nonsmall cell lung cancer: Integrating gene expression profiling, meta-analysis, and tissue microarray validation," *Clinical Cancer Research*, vol. 19, no. 1, pp. 194–204, 2013. [Online]. Available: http://clincancerres.aacrjournals.org/content/19/1/194.abstract

[57] C. Clarke, S. F. Madden *et al.*, "Correlating transcriptional networks to breast cancer survival: a large-scale coexpression analysis," *Carcinogenesis*, vol. 34, no. 10, pp. 2300–2308, 2013. [Online]. Available: http://carcin.oxfordjournals.org/content/34/10/2300.abstract

[58] L. Vallat, C. A. Kemper *et al.*, "Reverse-engineering the genetic circuitry of a cancer cell with predicted intervention in chronic lymphocytic leukemia," *Proceedings of the National Academy of*

Sciences, vol. 110, no. 2, pp. 459–464, 2013. [Online]. Available: http://www.pnas.org/content/110/2/459.abstract

[59] E. G. Cerami, B. E. Gross *et al.*, "Pathway commons, a web resource for biological pathway data," *Nucleic Acids Research*, vol. 39, no. suppl 1, pp. D685–D690, 2011.

[60] G. Wu, X. Feng, and L. Stein, "A human functional protein interaction network and its application to cancer data analysis," *Genome Biology*, vol. 11, no. 5, pp. 1–23, 2010.

[61] S. Razick, G. Magklaras, and I. M. Donaldson, "irefindex: A consolidated protein interaction database with provenance," *BMC Bioinformatics*, vol. 9, no. 1, pp. 1–19, 2008. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-9-405

[62] R. D. Finn *et al.*, "The pfam protein families database: towards a more sustainable future," *Nucleic Acids Research*, vol. 44, no. D1, pp. D279–D285, 2016. [Online]. Available: http://nar.oxfordjournals.org/content/44/D1/D279.abstract

[63] A. I. Graham, G. Sanguinetti, N. Bramall, C. W. McLeod, and R. K. Poole, "Dynamics of a starvation-to-surfeit shift: a transcriptomic and modelling analysis of the bacterial response to zinc reveals transient behaviour of the fur and soxs regulators," *Microbiology*, vol. 158, no. 1, pp. 284–292, 2012.

[64] J. L. Robbins-Manke, Z. Z. Zdraveski, M. Marinus, and J. M. Essigmann, "Analysis of global gene expression and double-strand-break formation in dna adenine methyltransferase- and mismatch repair-deficient escherichia coli," *Journal of bacteriology*, vol. 187, no. 20, pp. 7027–7037, October 2005. [Online]. Available: http://europepmc.org/articles/PMC1251628

[65] M. Babu *et al.*, "Genetic interaction maps in escherichia coli reveal functional crosstalk among cell envelope biogenesis pathways," *PLoS Genet*, vol. 7, no. 11, pp. 1–15, 11 2011.

[66] G. Butland *et al.*, "esga: E. coli synthetic genetic array analysis," *Nat Meth*, vol. 5, no. 3, pp. 789–795, jan 2008.

[67] P. Hu, S. C. Janga *et al.*, "Global Functional Atlas of Escherichia coli Encompassing Previously Uncharacterized Proteins," *PLoS Biol*, vol. 7, no. 4, pp. e1 000 096+, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1371/journal.pbio.1000096

[68] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, pp. 1–21, 03 2015.

[69] L. Lovász, "Random walks on graphs: A survey," in *Combinatorics, Paul Erdős is Eighty*, D. Miklós, V. T. Sós, and T. Szőnyi, Eds. Budapest: János Bolyai Mathematical Society, 1996, vol. 2, pp. 353–398.

[70] L. Lan, N. Djuric, Y. Guo, and V. S., "MS-kNN: protein function prediction by integrating multiple data sources," *BMC Bioinformatics*, vol. 14, no. Suppl 3:S8, 2013.

Marco Frasca received his Ph.D. degree in Computer Science from University of Milano, Italy in 2012. He is currently a post-doc research fellow in Computer Science at the University of Milano. His research interests include the study of neural networks models for unbalanced classification problem and the development of machine learning techniques for emerging problems in life sciences, such as protein function prediction, gene-disease prioritization, and drug repositioning.

Nicolò Cesa Bianchi is professor of Computer Science at the University of Milano, Italy. He held visiting positions with UC Santa Cruz, Graz Technical University, Ecole Normale Superieure (Paris), Google, and Microsoft Research. He received a Google Research Award and a Xerox University Affairs Committee Award. His research interests include theory and applications of machine learning, sequential optimization, and algorithmic game theory. On these topics, he published two monographs: *Prediction, Learning, and Games* and *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*.