UNIPred: Unbalance-aware Network Integration and Prediction of protein functions

Marco Frasca,* Alberto Bertoni* and Giorgio Valentini* †

July 9, 2015

Abstract

The proper integration of multiple sources of data and the unbalance between annotated and unannotated proteins represent two of the main issues of the Automated Function Prediction (*AFP*) problem. Most of supervised and semi-supervised learning algorithms for *AFP* proposed in literature do not jointly consider these items, with a negative impact on both sensitivity and precision performances, due to the unbalance between annotated and unannotated proteins that characterize the majority of functional classes and to the specific and complementary information content embedded in each available source of data. We propose *UNIPred* (Unbalance-aware Network Integration and Prediction of protein functions), an algorithm that properly combines different biomolecular networks and predicts protein functions using parametric semi-supervised neural models. The algorithm explicitly takes into account the unbalance between unannotated and annotated proteins both to construct the integrated network and to predict protein annotations for each functional class. Full-genome and ontology-wide experiments with three Eukaryotic model organisms show that the proposed method compares favourably with state-of-the-art learning algorithms for *AFP*.

Key words: Protein function prediction, unbalance-aware network integration, Hopfield networks.

*DI - Department of Computer Science, University of Milan, Italy
†Corresponding author

1 Introduction

The noticeable increasing in the quantity and variety of publicly available genomic and proteomic data and the inherent difficulty and cost of experimental validation have brought to the fore the automated prediction of protein functions (*AFP*) as one of the central problems of the post-genomic era (Radivojac *et al.*, 2013).

AFP is characterized by many issues, including the possibility of assigning multiple labels to proteins (multi-label classification), the hierarchical organization of functional classes, e.g. the Gene Ontology - GO (Ashburner *et al.*, 2000), the unbalance characterizing most functional classes (few annotated "positive examples" and much more unannotated proteins), and the need of methods able to integrate the available heterogeneous sources of genomic, proteomic and transcriptomic data to achieve more accurate predictions (Cesa-Bianchi *et al.*, 2012).

Several attempts have been proposed in the literature for the *AFP* problem, ranging from sequence-based methods (Martin *et al.*, 2004; Hawkins *et al.*, 2009; Juncker *et al.*, 2009), to network-based methods (Sharan *et al.*, 2007; Mostafavi *et al.*, 2008; Bertoni *et al.*, 2011; Re *et al.*, 2012; Frasca, 2015), structured output algorithm based on kernels (Sokolov and Ben-Hur, 2010; Sokolov *et al.*, 2013) and hierarchical ensemble methods (Obozinski *et al.*, 2008; Guan *et al.*, 2008; Cesa-Bianchi and Valentini, 2010; Valentini, 2014). In particular, network-based methods represent the information coming from different experiments through graphs, in which nodes are genes/proteins and edges their functional pairwise relationships, and novel annotations are inferred by exploiting the topology of the resulting biomolecular network (Lippert *et al.*, 2010; Pandey *et al.*, 2009; Kourmpetis *et al.*, 2010; Mostafavi and Morris, 2010; Zhang and Dai, 2012; Youngs *et al.*, 2014).

In this context, several computational approaches achieved more accurate predictions by appropriately combining the available heterogeneous genomic and proteomic data networks, e.g. by keeping the edges in the majority of single functional networks (Marcotte

et al., 1999), or by constructing an unweighted sum of single networks (Pavlidis *et al.*, 2002), or by weighted sum, determining the weight of each network according to the function being predicted. The network weights have been computed by using different approaches, including SVM optimization (Lanckriet *et al.*, 2004; Linghu *et al.*, 2008), Gaussian random fields (Tsuda *et al.*, 2005), logistic (Yao and Ruzzo, 2006) and linear (Mostafavi *et al.*, 2008) regression. Moreover, some approaches simultaneously assign network weights to groups of related functions (Lan *et al.*, 2013; Mostafavi and Morris, 2010; Valentini *et al.*, 2014). Similarly, Chua *et al.*, 2007 integrated functional networks by keeping each edge in at least one of the single networks and by weighting edges according to the function to be predicted. Bayesian networks have also been applied to integrate heterogeneous networks by exploiting gene context information (Myers and Troyanskaya, 2007).

The results of the recent CAFA challenge showed that the integration of multiple data sources plays a key role in the automated function prediction of proteins (Radivojac *et al.*, 2013). In particular, Cozzetto *et al.* (2013) integrated both a wide variety of biological information sources and different prediction algorithms, taking into account also the hierarchy of GO terms. In (Sokolov *et al.*, 2013) a collection of genomic data is integrated to combine species-specific and cross-species views in the context of structured output prediction of protein functions, whereas Wang *et al.* (2013) in order to predict protein functions combined sequence-based, profile-based and domain co-occurrence-based data, and Lan *et al.* (2013) averaged scores obtained from sequence similarity, PPI and gene expression data according to a multi-source kNN approach.

Nevertheless, most of these approaches partially or totally neglect the labeling unbalance that affects the functional classes. Labeling unbalance may affect function-specific methods in both integration and prediction phases (Ling and Sheng, 2007). Indeed, the majority of GO terms and mainly the most specific ones (those close to leaves in the direct acyclic graph representing the GO) have a considerably low number of annotations and most proteins are unannotated. In this setting, classical supervised or semi-supervised

learning algorithms (e.g. SVMs or label propagation methods (Bengio *et al.*, 2006)) usually suffer performance decays, resulting sometimes in "all-negatives" predictions with poor sensitivity and precision (Cesa-Bianchi *et al.*, 2012; Frasca *et al.*, 2013a,b).

In this work we propose *UNIPred* (Unbalance-aware Network Integration and Prediction of protein functions), a novel network-based algorithm able to both combine multiple sources of data and infer functions for unknown proteins, particularly when the proportion of annotated proteins is significantly reduced. *UNIPred* can capture the suitability of each source of data for the prediction of specific functions and can handle the unbalance of data labelings both in the network integration and prediction steps of the method. The core of the integration algorithm is the transformation of each protein/node of the network into a labeled point in a bi-dimensional space, such that: (a) the local "label unbalance" of the node/protein with respect to the labeling of its neighborhood is embedded in the point position; (b) the "unbalance-aware" weight assigned to networks can be efficiently learned by linearly separating positive and negative points. The computed weights allows to construct the "consensus" network, which is then processed by a recently proposed semi-supervised classification algorithm based on parametric Hopfield networks (Frasca *et al.*, 2013a), designed to explicitly take into account the unbalance between annotated an unannotated proteins for each specific functional class.

2 Methods

In this section we first formalize the *AFP* problem in a semi-supervised scenario, then we introduce our novel unbalance-aware approach for integrating different protein networks. Finally, we outline its theoretical and experimental motivations.

2.1 Learning protein functions in a network-based semi-supervised setting

In a graph scenario, proteins can be represented by a set of nodes $V = \{1, 2, \ldots, n\}$, and relationships between proteins are encoded through a symmetric $n \times n$ real matrix $\boldsymbol{W}$, where W_{ij} represents a pre-computed functional similarity between proteins i and j. For a given functional class c, the subset of labeled nodes $S \subset V$ is divided into positive S_+ and negative S_- instances according to the corresponding labeling function $L_c : S \longrightarrow \{+, -\}$. The *AFP problem* consists in determining a labeling also for unlabeled nodes U, with $V = S \sqcup U$, starting from the known labeling and the connections $\boldsymbol{W}$. Here the symbol $\sqcup$ represents the disjoint union.

2.2 Unbalanced network integration and function prediction

Given a set of m biological networks, we represent each network through a weighted graph $G^{(d)} = \langle V^{(d)}, \boldsymbol{W}^{(d)} \rangle$, where $d \in \{1, 2, \ldots, m\}$, $V^{(d)}$ and $\boldsymbol{W}^{(d)}$ are the set of proteins and the connection matrix of *d-th* network respectively. Fixed a functional class c and the corresponding labeling function L_c, the labeling unbalance can be represented through a coefficient $\epsilon = |S_+^{(d)}|/|S_-^{(d)}|$, where $S_+^{(d)}$ and $S_-^{(d)}$ are the sets of positive and negative nodes of *d-th* network.

Assumed that labels of class c are significantly unbalanced, i.e. $\epsilon << 1$, the *unbalanced function prediction and network integration* problem consists in:

- the integration of biological networks $G^{(d)}$ in a "consensus" network $G = \langle V, \boldsymbol{W} \rangle$;

- the label prediction for unlabeled proteins/nodes $v \in U$ using the consensus network G.

The integration is performed by associating each network $G^{(d)}$ with a weight $h_c^{(d)}$ related to its "informativeness" for class c, and then by computing the weighted sum of the component networks. Here we use the term "informativeness" to reflect how much a

given data source is effective for the prediction of a given functional class.

2.3 *UNIPred*

UNIPred is a multi-step network-based method for the unbalance-aware data integration and prediction of protein functions. *UNIPred* consists of two steps:

1. A supervised algorithm to construct a single function-specific "consensus" network from multiple protein networks derived from different genomic, proteomic and transcriptomic data sources

2. An algorithm based on Hopfield networks for predicting protein functions given the consensus network

The integration step can be further divided in three sub-steps:

1.1. Projection of nodes. For each function and each network separately, the set of labeled nodes/proteins in the network is associated with a set of labeled points in $\mathbb{R}^2$

1.2. Linear separation of projected points. A function-specific parametric line is learned by a supervised algorithm in order to separate positive and negative points. The optimal line provides an unbalance-aware weight related to the informativeness of the network

1.3. Network integration. For each functional class, the computed weights are properly used to combine the input networks in a unique consensus network

Fig. 1 provides a schematic representation of the algorithm.

[Figure 1 about here.]

The projection of nodes at Step 1.1 embeds the information encoded in the network topology and transfers the label unbalance of the node neighborhood in the geometrical

position of the corresponding point. The weights computed at Step 1.2 are related to the unbalance-aware linear separability of the projected points, and at Step 1.3 the consensus network is constructed as a weighted sum of the input networks by using the computed weights as coefficients. Finally a cost-sensitive Hopfield network is applied to the resulting "consensus" network to predict protein functions. In the following sections each step of the algorithm is described in detail.

2.3.1 Projection of nodes.

For each network $G^{(d)}$, each node $k \in S^{(d)} = S_+^{(d)} \cup S_-^{(d)}$ is associated with a point $\Delta^{(d)}(k) \equiv \left(\Delta_+^{(d)}(k), \Delta_-^{(d)}(k)\right) \in \mathbb{R}^2$, whose abscissa and ordinate are respectively the weighted sum of its positive and negative connections:

$$\Delta_+^{(d)}(k) = \sum_{j \in S_+^{(d)}} W_{kj}^{(d)}, \qquad \Delta_-^{(d)}(k) = \sum_{j \in S_-^{(d)}} W_{kj}^{(d)}$$

The position of each point in the plane thereby reflects the topology of the connections towards neighboring positive and negative nodes. For a given class c, the label of point $\Delta^{(d)}(k)$ is the label $L_c(k)$ of node k. The bipartition $(S_+^{(d)}, S_-^{(d)})$ of labeled nodes $S^{(d)}$ induces in a natural way a bipartition $(I_+^{(d)}, I_-^{(d)})$ of the projected points $I^{(d)} = \{\Delta^{(d)}(k) \mid k \in S^{(d)}\}$ in positive and negative points:

$$I_+^{(d)} = \{\Delta^{(d)}(k) \mid k \in S_+^{(d)}\}, \qquad I_-^{(d)} = \{\Delta^{(d)}(k) \mid k \in S_-^{(d)}\}. \tag{1}$$

After the node projection, the label unbalance problem can be handled by appropriately exploiting the information coded in point positions.

2.3.2 Linear separation of projected points.

Consider now a parametric straight line in the plane of equation $f_{\alpha,\gamma}(x, y) = \cos\alpha \cdot\ y - \sin\alpha \cdot\ x + \gamma = 0$. It separates the projected points into points $I_{\alpha,\gamma,+}^{(d)}$ "below" and points

$I^{(d)}_{\alpha,\gamma,-}$ "above" the line $f_{\alpha,\gamma}(x,y)=0$ (Fig. 2):

$$I^{(d)}_{\alpha,\gamma,+} = \{\Delta^{(d)}(k) \mid f_{\alpha,\gamma}(\Delta^{(d)}(k)) \leq 0\}$$
$$I^{(d)}_{\alpha,\gamma,-} = \{\Delta^{(d)}(k) \mid f_{\alpha,\gamma}(\Delta^{(d)}(k)) > 0\}$$

[Figure 2 about here.]

We classify as positive the points below the line because, according to Eq. (1), points close to the abscissa axis, and more in general below the bisector of the first quadrant angle, have a prevalence of positive neighbors. Each pair values $(\overline{\alpha},\overline{\gamma})$ for the parameters (α,γ) determines a different straight line $f_{\overline{\alpha},\overline{\gamma}}$ to separate positive from negative examples. We denote with TP the number of positive points correctly classified by the line, with FP the number of negative points wrongly classified and with FN the number of positive points classified as negative. To explicitly consider the data unbalance that characterizes the *AFP* problem, the classification performance is assessed through the F-score measure, where F-score $= \frac{2TP}{2TP+FP+FN}$. To maximize the F-score we adopt a 2-step approximated algorithm:

1. *Compute* $\hat{\alpha}$. The algorithm computes the slopes $\tan\alpha$ of the straight lines crossing the origin and each point $\Delta^{(d)}(k) \in I^{(d)}_{+} \cup I^{(d)}_{-}$. Then it searches the line which maximizes the F-score by sorting the computed lines according to their slopes in an increasing order. Since all the points lie in the first quadrant, this assures that the angle $\hat{\alpha}$ relative to the optimum line is in the interval $[0, \frac{\pi}{2}[$ (Fig. 3 (a)).

2. *Compute* $\hat{\gamma}$. The parallel lines having the slope $\tan\hat{\alpha}$ computed at the previous step and crossing the projected points, are scanned from right to left (Fig. 3 (b)). Let $\hat{q}$ be the intercept of the line having the highest F-score: if $y = \tan\hat{\alpha}\cdot x + \hat{q}$, then $\hat{\gamma} = -\hat{q}\cos\hat{\alpha}$. The weight associated with the network $G^{(d)}$ with respect to class c is computed according to the corresponding "optimal" F-score $F^{(d)}_{c}$.

[Figure 3 about here.]

Although in general projected points are not linearly separable, a linear classifier on the one hand allows preventing an excessive increase of the model complexity; on the other hand, it likely avoids or at least reduces possible overfitting problems, due to the small number of available annotations for the majority of the functional classes characterizing this context.

The main reasons to adopt this specific approximated linear classifier to estimate the parameters $\hat{\alpha}$ and $\hat{\gamma}$ are its efficiency and scalability coupled with a good accuracy. Indeed, both Step 1 and 2 (which are executed just once) can be computed in $\mathcal{O}(n \log n)$ computational time (due to the sorting), where n is the number of points. The algorithm is approximated because it selects a "meaningful" subset of all possible parametric straight lines $f_{\alpha,\gamma}$ in both Step 1 and 2, and searches for the best separator among these lines. Although an exact algorithm for this problem working in time $\mathcal{O}(n^2 \log n)$ does exist (a simple extension of this algorithm), the proposed approximated classifier represents an appropriate trade-off between the quality of solution and the computational complexity, thus allowing an efficient application to complex genome and ontology-wide prediction tasks. It is worth noting that other linear algorithms could be in principle applied to estimate the "optimal" separator straight line, including variants of SVMs and logistic regression algorithms which optimize the F-score (Musicant *et al.*, 2003; Joachims, 2005; Jansche, 2005); nevertheless, we discarded these methods for their increased computational complexity.

2.3.3 Network integration.

We apply *UNIPred* to each GO term c separately, obtaining the F-score vector $\boldsymbol{F}_c = \{F_c^{(1)}, F_c^{(2)}, \cdots, F_c^{(m)}\}$. Using this vector, we consider three strategies for the weighted integration of networks:

- ***WAP*** *(Weighted Average Per-class).* A consensus network for each term c is con-

structed using $\boldsymbol{h}_c = \{h_c^{(1)}, h_c^{(2)}, \cdots, h_c^{(m)}\}$ as network weights, with $h_c^{(d)} = \frac{F_c^{(d)}}{\sum_i F_c^{(i)}}$.

- ***WA*** *(Weighted Average).* A consensus network for each GO ontology (BP, MF, CC) is constructed by averaging $\boldsymbol{h}_c$ across the corresponding ontology terms: $h_D^{(d)} = \frac{1}{|D|}\sum_{c\in D} h_c^{(d)}$, where D is one of BP, MF and CC ontology and $|D|$ is the number of terms in D.

- ***WAC*** *(Weighted Average per-Category).* Since GO terms with similar specificity are characterized by similar label unbalance, we construct a composite network for each GO category. Similarly to Pena-Castillo *et al.* (2008), we consider different ranges of specificity, that is the number of training proteins annotated to the term: $3-10$, $11-30$, $31-100$, $101-300$, for a total of 4 categories for each GO ontology. The vector weight $\boldsymbol{h}_c$ is thereby averaged across each category separately.

Once computed the weights $\boldsymbol{h}_c$ with one of the above-mentioned strategies, we computed the consensus network as weighted sum of the corresponding adjacency matrices:

$$\boldsymbol{W} = \sum_{d=1}^{m} h^{(d)} \boldsymbol{W}^{(d)} \tag{2}$$

Moreover, in order to have a base line comparison, networks are also integrated by unweighted average sum (*UA*). The *WAP* strategy leads to the construction of a network well-suited to each specific functional class, while *WA* and *WAC* introduce a sort of "regularization" by averaging across categories of GO terms or across an entire ontology. *WAP* can fit well the data with respect to a specific GO term, but it can also overfit the data. On the contrary, *WA* and *WAC* can overcome this problem, but can undergo the opposite problem of underfitting.

2.3.4 Functional prediction with the consensus network.

The Step 2 of *UNIPred* has been performed by applying *COSNet*, COst-Sensitive neural Network (Frasca *et al.*, 2013a) to the constructed consensus network.

COSNet is based on parametric Hopfield networks $H =< \boldsymbol{W}, k, \rho >$, where k is the neuron activation threshold and ρ is a real number in $[0, \frac{\pi}{2}[$, that determines the two different values $\{\sin\rho, -\cos\rho\}$ for neuron activation. Informally, to deal with data unbalance, *COSNet* conceptually separates node labels and neuron activation values, which become parameters to be learned. The main steps of *COSNet* can be summarized as follows:

INPUT: symmetric connection matrix $W : V \times V \longrightarrow [0, 1]$, labeling function $L : V \longrightarrow \{+, -\}$, sets S_+, S_- and U of respectively positive, negative and unlabeled instances.

Step A. Generate an initial temporary bipartition (U_+, U_-) of U such that $\frac{|U_+|}{|U|} \simeq \frac{|S_+|}{|S|}$, where $S = S_+ \sqcup S_-$.

Step B. Find the optimal parameters $(\hat{\rho}, \hat{k})$ of the sub-network of labeled nodes, such that the state represented by known labels is "as close as possible" to an equilibrium state of the Hopfield network.

Step C. Extend the parameters $(\hat{\rho}, \hat{k})$ to the whole network and run the sub-network restricted to unlabeled nodes until an equilibrium state $\hat{\boldsymbol{u}}$ is reached and a final bipartition (U_+, U_-) of U is obtained.

OUTPUT: bipartition (U_+, U_-) of U.

Step A provides a temporary solution in order to exploit the connections among labeled and unlabeled nodes during the learning phase.

Step B learns from the labeled data the optimal parameters ρ and k of the parametrized Hopfield network: labeled nodes are projected to a bidimensional space and the line $f_{\rho,k}(x, y) = \cos\rho \cdot y - \sin\rho \cdot x + k = 0$ that "better" separates positive and negative points is learned to determine an estimate $(\hat{\rho},\ \hat{k})$ of the optimal parameters. In (Frasca

et al., 2013a) the authors showed that the dynamics of a Hopfield network with parameters $(\hat{\rho}, \hat{k})$ learned from labeled nodes $v \in S$ preserves convergence and optimization properties of the whole Hopfield network including both labeled and unlabeled nodes.

In Step C the dynamics of the network restricted to unlabeled nodes U is simulated by adopting neuron activation values $\{\sin\hat{\rho},\ -\cos\hat{\rho}\}$ and activation thresholds $\hat{k}$. Assuming that, up to a permutation, $U = \{1, 2, \cdots, h\}$ and $S = \{h+1, h+2, \cdots, n\}$, the initial state is set to $u_i(0) = 0$ for each neuron $i \in U$. The network evolves according to the following asynchronous update rule:

$$u_i(t) = \begin{cases} \sin\hat{\rho} & \text{if } \sum_{j=1}^{i-1} W_{ij}u_j(t) + \sum_{k=i+1}^{h} W_{ik}u_k(t-1) - \theta_i > 0 \\ -\cos\hat{\rho} & \text{if } \sum_{j=1}^{i-1} W_{ij}u_j(t) + \sum_{k=i+1}^{h} W_{ik}u_k(t-1) - \theta_i \leq 0 \end{cases}$$

where $u_i(t)$ is the value of neuron $i \in U$ at time t, $\theta_i = \hat{k} - \sum_{j=h+1}^{n} W_{ij}L_c(j)$ is the activation threshold of node i, which also includes the influence on this node of the labeled neurons S, which are not updated during the network dynamics (see Frasca *et al.* (2013a) for details).

At each time t, the state of the network is $\boldsymbol{u}(t) = (u_1(t), u_2(t), \ldots, u_h(t))$, and the following Lyapunov state function (energy function) is associated with the network:

$$E(\boldsymbol{u}) = -\frac{1}{2}\sum_{\substack{i,j=1 \\ j\neq i}}^{h} W_{ij}u_iu_j + \sum_{i=1}^{h} u_i\theta_i \tag{3}$$

As mentioned above, the dynamics converges to a fixed point $\hat{\boldsymbol{u}}$ corresponding to a minimum of E. The final solution (U_+, U_-) is:

$$U_+ = \{i \in U \mid \hat{u}_i = +\sin\hat{\rho}\}$$
$$U_- = \{i \in U \mid \hat{u}_i = -\cos\hat{\rho}\}$$

2.4 Algorithm motivation

The intuitive rationale behind *UNIPred* comes from the projection of the network nodes into the plane, which provides an alternative representation of the labeling unbalance. Indeed a point close to the y axis has more negative than positive neighbor nodes, and the opposite is true for points close to the x axis (Fig. 4). By adopting the F-score as maximization criterion, the learning algorithm described in Sect. 2.3.2 tends to discard the lines which incorrectly classify positive points; the learned parameters thereby tend to counterbalance the strong prevalence of negatives.

[Figure 4 about here.]

In addition to this intuitive rationale, *UNIPred* is supported by both theoretical and experimental motivations.

Theoretical motivation. Suppose that we apply *UNIPred* to assign an unbalance-aware weight to a given network $G = \langle V, \boldsymbol{W} \rangle$. Consider the parametric Hopfield network $H = < \boldsymbol{W}, k, \rho >$ constructed by *COSNet* using network G, and let L_c be the labeling function associated with class c, and S the set of labeled nodes in G. If $f_{\hat{\alpha},\hat{\gamma}}$ is the optimum line computed by *UNIPred* with respect to the labeling function L_c, and F_c the corresponding optimal F-score value, then the following fact holds:

Theorem 1 . *If $\rho = \hat{\alpha}$ and $k = \hat{\gamma}$, then $F_c = 1$ iff $L_c(S)$ is an equilibrium state of H restricted to neurons in S.*

Proof: See Supplementary Data.

Theorem 1 shows that F-scores computed by *UNIPred* are related to the "stability" of the associated *COSNet* parametric Hopfield network: the higher F_c, the closer $L_c(S)$ to an equilibrium state of the sub-network restricted to neurons in S. Since the equilibrium state of the Hopfield network (minimum of the energy function (3)) is related to the "consistence" with the prior information coded in the network topology, by Theorem 1 we can conclude that the higher *UNIPred* weights F_c, the more the consistence with the

prior information coded in the data network. Theorem 1 also shows that does exist a relationship between the first two steps of the integration procedure of *UNIPred* and Step B of *COSNet*. The following section experimentally verifies this observation.

Experimental motivation. For each GO term c separately, we compared the F-scores $F_c^{(d)}$, computed in the *UNIPred* integration steps (specifically Step 1.1 and 1.2) on each input network $G^{(d)}$, with the F-scores $PF_c^{(d)}$ achieved by *COSNet* algorithm applied to predict the same network $G^{(d)}$, without integration. The obtained vectors across GO terms are respectively denoted with $\boldsymbol{F}^{(d)}$ and $\boldsymbol{PF}^{(d)}$. More precisely, we considered 17 mouse networks described in Section 3.1.1 (thus $d \in \{1, 2, \cdots, 17\}$), and thousands of GO terms belonging to all the GO ontologies BP, MF and CC. Our aim is to verify whether higher $PF_c^{(d)}$ corresponds to higher $F_c^{(d)}$, that is whether the weight assigned by *UNIPred* is larger when *COSNet* performs better and viceversa. Accordingly, we computed for each network $G^{(d)}$ the Pearson correlation between $\boldsymbol{F}^{(d)}$ and $\boldsymbol{PF}^{(d)}$. The results show that a non negative correlation holds for all the considered networks (Fig. 5). Supplementary Data include correlations computed separately for each GO ontology (Fig. S3), confirming that the non negative correlations hold for each ontology.

[Figure 5 about here.]

This means that the F-scores $F_c^{(1)}, F_c^{(2)}, \cdots$ provide in advance an advice about the most predictive networks for *COSNet* thus justifying their usage in the network integration steps of *UNIPred*.

Moreover, the correlation is higher for some networks (e.g. networks 8, 9, 12-15) and lower for the others. This may be due to the different size of networks, since networks having a lower number of proteins, e.g. network 10 (*Pheno*) and 11 (*Omim*), are characterized for the majority of considered GO terms by a very low number of annotations for test proteins (1 positive example for most of the terms). This leads to difficult prediction tasks and consequently to low predictive performances of *COSNet* ($PF_c^{(d)} \simeq 0$, for each GO term c). In order to verify this observation, we also report in Fig. 5 (b) the

correlation relative only to the GO terms with more than 30 annotations. As expected, the correlation considerably improves for most of the considered networks.

3 Results and discussion

We applied *UNIPred* to the prediction of GO annotations in *M. musculus*, comparing our method with state-of-the-art learning algorithms that participated to the *MouseFunc* challenge (Pena-Castillo *et al.*, 2008). To assess the effectiveness of the proposed approach with respect to the novel experimental annotations accumulated in more recent years, we reported also the results relative to more recent GO annotations of mouse proteins. Then we performed a genome-wide analysis of protein functions in *D. melanogaster* and *S. cerevisiae* comparing our proposed methods with the classical guilt-by-association algorithm (GBA) (Mayer and Hieter, 2000), with the state-of-the-art GeneMANIA-SW Mostafavi and Morris (2010), derived from one of top methods in the MouseFunc challenge (Mostafavi *et al.*, 2008), and with MS-kNN, one of the top-ranked methods in the recent CAFA challenge (Lan *et al.*, 2013). Since the three proposed integration strategies (Section 2.3.3) achieve very similar overall results, we report in the following sections only the results of the *WA* strategy. Additional information about data and results are available in the Supplementary Data. **Finally, at the end of this section, we analyse the "informativeness" of each single data source when predicting protein functions, and in particular we investigate whether it is possible to rank data sources according to their informativeness and whether this rank depends on the protein function to be predicted.**

3.1 Experimental setting

3.1.1 Mouse

To compare our results with those achieved by participants to the MouseFunc challenge, we adopted the same data and annotations (GO annotations 17 February 2006; version

1.612) used for the challenge. In the MouseFunc setting, 21603 mouse proteins and 2815 GO terms with a number of annotations ranging from 3 to 300 have been considered, excluding GO annotations based solely on the "inferred from electronic annotation" (IEA) evidence code. A randomly selected set of 1718 proteins is held-out and their annotations have to be predicted using the annotations of the remaining proteins. We integrated 17 mouse networks in a consensus network with 21603 nodes, and adopted GO terms with at least one annotation in test set, obtaining 1174 BP, 442 MF and 231 CC terms (see Supplementary Data for details).

3.1.2 Yeast and Fly.

We applied *UNIPred* to integrate 16 *S. cerevisiae* and 10 *D. melanogaster* networks downloaded from the GeneMANIA website (`www.genemania.org`) and 3 GO networks (release 23-3-13 for yeast and 15-5-13 for fly), covering a set of 5775 yeast and 9361 fly proteins. The networks have been selected to cover different types of data, ranging from co-expression, to genetic interactions, protein ontologies and physical interactions (see Supplementary Data Table S4 and S5 for more details). Since GeneMANIA networks already provide for each pair of proteins a real score representing a measure of their functional similarity, no preprocessing has been applied.

Finally, we considered the GO terms with $3-300$ positive annotated proteins, obtaining 3469 terms (2021 BP, 805 MF and 593 CC) for yeast and 4350 terms (2781 BP, 1023 MF and 546 CC) for fly. We assessed the generalization performances through 10-fold cross-validation techniques.

3.1.3 Metrics.

The evaluation of the performance for AFP problems raised heated discussions in the scientific community (Gillis and Pavlidis, 2013). We adopted a "function's point of view" evaluation (i.e. the performance are measured on each GO term), to avoid the problems related to the "protein-centric" evaluation adopted in the recent CAFA challenge (Radi-

vojac *et al.*, 2013). More precisely, we measured the Area Under the Curve (AUC), the precision at 20% recall (P20R) and the F-score, to properly take into account the unbalance that characterizes GO terms. We used the classic definition of F-score, i.e. the harmonic mean between precision and recall.

Finally, we point out that in the context of *AFP*, where GO terms are usually unbalanced, the F-score and precision at a given recall are more significant than AUC to evaluate the accuracy of prediction methods, since AUC does not properly take into account the labeling unbalance.

3.2 MouseFunc benchmark results

We compared *UNIPred* with the best 8 methods which participated in the MouseFunc challenge (Table S2 describes the compared methods) and with *COSNet* applied to the *UA* consensus network. *UNIPred* achieves the best results in terms of average F-score and P20R, except for P20R on MF terms, since the Funkenstein algorithm (Method G – Tian *et al.* (2008)) performs slightly better (Table 1). Furthermore, *UNIPred* improves the performance of *COSNet-UA* in all the performed experiments, showing the effectiveness of *UNIPred* in boosting *COSNet* predictive capability.

UNIPred improvements are almost always significant with respect to the compared methods (Wilcoxon signed rank test, $\alpha = 0.01$). Detailed results about the statistical comparison between methods are available in Table S7 in the Supplementary Data.

[Table 1 about here.]

COSNet is a classifier, and to obtain results in terms of P20R and AUC we constructed a ranker by simply considering for each node the internal energy at equilibrium as ranking score (Frasca and Pavesi, 2013):

$$s(i) = \sum_{j \neq i} (W_{ij} \hat{u}_j - \theta_i) \tag{4}$$

where $s(i)$ is the score assigned to node i, $\hat{\boldsymbol{u}}$ is the equilibrium state of the Hopfield network (Sect. 2.3.4), θ_i is the activation threshold for node i.

In terms of AUC, GeneMANIA (Method C – Mostafavi *et al.* (2008)) is the best method, but *UNIPred* achieves results comparable with the best performing methods (Table S6), and significantly worse only than GeneMANIA and two hierarchical methods (Method D and G) on BP terms.

In addition to results averaged by GO ontology, Fig. 6 reports the F-score averaged across GO terms, grouped by cardinality of the annotations (categories). According to the MouseFunc experimental set-up we considered categories with 3 to 10 ($3-10$), $11-30$, $31-100$ and $101-300$ annotations. Interestingly, *UNIPred* obtains the best results in each ontology and each category, except for the smallest BP and MF categories. AUC and P20R results grouped by categories are shown in Fig. S4 in the Supplementary Data.

[Figure 6 about here.]

Summarizing, by exploiting its unbalance-aware characteristics, *UNIPred* outperforms in terms of F-score and P20R the other state-of-the-art MouseFunc challenge methods. In terms of AUC *UNIPred* is comparable with the best MouseFunc methods and significantly worse than GeneMANIA and two hierarchical methods (Method D and G) only with the BP ontology (Table S7 - Supplementary Data).

3.3 MouseFunc results with updated annotations

We repeated the experiments described in the previous section using annotations updated to the 15/08/2012 GO release. In terms of the F-score and P20R, *UNIPred* BP predictions are strongly enhanced (Wilcoxon signed rank test, $\alpha = 0.01$), while for MF and CC the results are comparable with those obtained with the previous annotation (February 2006), showing that on the average several proteins predicted as false positive are actually true positive, according to the updated annotation (Table 2).

[Table 2 about here.]

AUC, F-score and P20R results for each of the analyzed 1782 GO terms are available on-line at *http://frasca.di.unimi.it/data/*.

3.4 Experiments with yeast and fly

In Table 3 we report the results averaged by GO ontology for both yeast and fly.

[Table 3 about here.]

UNIPred outperforms both *GeneMANIA-SW* and the top-ranked CAFA method *MS-kNN* (Lan *et al.*, 2013), in terms of both P20R and F-score in all the GO ontologies with both yeast and fly model organisms (except for the BP ontology in yeast). Fig. 7 shows yeast results averaged by GO category, whereas fly results are reported in Supplementary Data – Fig. S6.

[Figure 7 about here.]

COSNet with *UA* integration is the second best method, except for fly results in the MF $11-30$ and $31-100$ categories, where *MS-kNN* is the second best method. It is worth noting that the very low F-score results of *GeneMANIA-SW* (Table 3) are likely due to the non optimized choice of the threshold (see Section 3 in Supplementary Data for details): *GeneMANIA-SW* is basically a ranking method and a well-tuned selection of the threshold for the computation of the F-score may lead to significantly better results.

Finally, confirming mouse results, *UNIPred* achieves better performances than the simple *UA* integration.

3.5 Analysis of the informativeness of network sources

In this section we discuss the performance of *COSNet* on each specific mouse network (see Section 1 of Supplementary Data for a detailed description of each network), in order to analyse which type of data is more "useful" in inferring protein functions. Since the 17 networks are composed by different subsets of proteins, in our setting we considered the 650 GO terms with

at least one positive in the test set in all the networks. Firstly, we studied which sources are more predictive on the average across all the considered GO terms. To this end, in Figure 8 we report the F-score achieved by *COSNet* averaged across the 650 GO terms, for each mouse network separately. These results show that protein domains/families/sequence patterns (Pfam and Interpro, network 8 and 9) are the more informative with respect to the entire ontology, together with the protein-protein interaction data (PPIbin, network 1), achieving the highest F-score values in most of the considered terms. Nevertheless, it is worth noting that this does not mean that the remaining networks are not predictive for any GO term. This fact can be observed in Figure 9, where we show the heat map representing the per class F-score values obtained by *COSNet* with regard to every term and mouse network. The values are decreasingly ordered with respect to network 1 (protein-protein interactions). Lighter colors represent higher F-scores.

[Figure 8 about here.]

[Figure 9 about here.]

The figure emphasizes that the informativeness of each network may significantly vary with the considered GO term, and interestingly networks which in average are less predictive, e.g. genetic interactions (networks 5, 6) or orthologs data (networks 13, 14), have some light bands in correspondence of dark bands for PPI, Pfam and Interpro data (networks 1, 8 and 9): that is, for some specific GO terms, just the networks in average less informative allow to correctly predict some specific protein functions. For example, in Table 4 we report the F-score obtained by *COSNet* for each network on some selected terms. Genetic interactions data (network 6) are the most informative for the *positive regulation of T cell proliferation* term (GO:0042102), whereas

OMIM data (network 11) are the most informative for the *damaged DNA binding* term (GO:0003684)[a].

[Table 4 about here.]

These results, which confirm previous analyses reported in literature (Myers and Troyanskaya, 2007), are to some extent expected, since each single source provides a distinct "view" of the functional domain of a protein, and potentially encodes different patterns that may be relevant to detect some functions, but scarcely relevant or completely irrelevant for other functions.

4 Conclusions

By explicitly considering the labeling unbalance that characterizes the *AFP* problem and the specific characteristics of each source of data, our proposed approach is able to properly combine different biomolecular networks and learn the GO terms associated to each protein included in the integrated network. GO terms are usually characterized by a relevant unbalance between annotated and unannotated proteins, and *UNIPred* explicitly addresses this issue, both in the integration and in the prediction steps of the method. To our knowledge this is the first network-based method that introduces an unbalance-aware combination of networks for the *AFP* problem. Both theoretical and experimental results show that *UNIPred* is a well-suited method for *AFP*. In particular, results with Eukaryotic organisms show the effectiveness of *UNIPred* with respect to several state-of-the-art learning algorithms that participated to the MouseFunc and CAFA *AFP* challenges, especially when unbalance-aware metrics are considered. Unbalance aware integration and prediction items are of paramount importance for network-based *AFP*, confirming

[a]Note that these considered terms have very few positives in test set: for instance, F-score = 0 means that the few available positive examples have been predicted as negative; F-score = 0.667 may correspond to the case in which just one of two available positive items has been correctly classified, or that the only available positive has been correctly classified and one negative item has been predicted as positive.

previous results obtained with inductive methods (Cesa-Bianchi *et al.*, 2012), but the hierarchical correction of the predictions (Mostafavi and Morris, 2009; Cesa-Bianchi *et al.*, 2012; Cozzetto *et al.*, 2013; Robinson *et al.*, 2015), and the multi-species setting of the classification problem (Wong *et al.*, 2012; Mesiti *et al.*, 2014) could further improve the performances of the proposed method, **as shown by our recent results obtained with multi-category Hopfield networks, a variant of *COSNet* well-suited to multi-species protein function prediction problems (Frasca *et al.*, 2015).**

Acknowledgement

The authors acknowledge partial support from the *PRIN* project H41J12000190001: "Automi e linguaggi formali: aspetti matematici e applicativi", funded by the italian Ministry of University.

Author Disclosure Statement

No competing financial interests exist.

References

Ashburner, M., Ball, C. A., Blake, J. A., *et al.* (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, **25**(1), 25–29.

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

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

Bertoni, A., Frasca, M., and Valentini, G. (2011). Cosnet: A cost sensitive neural network for semi-supervised learning in graphs. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *ECML/PKDD (1)*, volume 6911, pages 219–234.

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

Cesa-Bianchi, N., Re, M., and Valentini, G. (2012). Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Mach. Learn.*, **88**(1-2), 209–241.

Chen, Y. and Xu, D. (2004). Global protein function annotation through mining genome-scale data in yeast saccharomyces cerevisiae. *Nucleic Acids Res*, **32**(21), 6414–6424.

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

Cozzetto, D., Buchan, D. W. A., Bryson, K., *et al.* (2013). Protein function prediction by massive integration of evolutionary analyses and multiple data sources. *BMC Bioinformatics*, **14**(Suppl 3:S1).

Frasca, M. (2015). Automated gene function prediction through gene multifunctionality in biological networks. *Neurocomputing*, **162**(0), 48 – 56.

Frasca, M. and Pavesi, G. (2013). A neural network based algorithm for gene expression prediction from chromatin structure. In *IJCNN*, pages 1–8. IEEE.

Frasca, M., Bertoni, A., Re, M., *et al.* (2013a). A neural network algorithm for semi-supervised node label learning from unbalanced data. *Neural Networks*, **43**(0), 84 – 98.

Frasca, M., Bertoni, A., and Sion, A. (2013b). A neural procedure for gene function prediction. In *Neural Nets and Surroundings*, volume 19 of *Smart Innovation, Systems and Technologies*, pages 179–188. Springer Berlin Heidelberg.

Frasca, M., Bassis, S., and Valentini, G. (2015). Learning node labels with multi-category hopfield networks. *Neural Computing and Applications*.

Gillis, J. and Pavlidis, P. (2013). Characterizing the state of the art in the computational assignment of gene function: lessons from the first critical assessment of functional annotation (CAFA). *BMC Bioinformatics*, **14**(Suppl 3:S15).

Guan, Y., Myers, C. L., and Hess, D. C. (2008). Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome biology*, **9 Suppl 1**.

Hawkins, T., Chitale, M., Luban, S., *et al.* (2009). Pfp: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. *Proteins*, **74**(3), 566–82.

Jansche, M. (2005). Maximum expected F-measure training of logistic regression models. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 692–699, Morristown, NJ, USA. Association for Computational Linguistics.

Joachims, T. (2005). A support vector method for multivariate performance measures. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 377–384, New York, NY, USA. ACM.

Juncker, A. S., Jensen, L. J., Pierleoni, A., *et al.* (2009). Sequence-based feature prediction and annotation of proteins. *Genome Biology*, **10:206**.

Kim, W., Krumpelman, C., and Marcotte, E. (2008). Inferring mouse gene functions from genomic-scale data using a combined functional network/classification strategy. *Genome Biology*, **9**(Suppl 1), S5+.

Kourmpetis, Y. A. I., van Dijk, A. D. J., Bink, M. C. A. M., *et al.* (2010). Bayesian markov random field analysis for protein function prediction based on network data. *PLoS ONE*, **5**(2:e9293).

Lan, L., Djuric, N., Guo, Y., *et al.* (2013). MS-kNN: protein function prediction by integrating multiple data sources. *BMC Bioinformatics*, **14**(Suppl 3:S8).

Lanckriet, G. R. G., De Bie, T., Cristianini, N., *et al.* (2004). A statistical framework for genomic data fusion. *Bioinformatics*, **20**(16), 2626–2635.

Lee, H., Tu, Z., Deng, M., *et al.* (2006). Diffusion kernel-based logistic regression models for protein function prediction. *Omics : a journal of integrative biology*, **10**(1), 40–55.

Ling, C. X. and Sheng, V. S. (2007). *Cost-sensitive Learning and the Class Imbalanced Problem*.

Linghu, B., Snitkin, E. S., Holloway, D. T., *et al.* (2008). High-precision high-coverage functional inference from integrated data sources. *BMC Bioinformatics*, **9**, 119.

Lippert, G. *et al.* (2010). Gene function prediction from synthetic leathality networks via ranking on demand. *Bioinformatics*, **26**(7), 912–918.

Marcotte, E. M., Pellegrini, M., Thompson, M. J., *et al.* (1999). A combined algorithm for genome-wide prediction of protein function. *Nature*, **402**(6757), 83–86.

Martin, D., Berriman, M., and Barton, G. (2004). Gotcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics*, **5**, 178.

Mayer, M. L. and Hieter, P. (2000). Protein networks-built by association. *Nat Biotechnol*, **18**(12), 1242–3.

Mesiti, M., Re, M., and Valentini, G. (2014). Think globally and solve locally: secondary memory-based network learning for automated multi-species function prediction. *GigaScience*, **3**(5).

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

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

Mostafavi, S., Ray, D., Farley, D. W., *et al.* (2008). GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, **9**(Suppl 1), S4+.

Musicant, D. R., Kumar, V., and Ozgur, A. (2003). Optimizing f-measure with support vector machines. In *In Proceedings of the international*, pages 356–360.

Myers, C. L. and Troyanskaya, O. G. (2007). Context-sensitive data integration and prediction of biological networks. *Bioinformatics*, **23**(17), 2322–2330.

Obozinski, G., Lanckriet, G., Grant, C., *et al.* (2008). Consistent probabilistic outputs for protein function prediction. *Genome Biol*, **9 Suppl 1**, S6.

Pandey, G., Myers, C., and Kumar, V. (2009). Incorporating functional inter-relationships into protein function prediction algorithms. *BMC Bioinformatics*, **10**(1-142).

Pavlidis, P., Cai, J., Weston, J., *et al.* (2002). Learning gene functional classifications from multiple data types. *Journal of Computational Biology*, **9**, 401–411.

Pena-Castillo, L., Tasan, M., Myers, C., *et al.* (2008). A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. *Genome Biology*, **9**, S1.

Qi, Y., Seetharaman, J. K., and Joseph, Z. B. (2007). A mixture of feature experts approach for protein-protein interaction prediction. *BMC Bioinformatics*, **8**(Suppl 10), S6+.

Radivojac, P., Clark, W. T., Oron, T. R., *et al.* (2013). A large-scale evaluation of computational protein function prediction. *Nature Methods*, **10**(3), 221–227.

Re, M., Mesiti, M., and Valentini, G. (2012). A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks. *IEEE ACM Transactions on Computational Biology and Bioinformatics*, **9**(6), 1812–1818.

Robinson, P., Frasca, M., Kohler, S., Notaro, M., Re, M., and Valentini, G. (2015). A hierarchical ensemble method for DAG-structured taxonomies. In *Multiple Classifier Systems - MCS 2015*, volume 9132 of *Lecture Notes in Computer Science*, pages 15–36. Springer.

Sharan, R., Ulitsky, I., and Shamir, R. (2007). Network-based prediction of protein function. *Mol. Sys. Biol.*, **8**(88).

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

Sokolov, A., Funk, C., Graim, K., *et al.* (2013). Combining heterogeneous data sources for accurate functional annotation of proteins. *BMC Bioinformatics*, **14**(Suppl 3:S10).

Tian, W., Zhang, L., Tasan, M., *et al.* (2008). Combining guilt-by-association and guilt-by-profiling to predict saccharomyces cerevisiae gene function. *Genome biology*, **9 Suppl 1**(Suppl 1), S7+.

Tsuda, K., Shin, H., and Scholkopf, B. (2005). Fast protein classification with multiple networks. *Bioinformatics*, **21**(suppl_2), ii59–65.

Valentini, G. (2014). Hierarchical Ensemble Methods for Protein Function Prediction. *ISRN Bioinformatics*, **2014**(Article ID 901419), 1–34.

Valentini, G., Paccanaro, A., Caniza, H., Romero, A., and Re, M. (2014). An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods. *Artificial Intelligence in Medicine*, **61**(2), 63–78.

Wang, Z., Cao, R., and Cheng, J. (2013). Three-Level Prediction of Protien Function by Combining Profile-Sequence Search, Profile-Profile Search, and Domain Co-Occurence Networks. *BMC Bioinformatics*, **14**(Suppl 3:S3).

Wong, A. K., Park, C. Y., Greene, C. S., *et al.* (2012). Imp: a multi-species functional genomics portal for integration, visualization and prediction of protein functions and networks. *Nucleic acids research*, **40**(W1), W484–W490.

Yao, Z. and Ruzzo, W. L. (2006). A regression-based k nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC Bioinformatics*, **7**(S-1).

Youngs, N., Penfold-Brown, D., Bonneau, R., and Shasha, D. (2014). Negative Example Selection for Protein Function Prediction: The NoGO Database. *PLoS Comput Biol*, **10**(6), e1003644+.

Zhang, X. and Dai, D. (2012). A framework for incorporating functional interrelationships into protein function prediction algorithms. *IEEE ACM Transactions on Computational Biology and Bioinformatics*, **9**(3), 740–753.

Address correspondence to:

Dr. Marco Frasca

DI - Department of Computer Science

University of Milan

Via Comelico 39, 20135 Milan, Italy

e-mail: frasca@di.unimi.it

phone: +39 0250316321

and

Prof. Alberto Bertoni

DI - Department of Computer Science

University of Milan

Via Comelico 39, 20135 Milan, Italy

and

Prof. Giorgio Valentini

DI - Department of Computer Science

University of Milan

Via Comelico 39, 20135 Milan, Italy

e-mail: valentini@di.unimi.it

phone: +39 0250316255

fax: +39 0250316373

List of Figures

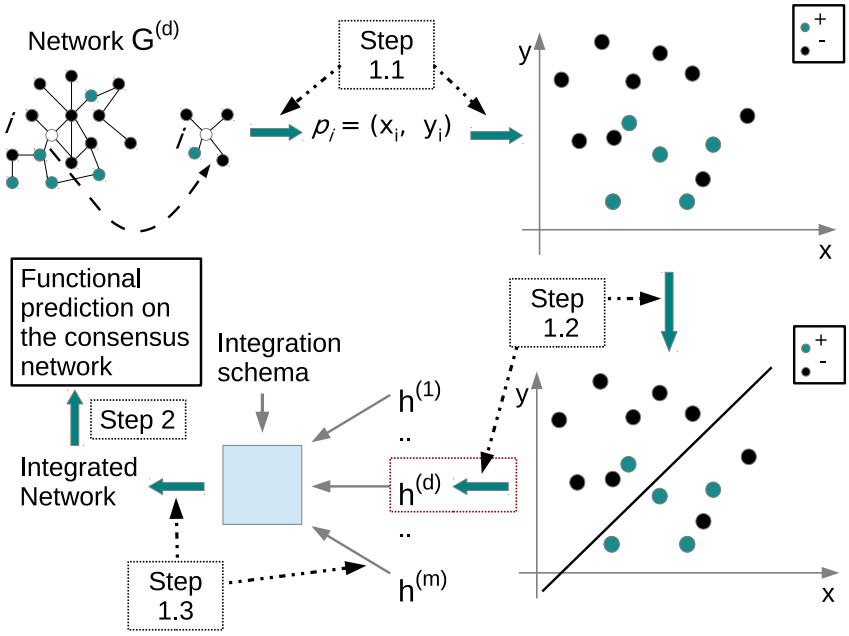


Figure 1: *UNIPred* steps. Step 1.1) Projection of nodes. Step 1.2) Linear separation of projected points. Step 1.3) Network integration. Step 2) Functional prediction on the consensus network.

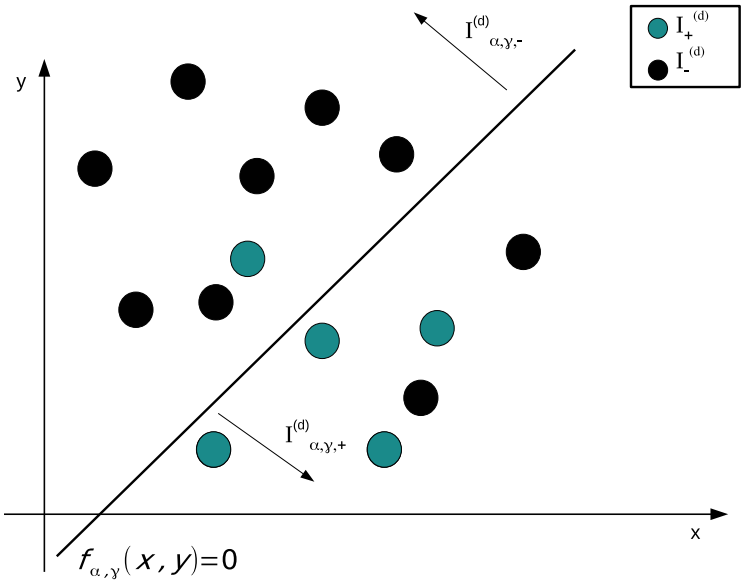
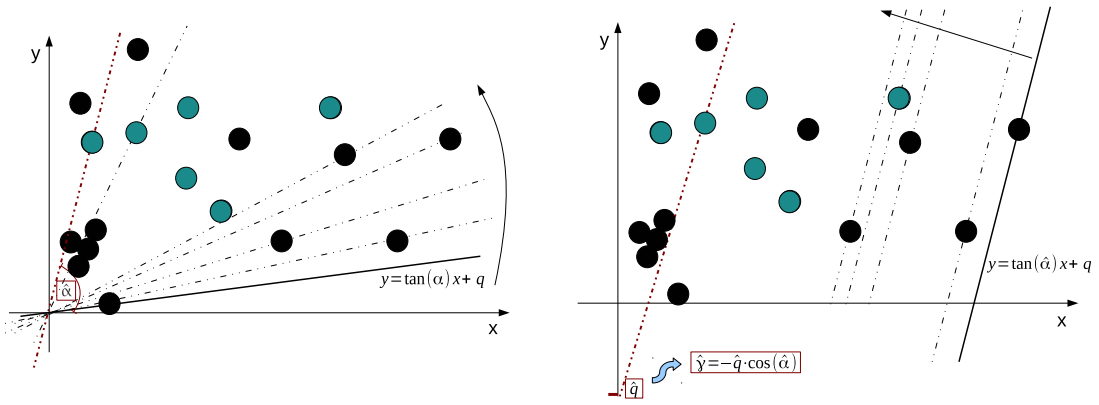


Figure 2: Positive and negative point in *d-th* network are separated by a straight line $f_{\alpha,\gamma}(x,y) = 0$.



(a) (b)

Figure 3: Optimization of the slope (a) and the intercept (b) of the parametric line.

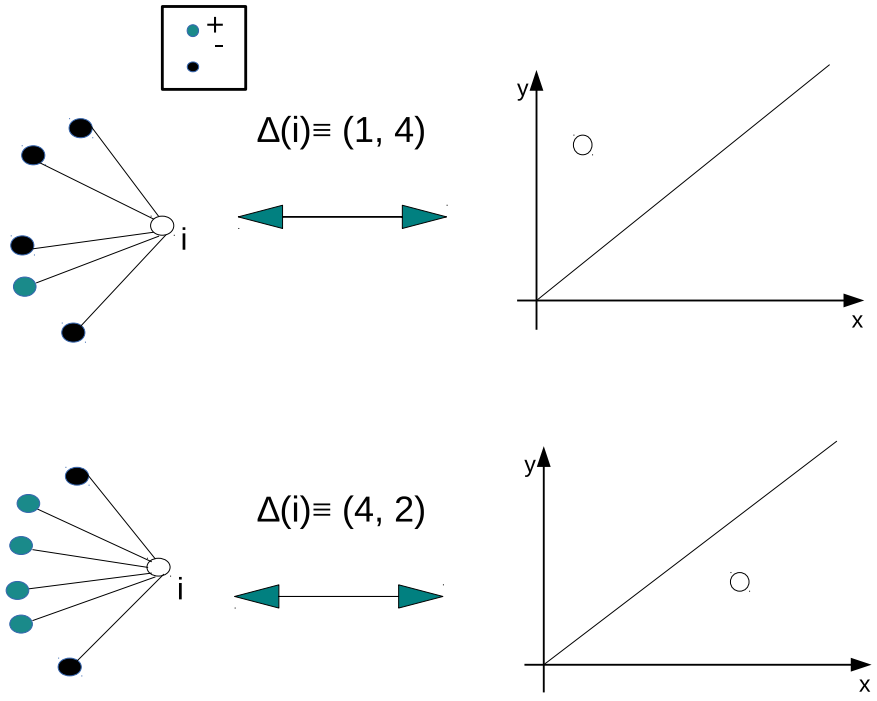
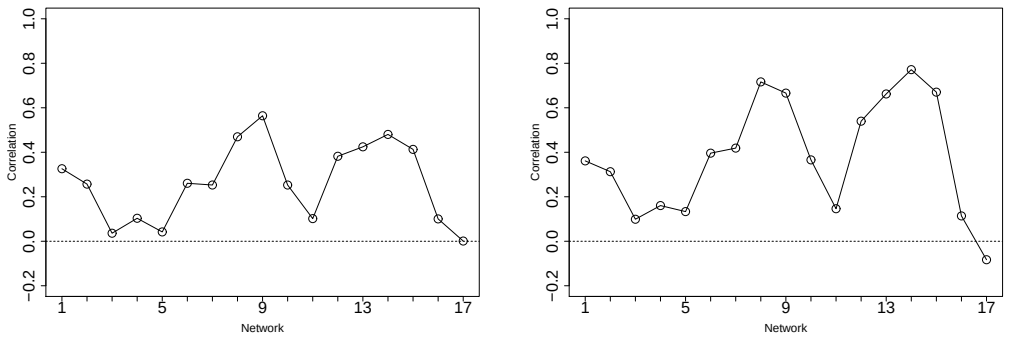


Figure 4: Projection of a node in the network into a point in the plane.



(a) (b)

Figure 5: (a) Pearson's correlation between vectors $\boldsymbol{F}^{(d)}$ and $\boldsymbol{PF}^{(d)}$ for each network $G^{(d)}$, with $d = \{1, 2, \cdots, 17\}$. (b) Correlation accounting only for terms having at least 30 annotations. See Table S1 in Supplementary Data for the correspondence index-network.

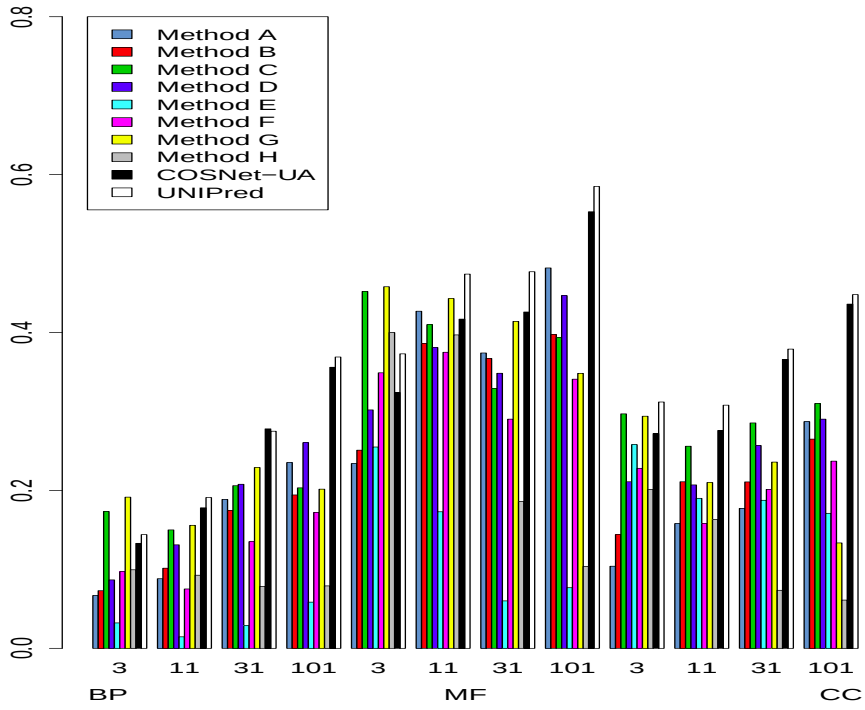
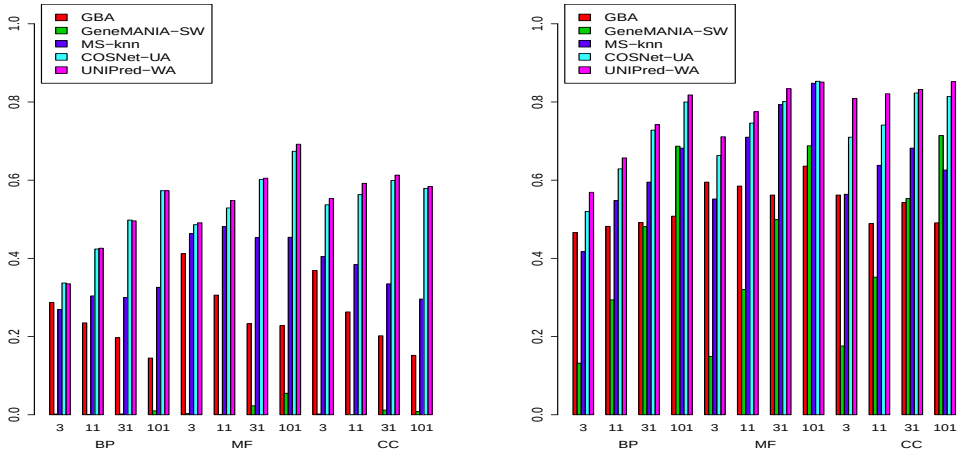


Figure 6: Comparison in terms of the F-score of the MouseFunc methods, *COSNet* applied to *UA* consensus network and *UNIPred*. Results are compared by GO categories.



(a) (b)

Figure 7: Yeast F-score (a) and P20R (b) results averaged by ontology and cardinality of annotations of GO terms.

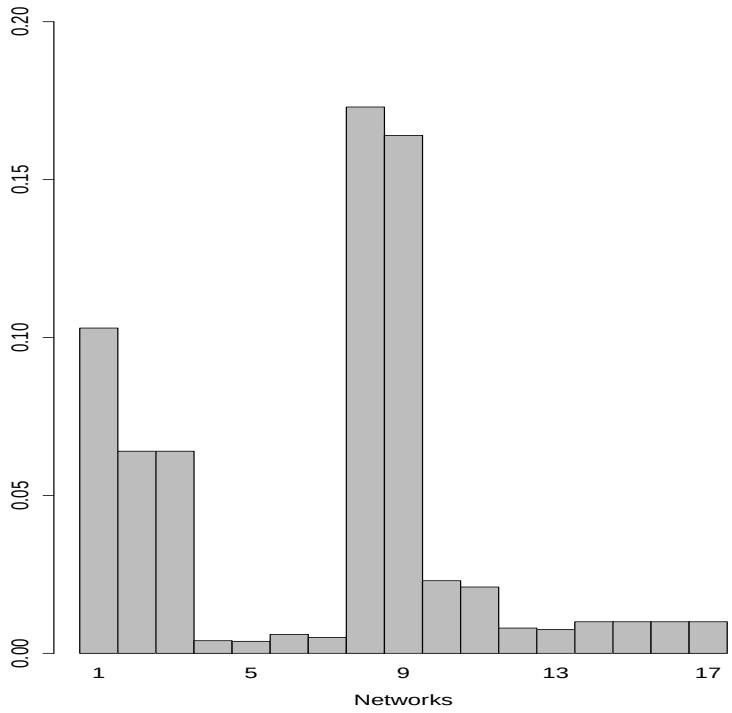


Figure 8: Averaged F-score across GO terms achieved by *COSNet* on each single mouse network. For the description of the 17 networks used in the experiments, please refer to Table S1 in the Supplementary Data.

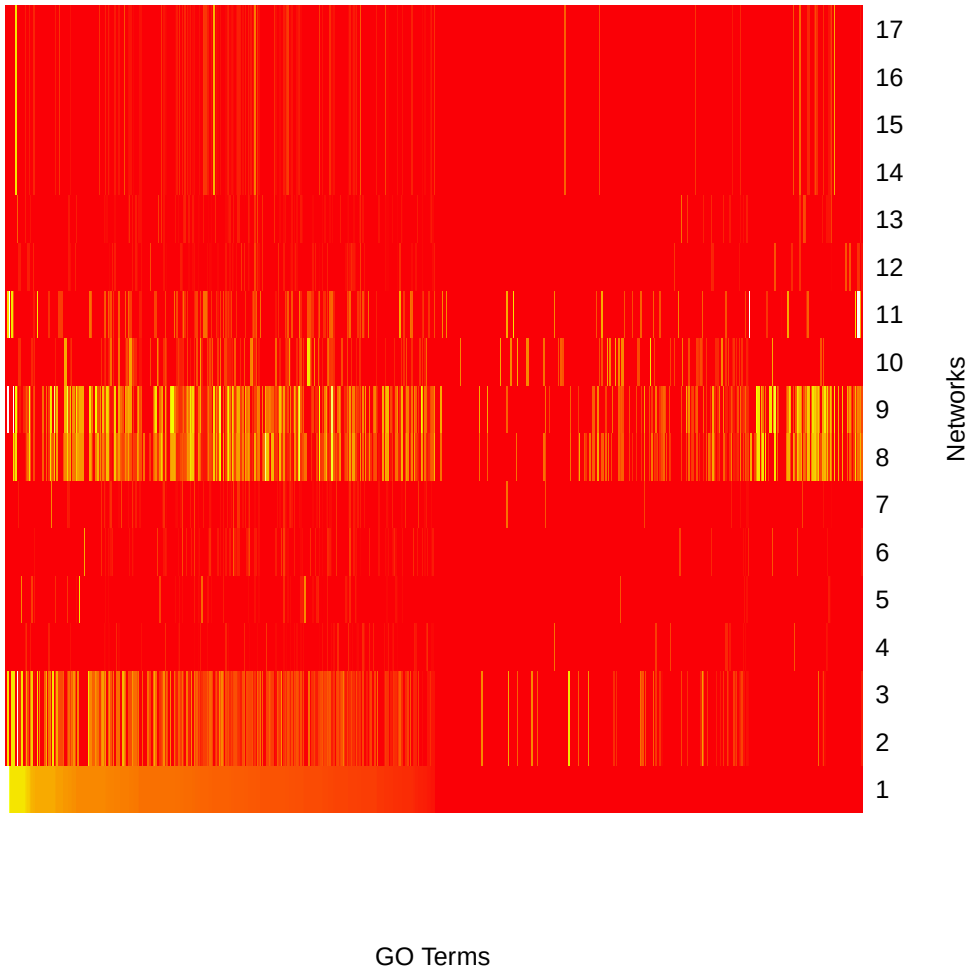


Figure 9: Heat map representing the F-score values obtained by *COSNet* when predicting on the single mouse networks. The lighter the color the higher the corresponding value of F-score. The columns represent the 650 considered GO terms with at least a positive annotation in test set, the rows the 17 mouse networks we considered.

List of Tables

Methods	**F-score**			**P20R**		
	BP	**MF**	**CC**	**BP**	**MF**	**CC**
Method A (Obozinski *et al.*, 2008)	0.113	0.340	0.163	0.204	0.470	0.284
Method B (Lee *et al.*, 2006)	0.113	0.328	0.197	0.204	0.469	0.334
Method C (Mostafavi *et al.*, 2008)	0.175	0.406	0.281	0.314	0.607	0.479
Method D (Barutcuoglu *et al.*, 2006)	0.140	0.346	0.229	0.320	0.591	0.423
Method E (Kim *et al.*, 2008)	0.028	0.170	0.208	0.209	0.492	0.366
Method F (Chen and Xu, 2004)	0.104	0.340	0.198	0.203	0.529	0.343
Method G (Tian *et al.*, 2008)	0.188	0.434	0.231	0.351	**0.653**	0.467
Method H (Qi *et al.*, 2007)	0.091	0.322	0.143	0.194	0.462	0.297
COSNet-UA	0.196	0.392	0.314	0.329	0.587	0.465
UNIPred	**0.205**	**0.443**	**0.342**	**0.356**	0.648	**0.494**

Table 1: Comparison of *UNIPred* with the 8 best methods of the MouseFunc challenge and *COSNet* applied to the UA consensus network. The best results for each ontology are shown in boldface. Results significantly better than all the other methods (Wilcoxon signed rank test, $\alpha = 0.01$) are underlined.

GO annotation	**F-score**		
	BP	**MF**	**CC**
2006 GO release	0.202	0.445	0.348
2012 GO release	0.265	0.436	0.339
	P20R		
	BP	**MF**	**CC**
2006 GO release	0.355	0.642	0.502
2012 GO release	0.426	0.629	0.522

Table 2: Accuracy of *UNIPred* mouse predictions compared between GO 2006 and GO 2012 annotations.

Methods	F-score			P20R		
	BP	MF	CC	BP	MF	CC
			Yeast			
GBA	0.247	0.354	0.300	0.478	0.590	0.536
GeneMANIA	0.002	0.008	0.004	0.278	0.263	0.322
MS-kNN	0.288	0.465	0.380	0.503	0.636	0.605
COSNet-UA	**0.406**	0.521	0.556	0.606	0.710	0.744
UNIPred	0.405	**0.530**	**0.574**	**0.641**	**0.750**	**0.819**
			Fly			
GBA	0.134	0.206	0.204	0.274	0.378	0.382
GeneMANIA	0.001	0.002	0.006	0.280	0.433	0.406
MS-kNN	0.179	0.381	0.260	0.335	0.552	0.471
COSNet - UA	0.251	0.437	0.407	0.374	0.586	0.562
UNIPred	**0.253**	**0.472**	**0.415**	**0.388**	**0.652**	**0.601**

Table 3: Results on yeast and fly organisms in terms of F-score and P20R averaged by GO ontology. The best results for each ontology are shown in boldface. Results significantly better than all the other methods (Wilcoxon signed rank test, $\alpha = 0.01$) are underlined.

Network	**GO term**				
	GO:0001759	GO:0003684	GO:0009165	GO:0001633	GO:0042102
1	0.400	0	0	0.667	0.4
2	0	0	0.2	0	0
3	0	0	0.2	0	0
4	0	0	0	0	0
5	0.667	0	0	0	0
6	0	0	0	0	0.5
7	0	0	0	0	0
8	0.444	0	0.2	0.667	0
9	0.5	0.25	0.75	0	0
10	0	0	0	0	0
11	0	1	0	0	0
12	0	0	0	0	0
13	0	0	0.333	0	0
14	0	0	0	0.667	0
15	0	0	0	0.667	0
16	0	0	0	0.667	0
17	0	0	0	0.667	0

Table 4: F-score achieved by *COSNet* for some selected GO terms.