Automated Gene Function Prediction through Gene Multifunctionality in Biological Networks

Marco Frasca

Dipartimento di Informatica, Università degli Studi di Milano
Via Comelico 39 Milano, 20137, Italy
frasca@di.unimi.it

Abstract
As the number of sequenced genomes rapidly grows, automated prediction of gene function (AFP) is now a challenging problem. Despite significant progresses in the last several years, the accuracy of gene function prediction still needs to be improved in order to be used effectively in practice. Two of the main issues of AFP problem are the imbalance of gene functional annotations and the ‘multifunctional properties’ of genes. While the former is a well studied problem in machine learning, the latter has recently emerged in bioinformatics and few studies have been carried out about it. Here we propose a method for AFP which appropriately handles the label imbalance characterizing biological taxonomies, and embeds in the model the property of some genes of being ‘multifunctional’. We tested the method in predicting the functions of the Gene Ontology functional hierarchy for genes of yeast and fly model organisms, in a genome-wide approach. The achieved results show that cost-sensitive strategies and ‘gene multifunctionality’ can be combined to achieve significantly better results than the compared state-of-the-art algorithms for AFP.

Keywords: Gene multifunctionality, biological networks, Hopfield networks, gene function prediction, gene ranking, cost-sensitive learning.

1. Introduction

High throughput biomolecular technologies have made available a vast amount of genomic, proteomic and transcriptomic data and the experimental determination of gene functions is the most reliable way to characterize genes and their products. However, due to its inherent difficulty and expense, the experimental characterization of functions cannot appropriately scale up and the automated annotation of gene functions has therefore emerged as a challenging problem in computational and molecular biology [1]. The Automated Prediction of gene Functions (AFP) is a complex problem, with several distinctive features: functional classes (biological functions) are structured in a hierarchy with different levels of specificity (e.g. the Gene Ontology (GO) [2]) and labelings are not
independent; each gene may have multiple labels (multi-label classification); classes are thousands (GO) and often highly unbalanced, with few positive and much more negative genes; negative instances are not uniquely defined, since usually only positive gene memberships are known for functional classes, and negatives in principle can be chosen with different strategies [3]; data are noisy and usually large-scale and high-dimensional; several heterogeneous sources of biological data are available, each one describing specific properties of genes, and, to achieve more reliable predictions, their integration with suitable methods is needed [4].

In this work we take into consideration two of these issues: the imbalance of class labelings and the multiple annotation of genes. Many attempts have been proposed in the literature for AFP. More general approaches characterize genes by a set of features, which in turn are exploited by machine learning algorithms to typically address a set of binary classification problems: predict whether or not a gene should be associated with a functional class [5]. Another commonly used approach is based on sequence homology, which adopts sequence alignment tool, e.g. BLAST [6], to find sequences of gene products (such as proteins) similar to the target sequence, and then transfers their known functional annotations to the target sequence as predictions [7, 8]. Moreover, the availability of large-scale networks of genetic and physical interactions, where nodes are genes/gene products and connections among nodes the gene pairwise relationships, has focused the investigation also on the design of network-based algorithms for AFP. The first network-based approaches have been based on the so called guilt-by-association (GBA) rule, which makes predictions based on the interacting genes, assuming that interacting genes are likely to share similar functions [9, 10, 11]. Indirect neighbours have also been exploited to modify the notion of pairwise-similarities among nodes by accounting for pairs of nodes connected through intermediate ones [12, 13, 14].

Furthermore, gene functions can be predicted by propagating node labels through the network with an iterative process until convergence [15, 16], by tuning the amount of propagation we allow in the graph through Markov Random Walks [17, 18], by evaluating the functional flow through the nodes [19]. Other relevant studies also adopted techniques based on Global graph consistency [20], on Hopfield networks [21, 22, 23], on Markov [24] and Gaussian Random Fields [25, 26, 27].

Despite their proved effectiveness, these methods totally or partially neglect two main issues of AFP. First, they do not appropriately handle the label imbalance affecting classes in biological taxonomies. The Gene Ontology is the most popular repository for biological functions and structures genes in three major ontologies (direct acyclic graphs): Molecular Function (MF), Biological Process (BP), and Cellular Component (CC). The most specific classes, which are those better describe the functions of genes, have usually very few annotations (genes that previous studies have shown having the function). This lack of information makes the prediction task very difficult, and cost-insensitive algorithms may suffer high decay in performance [28, 29]. Second, when predicting in a flat setting, i.e. without considering the hierarchical structure of GO, such methods do
not embed in their framework the multi-functional properties of genes. Indeed, recent works have introduced the concept of *gene multifunctionality* [30], which regards the property of some genes which are annotated with many classes of being really multifunctional in the cell cycle. The authors have shown that multifunctionality drives most computational predictions made by GBA-based methods, and, furthermore, that there exists a relationship between multifunctionality and the number of interacting partners in the network. Overall, multifunctionality has been investigated as a possible limitation of generalization capabilities of algorithms that infer gene functions exploiting solely the GBA rule, and some strategies have been suggested to prevent this limitation (e.g. avoiding the network sparsification). On the other hand, they disregard high gene degree may be a good indicator of the gene cell activity, and do not develop any strategy which exploits gene multifunctionality to improve the reliability of functional predictions.

In this work we propose an approach to cast gene multifunctionality in the prediction model of a network-based imbalance-aware algorithm, *COSNet* [23], recently proposed to predict node labels in partially labeled graphs. We analyzed biomolecular networks from model organisms to investigate the role multifunctionality has on the predictive capability of *COSNet*. Interestingly, we found that for almost all GO functions, the considered networks (that as suggested, we do not sparsify) contain several *exceptional genes*, which are genes annotated with the function *c* being predicted, without interacting partners annotated with *c*, but with high node degree (i.e. expected high multifunctionality). Such genes are likely to be wrongly predicted by most of all network-based AFP methods. Our strategy is designed to explicitly take into account the presence of exceptional genes and to exploit their multifunctionality to improve the accuracy of the prediction. The experimental validation carried out on two eukaryotic organisms in a genome-wide approach shows our method favourably compares with the state-of-the-art algorithms for AFP.

In the following, the AFP problem is formalized in Section 2, Section 3 introduces the multifunctionality in gene networks, whereas Sections 4.1 and 4.2 are dedicated to the description of *COSNet* and its extension to multifunctionality, respectively. The experimental validation of the proposed algorithm is discussed in Section 5.

2. Automated Function Prediction in Gene Networks

In the *Automated Function Prediction* (AFP) problem, genes are represented by a set of vertices *V*, and the relationships among genes are encoded in the symmetric matrix \( W : V \times V \rightarrow [0, 1] \), where \( W_{ij} \) is a precomputed measure of ‘functional similarity’ between genes *i*, *j* \( \in V \). For a given functional class *c* (e.g. a term of the Gene Ontology), a labeling function \( L_c : S \rightarrow \{ +, - \} \) is known, where \( S \subset V \) is the set of labeled vertices. Moreover, a bipartition \( (S_+, S_-) \) of \( S \) is given, where \( S_+ = \{ i \in S | L(i) = + \} \) is the set of positive vertices and \( S_- = \{ i \in S | L(i) = - \} \) the set of those negative.
The aim is to derive a score function $\psi : U \rightarrow \mathbb{R}$, which ranks unlabeled nodes according to the values of $\psi(i)$: the higher the score, the higher the likelihood that a gene belongs to the given functional class. $U = V \setminus S$ is the set of unlabeled vertices.

3. Multifunctionality in Gene Networks

The concept of ‘multifunctionality’ has been recently introduced in the scientific community to analyze the role ‘multifunctional genes’ have in the computational prediction of gene functions [30]. The gene multifunctionality can be defined as ‘the number of molecular functions a gene is involved in’, depending on the context and the interacting partners (other gene products). From a computational standpoint, multifunctionality is the number of classes an instance is classified as member.

Using GO as source of functional annotations for genes, Ranking by multifunctionality means assigning higher rank to genes annotated with more GO terms. Indeed, if a gene is involved in many biological functions, the degree to which the gene has also a chosen function is higher than another gene which is, for example, annotated just with one GO term. In other words, algorithms which assign new functions to genes which are already are annotated with many GO terms are expected to achieve good performance for the majority of functions. Gillis and Pavlidis [30] define the multifunctionality score of gene $i$ as follows:

$$Score_c(i) = \sum_{c \in GO} \frac{1}{I_c \times O_c}$$

where $I_c$ and $O_c$ are respectively the number of genes annotated and not annotated with term $c$. If we ignore the normalization by the product of $I_c$ and $O_c$, $Score_c(i)$ is simply the number of functions the gene $i$ has. This score provides a ranking which makes correct predictions for almost all the considered GO terms, achieving a mean AUC of 0.9. Unfortunately, when predicting a single term with a flat approach, this score cannot be computed. On the other hand, the authors have also shown that gene multifunctionality is related to the node degree in gene networks, that is the number of genes interacting in a particular context. A greater number of interaction partners reflects (at least partially) the involvement in the biomolecular functions the partners have (hence expected higher multifunctionality). They show that, if the data used for prediction is in some way a proxy for multifunctionality, and the algorithm used for classification can exploit this, very good prediction performance can result. Accordingly, the node (gene) degree can be assumed as suitable estimate of gene multifunctionality, and it can be exploited as prior knowledge to improve the predictive capabilities of network-based AFP methods.
4. Methods

In this section we first recall a recently proposed method for AFP, COSNet, designed to properly handle the class label imbalance, then we describe our approach to take into account the gene multifunctionality.

4.1. COSNet

COSNet (COst-Sensitive neural Network) [23] is a semi-supervised learning algorithm for predicting node labels in graphs with unbalanced labeling. COSNet is based on a family of parametric Hopfield networks $H = \langle W, k, \rho \rangle$ on neurons $V = \{1, 2, \ldots, n\}$, where $k$ is the neuron activation threshold and $\rho$ is a real number in $[0, \pi]$ that determines the two different values $\{\sin \rho, -\cos \rho\}$ for neuron activation. Node labels and neuron activation values are conceptually separated, and neuron activation values are now parameters to be learned to deal with data imbalance. A sketch of COSNet is given in the following (Figure 1):

INPUT: a connection strength matrix $W : V \times V \rightarrow [0, 1]$; the function $L_c : S \rightarrow \{+, -\}$; the sets $S$ and $U$ of respectively labeled and unlabeled instances w.r.t the functional class $c$ to be predicted. Up to a permutation, we assume $U = \{1, 2, \cdots, h\}$ and $S = \{h + 1, h + 2, \cdots, n\}$.

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

Figure 1: COSNet main steps.
Step 1. A temporary solution \((U_+, U_-)\) is generated such that \(\frac{|U_+|}{|U|} \approx \frac{|S_+|}{|S|} \).

Step 2. The optimal parameters \((\hat{\rho}, \hat{k})\) are estimated in order to make the state \(L_c(S)\) (state represented by known labels) “as close as possible” to an equilibrium state of the sub-network restricted to \(S\).

Step 3. The parameters \((\hat{\rho}, \hat{k})\) are extended to the whole network and the sub-network restricted to unlabeled nodes is simulated. Starting with initial value \(u_i(0) = 0\) for each neuron \(i\), the network evolves according to the following asynchronous dynamics:

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

(2)

where \(u_i(t)\) is the value of neuron \(i \in U\) at time \(t\). Here \(\theta_i = \hat{k} - \sum_{j=1}^{n} W_{ij} L_c(j)\) is the activation threshold of node \(i\), which also includes the influence on this node of the labeled neurons \(S\) (whose values are clamped during the network dynamics). At each time \(t\), the state of the network is \(u(t) = (u_1(t), u_2(t), \ldots, u_h(t))\), and a Lyapunov state function named energy function is associated to the network:

\[
 E(u) = -\frac{1}{2} \sum_{i,j=1}^{h} W_{ij} u_i u_j + \sum_{i=1}^{h} u_i \theta_i 
\]  

(3)

The dynamics converges to an equilibrium state \(\hat{u}\) corresponding to a minimum of \(E\) [23]. The final solution \((U_+, U_-)\) is:

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

The step 1 provides a temporary solution in order to exploit the connections among labeled and unlabeled nodes during learning phase. In step 2, each labeled node is projected into a labeled point in the plane, where the abscissa is the weighted sum of positive connections, the ordinate is the weighted sum of negative connections. In this way the unbalance at each point is embedded in the point position. Then, a fast quasi-linear approximated algorithm learns a parametric straight line to separate positive and negative points by maximizing a specific criterion that accounts for the label imbalance. The learned line provides the values \((\hat{\rho}, \hat{k})\) for the parameters \((\rho, k)\) to be adopted in the network dynamics described at Step 3.

\(\text{COSNet}\) is fast and nicely scale on large-size data, taking overall time \(O(|S| \log |S| + |W|)\), which is quasi-linear when the input connection matrix is sparse.
4.2. Multifunctionality-Based Ranking: COSNetM

In this section we describe COSNetM (COSNet Multifunctionality-based ranking), an algorithm for ranking genes to deal with AFP problem. COSNet is a binary classifier and it has not been designed to rank instances; nevertheless, in [31] COSNet has been adopted as ranker by assigning to each neuron a score related to its internal energy at equilibrium. More precisely, the score assigned to neuron \( i \in U \) is the following:

\[
 r(i) = \sum_{j \neq i} (W_{ij} \hat{u}_j - \theta_i) \tag{4}
\]

Reminding that the equilibrium state \( \hat{u} \) is a minimum of the energy \( E \), it is interesting to observe that the score (4) corresponds to a global and local consistency. Global because the dynamics allows to propagate the node labels through the network, so that neurons can get information also from non neighboring neurons, until an equilibrium of the whole network is reached. Local because if we consider the contribution \( E(\hat{u}_i) = -\hat{u}_i r(i) \) of node \( i \) to the energy \( E \), we can observe that for positive predictions \( (\hat{u}_i = \sin \hat{\rho}) \), the score \( r(i) \) is positive and the larger the presence of positives in its neighborhood (i.e. larger values of \( r(i) \)), the lower the value of the energy \( E(\hat{u}_i). \) For negative predictions \( (\hat{u}_i = -\cos \hat{\rho}) \), the score \( r(i) \) is negative and a larger presence of negative neighbors corresponds to lower values of \( E(\hat{u}_i). \)

Although this score has been shown being effective for ranking genes, there are some specific cases in which it may ‘fail’. Indeed, due to lack of knowledge and/or presence of noise both in the connection matrix \( W \) [32] and in GO annotations (for less studied terms usually only few annotations are available), some genes in the network, which we name ‘exceptional genes’ (EGs), may have the following properties:

(i) Being positive for the GO term to be predicted
(ii) No positive interacting partners in the network
(iii) High node degree

For such genes, prediction algorithms based solely on the functions of interacting partners definitely make wrong predictions. Nevertheless, the score (4) may correctly classify exceptional genes, since the network dynamics propagates labels through the network, and positive labels may come from neighbors at level two or more.

4.2.1. Distribution of Exceptional Genes

We analyzed the gene networks of yeast and fly organisms described in Section 5.1 and the corresponding GO annotations to detect the presence of EGs for each function separately. We considered 3419 and 4317 GO terms with a number of annotated genes ranging from 3 to 300, for 5775 yeast and 9361 fly genes. In both networks, a large number of EGs has been found: 986 for yeast and 5239 for fly data. Moreover, 759 and 2607 GO terms have at least one EG respectively in yeast and fly, with maximum rate of EGs equal to 1 for 12 GO
Table 1: Examples of exceptional yeast genes. ‘Rank by node degree’ is the position of the gene in the ranking given by node degrees.

<table>
<thead>
<tr>
<th>Gene</th>
<th>GO term</th>
<th>Positive annotations</th>
<th>Rank by node degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>YKR031C</td>
<td>GO:0016298</td>
<td>22</td>
<td>290</td>
</tr>
<tr>
<td>YOL027C</td>
<td>GO:0030004</td>
<td>35</td>
<td>74</td>
</tr>
<tr>
<td>YER120W</td>
<td>GO:0051224</td>
<td>4</td>
<td>172</td>
</tr>
<tr>
<td>YNL197C</td>
<td>GO:0051224</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>YER151C</td>
<td>GO:004583</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>YNL264C</td>
<td>GO:0006658</td>
<td>4</td>
<td>663</td>
</tr>
<tr>
<td>YOL011W</td>
<td>GO:0006658</td>
<td>4</td>
<td>182</td>
</tr>
<tr>
<td>YCR094W</td>
<td>GO:0031902</td>
<td>6</td>
<td>147</td>
</tr>
<tr>
<td>YGR270W</td>
<td>GO:0042406</td>
<td>4</td>
<td>445</td>
</tr>
<tr>
<td>YER120W</td>
<td>GO:0051051</td>
<td>5</td>
<td>172</td>
</tr>
<tr>
<td>YKR031C</td>
<td>GO:0004620</td>
<td>11</td>
<td>290</td>
</tr>
</tbody>
</table>

terms (3 EGs out of 3 positive genes). At condition (iii) of exceptional gene, we considered high node degree those degrees which rank in the top half of node degree ranking.

In Table 1 we report some examples of exceptional yeast genes and GO terms which have at least one EG. Interestingly, a gene may be ‘exceptional’ for more than one term (e.g. genes YKR031C, YER120W), and the same GO term may have more than one EG (for instance GO:0051224, GO:0006658). When no further information is given, the gene multifunctionality thereby represents a supplementary information which can be fundamental in correcting gene rankings, mainly for those EGs which result in a very high node degree rank (e.g. genes YNL197C, YER151C).

Furthermore, in order to understand how the EGs are distributed across the functional classes, and whether some classes are expected having more EGs, in Figure 2 we grouped the GO terms by number of positives, and then computed the averaged per group proportion of positive genes which are exceptional. We can observe a unimodal distribution, with modal peak (0.115 yeast and 0.188 fly) corresponding to the GO group with 3 positives. Interestingly, the average proportion of EGs is higher for more unbalanced terms, and tends to decrease when the number of positives increases; this is quite expected, since a real positive is likely to have no positive neighbors when few positives are available (condition (ii) of EG). This also suggests the complexity of predicting more unbalanced GO terms is also due to the higher proportion of EGs. Finally, we can observe a faster decreasing to 0 of EG proportion in yeast data w.r.t. fly data, probably due to the different label imbalance (fly organism has much more genes than yeast), leading us to hypothesize the existence of a direct relationship
Figure 2: Proportion of EGs on yeast (a) and fly (b) data averaged across groups of GO terms with the same number of positives (annotations).

4.2.2. COSNetM Ranking

The analysis of EG distribution allows to modify the score (4) in order to take into account the presence of exceptional genes:

$$\psi_1(i) = \frac{r(i)}{R} + \frac{d(i)}{D}$$

(5)

where $R = \sum_{i \in U} r(i)$, $D = \sum_{i \in U} d(i)$ and $d(i)$ is the degree of node $i$. Without normalization, the score (5) is simply the sum of node degree and node incoming functional contribution at equilibrium. For exceptional genes, the score $r(i)$ is likely to be negative, but the term $\frac{d(i)}{D}$ compensates this lack of information and moves up the gene in the ranking, since, by definition of exceptional gene, $d(i)$ is large. The proposed ranking function thereby ensures a better ranking of EGs. On the other hand, we want also to investigate how this score ranks non-exceptional genes, and the following cases are possible:

(a) $i$ is positive and $r(i)$ is positive

(b) $i$ is negative and $r(i)$ is negative

(c) $i$ is negative and $r(i)$ is positive

In the case (a), since $d(i) \geq 0$, we have $\psi_1(i) > 0$, i.e. the instance $i$ is correctly classified. In the case (b), $\psi_1$ is likely to be negative (correct classification), but it may happen that the score becomes positive when the instance $i$ has many connections. Finally, the instance $i$ is misclassified by function $r$ in the case (c),
and even $\psi_1(i) > 0$. However, when $i$ has a low degree, $\psi_1$ moves down the instance $i$ in the ranking w.r.t. $r$.

Overall, the ranking function (5) is able in correcting the ranking of exceptional genes, thus satisfying our initial purpose, and for non exceptional genes, only in case (b) it may misclassify an instance correctly classified by score $r$. Moreover this ranking can also exploit the properties of the equilibrium state of the network to provide class-specific rankings. It is worth also noting that the proposed ranking function exploits solely the information deriving from node degree, although in principle each neighbor may be in turn multifunctional. Having neighbors which are expected being multifunctional, provides further information about the multifunctionality of the gene. Accordingly, we want to modify the equation (5) to take into account also the degrees of level-two neighbors as follows:

$$\psi_2(i) = \frac{r(i)}{R} + \frac{dn(i)}{DN}$$

where $dn(i) = W_i \cdot d$, $\cdot$ is the dot product, $W_i$ is the vector of connection weights of node $i$, $d$ is the vector of node degrees and $DN = \sum_{i \in U} dn(i)$. It easy to see that this score preserves the properties described for the ranking function (5), since high values of $d(i)$ in general correspond to high values of $dn(i)$.

5. Results and Discussion

5.1. Experimental Setup

To validate our approach, COSNetM has been applied in predicting functions of the whole genome of two model eukaryotic organisms. 16 S. cerevisiae and 10 D. melanogaster networks downloaded from the GeneMANIA website (www.genemania.org) have been integrated through unweighted sum, considering the union of genes in the single networks. The selected networks cover different types of data, including co-expression, genetic interactions, protein ontologies and physical interactions. The details about selected networks are reported in Tables 2 and 3. After integrating networks, we obtain a total of 5775 yeast and 9361 fly genes. No preprocessing has been applied to single networks, since GeneMANIA networks already provide a real score for each pair of genes representing a measure of their functional similarity. Each network, denoted by the corresponding connection matrix $W$, has been then normalized as follows:

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

where $D$ is a diagonal matrix and $d_{ii} = \sum_j W_{ij}$ its diagonal elements.

The Gene Ontology terms have been adopted as functional classes: GO annotations release 23-3-13 for yeast and 15-5-13 for fly. In order to predict the more specific (and thus more unbalanced) terms in the ontology, we selected all the GO terms with $3 \leq 300$ positive annotated genes, obtaining 3419 terms (2021, 805 and 593 respectively for BP, MF and CC ontologies) for yeast and 4317 terms for fly (2769 BP, 1004 MF and 544 CC).
Table 2: Yeast networks description.

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-expression</td>
<td>Busti et al. [33]</td>
<td>5436</td>
</tr>
<tr>
<td>Co-expression</td>
<td>Chin et al. [34]</td>
<td>5585</td>
</tr>
<tr>
<td>Co-expression</td>
<td>Beln Sanz et al. [35]</td>
<td>5585</td>
</tr>
<tr>
<td>Co-expression</td>
<td>Kovacs et al. [36]</td>
<td>5585</td>
</tr>
<tr>
<td>Genetic interactions</td>
<td>Aguilar et al. [37]</td>
<td>321</td>
</tr>
<tr>
<td>Genetic interactions</td>
<td>Alamgir et al. [38]</td>
<td>90</td>
</tr>
<tr>
<td>Genetic interactions</td>
<td>Costanzo et al. [39]</td>
<td>4346</td>
</tr>
<tr>
<td>Genetic interactions</td>
<td>Libuda and Winston [40]</td>
<td>143</td>
</tr>
<tr>
<td>Genetic interactions</td>
<td>BioGRID [41]</td>
<td>4280</td>
</tr>
<tr>
<td>Physical interactions</td>
<td>Breitkreutz et al. [42]</td>
<td>887</td>
</tr>
<tr>
<td>Physical interactions</td>
<td>Kaake et al. [43]</td>
<td>332</td>
</tr>
<tr>
<td>Physical interactions</td>
<td>Muller et al. [44]</td>
<td>266</td>
</tr>
<tr>
<td>Physical interactions</td>
<td>Ossareh-Nazari et al. [45]</td>
<td>406</td>
</tr>
<tr>
<td>Physical interactions</td>
<td>BioGRID [41]</td>
<td>4752</td>
</tr>
<tr>
<td>Shared protein domains</td>
<td>InterPro [46]</td>
<td>3964</td>
</tr>
<tr>
<td>Shared protein domains</td>
<td>Pfam [47]</td>
<td>3541</td>
</tr>
</tbody>
</table>

Table 3: Fly networks description.

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-expression</td>
<td>Baradaran-Heravi et al. [48]</td>
<td>8857</td>
</tr>
<tr>
<td>Co-expression</td>
<td>Busser et al. [49]</td>
<td>8857</td>
</tr>
<tr>
<td>Co-expression</td>
<td>Colombani et al. [50]</td>
<td>8857</td>
</tr>
<tr>
<td>Co-expression</td>
<td>Lundberg et al. [51]</td>
<td>8857</td>
</tr>
<tr>
<td>Genetic interactions</td>
<td>BioGRID [41]</td>
<td>929</td>
</tr>
<tr>
<td>Physical interactions</td>
<td>Yu et al. [52]</td>
<td>1414</td>
</tr>
<tr>
<td>Physical interactions</td>
<td>Guruharsha et al. B [53]</td>
<td>3833</td>
</tr>
<tr>
<td>Physical interactions</td>
<td>BioGRID [41]</td>
<td>558</td>
</tr>
<tr>
<td>Shared protein domains</td>
<td>InterPro [46]</td>
<td>5627</td>
</tr>
</tbody>
</table>
5.2. Results

**COSNetM** has been compared with the state-of-the-art methods proposed in the literature for the gene function prediction problem in a “flat” setting. In particular, we considered GeneMANIA [26], an algorithm based on ridge regression integration and Gaussian Random Fields, that ranked among the best methods in the MouseFunc competition for mouse AFP [54], and, as a baseline, the classical guilt-by-association algorithm (GBA) [55]. We also evaluated a classical inductive method, the Support Vector Machine (SVM), largely applied in computational biology and in AFP; more precisely, we tested the probabilistic version of SVM [56], which provides a probabilistic score to genes with respect to the functional class being predicted. Moreover, since we extended COSNet, we also consider its original version, which ranks instances through the ranking function (4) presented in Sect. 4.2. Finally, we also report the results achieved by simply ranking genes by their node degree.

To estimate the generalization performances of the considered methods, we adopted a classical 10-fold cross-validation and we applied the Wilcoxon signed-ranks test [57] to compare the overall results. The performances have been assessed using the Area Under the ROC Curve (AUC) and the Area Under the Precision-Recall Curve (AUPRC), pointing out that, unlike AUPRC, AUC is not properly suitable for classes highly unbalanced toward negatives, like those characterizing AFP.

5.2.1. Comparing Multifunctional Scores

In order to assess the contribution of the gene multifunctionality to the predictive capability of COSNet, in Table 4 we report the results of COSNet and COSNetM (ranking functions \(\psi_1\) and \(\psi_2\)) averaged by GO ontology. First of all, the ranking function \(\psi_2\) outperforms \(\psi_1\) (Wilcoxon test, p-value < 10\(^{-5}\)) in all the data sets and in terms of both AUC and AUPRC, except for AUC values on yeast data in MF and CC ontologies, where the difference is not statistically significant. A severe improvement in AUC is registered for yeast data in BP and CC ontologies, suggesting that the multifunctionality contribution computed by accounting also neighbor degrees (exploited by \(\psi_2\)) is more informative than the simple node degree. When comparing with COSNet, COSNetM achieves significantly better performance (Wilcoxon test, p-value < 10\(^{-6}\)) in all the performed experiments, considerably improving the AUPRC on yeast data. Particularly interesting are the results w.r.t COSNet that COSNetM achieves in terms of AUC on BP and CC terms when predicting fly genes, that are clearly due to the multifunctionality information we embedded in the model. This confirms that the node degree is a good estimator of multifunctionality and that, although node degree scores alone do not achieve good results (see Section 5.2.2), the combination of node degree and internal energy at equilibrium improves the overall performance. To better understand the effect of multifunctionality, in Figure 3 we also report the per class differences in terms of both AUC and AUPRC for COSNet and COSNetM. Each point corresponds to a GO term. The terms are increasingly sorted according to the number of annotations (positives). A positive difference (point above the line) means better performance...
Table 4: COSNet and COSNetM performance averaged across each GO ontology.

<table>
<thead>
<tr>
<th>Method</th>
<th>YEAST</th>
<th>FLY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BP</td>
<td>MF</td>
</tr>
<tr>
<td>COSNet</td>
<td>0.845</td>
<td>0.891</td>
</tr>
<tr>
<td>COSNetM-$\psi_1$</td>
<td>0.883</td>
<td>0.907</td>
</tr>
<tr>
<td>COSNetM-$\psi_2$</td>
<td>0.892</td>
<td>0.907</td>
</tr>
</tbody>
</table>

for COSNetM, the difference is negative (point below the line) when COSNet outperforms COSNetM. First, for both AUC and AUPRC, the large majority of points lies above the line, and mainly for AUPRC results in both yeast and fly organisms. Second, the absolute difference tends to be larger for more unbalanced classes, and lower when the number of positives increases. This is likely due to the fact that for more unbalanced terms the expected proportion of positives which are EGs is higher (see Figure 2). Moreover, most of all the differences become positive when the number of positive increases. The negative differences (i.e. COSNet performs better) may depend on the effect the score $\psi_2$ has on non-exceptional genes (see case (b) described in Section 4.2.2). However, the number of points below the line is small, confirming the average improvements shown in Table 4. Finally, and mainly for AUPRC results, the larger improvements for COSNetM are related to the most specific terms, which is of preeminent importance to lead biologists in the analysis of the functions that better characterize the functional role of genes.

As final investigation, we also want to verify whether COSNetM is able in improving the ranking of exceptional genes, thus in Table 5 we report the ranks assigned by COSNet and COSNetM to EGs considered in Table 1. As expected, COSNetM ranks EGs better than COSNet, and the improvement is noticeable for some genes, e.g. YER151C and YNL197C, which are respectively ranked in position 5733 and 5678 (almost at the bottom) by COSNet and in position 21 and 42 (almost at the top) by COSNetM. Confirming the results averaged across all the terms, in the majority of classes the score function $\psi_2$ assigns higher ranks to EGs than $\psi_1$: however, in some cases (for instance gene YER120W for GO term GO:0051051) $\psi_1$ ranks better than $\psi_2$, probably for those EGs whose neighbors are not highly multifunctional. Finally, due to its dynamics which allows nodes to propagate their labels to nodes at more than one-edge distance, also COSNet sometimes correctly ranks EGs (see gene YGR270W for GO term GO:0042406).
Figure 3: COSNet and COSNetM per term differences ($\Delta$) in terms of AUC ((a) yeast, (c) fly) and AUPRC ((b) yeast, (d) fly). Each point corresponds to one of the considered GO terms, and terms are increasingly sorted by number of positive annotated genes. Points above the vertical line correspond to better performance for COSNetM.
Table 5: Examples of rankings assigned by COSNet, COSNetM-$\psi_1$ and COSNetM-$\psi_2$ to exceptional yeast genes.

<table>
<thead>
<tr>
<th>Gene</th>
<th>GO term</th>
<th>COSNet</th>
<th>COSNetM-$\psi_1$</th>
<th>COSNetM-$\psi_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>YKR031C</td>
<td>GO:0016298</td>
<td>5643</td>
<td>4584</td>
<td>2718</td>
</tr>
<tr>
<td>YOL027C</td>
<td>GO:0030004</td>
<td>5748</td>
<td>5652</td>
<td>5172</td>
</tr>
<tr>
<td>YER120W</td>
<td>GO:0051224</td>
<td>4607</td>
<td>351</td>
<td>605</td>
</tr>
<tr>
<td>YNL197C</td>
<td>GO:0051224</td>
<td>5678</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>YER151C</td>
<td>GO:0048583</td>
<td>5733</td>
<td>23</td>
<td>21</td>
</tr>
<tr>
<td>YNL264C</td>
<td>GO:0006658</td>
<td>2645</td>
<td>1019</td>
<td>1160</td>
</tr>
<tr>
<td>YOL011W</td>
<td>GO:0006658</td>
<td>1264</td>
<td>358</td>
<td>490</td>
</tr>
<tr>
<td>YCR094W</td>
<td>GO:0031902</td>
<td>5562</td>
<td>967</td>
<td>735</td>
</tr>
<tr>
<td>YGR270W</td>
<td>GO:0042406</td>
<td>149</td>
<td>157</td>
<td>138</td>
</tr>
<tr>
<td>YER120W</td>
<td>GO:0051051</td>
<td>4903</td>
<td>377</td>
<td>601</td>
</tr>
<tr>
<td>YKR031C</td>
<td>GO:0004620</td>
<td>5658</td>
<td>4365</td>
<td>2494</td>
</tr>
</tbody>
</table>

5.2.2. Comparison with State-of-the-art AFP Methods

In Figure 4 we show AUC and AUPRC results averaged across GO BP, MF and CC ontologies also for the other AFP methods. The results in terms of AUPRC show that COSNetM largely outperforms the compared methods in both yeast and fly organisms, and the difference is always statistically significant (Wilcoxon test, $p$-value $< 10^{-33}$). Moreover, GBA outperforms GeneMANIA and SVM on yeast data, whereas on fly data GeneMANIA is the second best method. As expected, node degree is the worst method, since it assigns the same gene ranking to all the GO terms. Nevertheless, in terms of AUC the node degree algorithm has a certain learning, achieving a mean AUC $> 0.57$ in all the considered experiments, which is significantly different than 0.5 (random ranking) and confirming the results shown in [30]. COSNetM also achieves the best AUC values ($p$-value $< 10^{-7}$) in all the ontologies and organisms, except for MF terms on yeast data. GBA is the best method when predicting yeast genes in MF ontology (but no significant difference with COSNetM), and the second best in BP and CC ontologies. On fly data, GeneMANIA is the second top method when predicting BP and CC terms, whereas GBA is the second method in the MF ontology. Finally, the SVM algorithm poorly performs in almost all the experiments, even worse in terms of AUC than Node Degree rankings.

6. Conclusions

The automated protein function prediction is a challenging problem. In this paper we focused on the investigation of the labeling imbalance existing in functional taxonomies and on the ‘multifunctional’ properties characterizing genes.
in biomolecular networks. We propose a strategy to cast ‘gene multifunctionality’ in a parametric Hopfield Network designed to appropriately handle the label imbalance when predicting gene functions. The method has been tested in a genome-wide approach on yeast and fly model organisms in predicting functions of the Gene Ontology hierarchy, favorably comparing with the state-of-the-art methods for AFP. Moreover, the improvements of our method are larger on the most specific functions in the hierarchy, which are those better describe gene functions, and this is extremely important for biologist to suggest and guide the expensive laboratory experiments for verifying the involvement of gene activities in specific biomolecular functions.

We point out that our method purposely predicts functions without considering the structure of the Gene Ontology (direct acyclic graph) and thus the functional relationship among functions. Embedding in the model the information represented by the functional hierarchy may likely lead to higher prediction accuracy, as shown in recent studies [58].
References


[33] S. Busti, et al., Overexpression of far1, a cyclin dependent kinase inhibitor, induces a large transcriptional reprogramming in which rna synthesis senses far1 in a sfp1-mediated way, Biotechnology Advances 30 (1) (2012) 185–201.


[35] A. Beln Sanz, et al., Chromatin remodeling by swi/snf complex is essential for transcription mediated by the yeast cell wall integrity mapk pathway, Molecular Biology of the Cell.


Marco Frasca received the M.Sc degree in Computer Science from Salerno University, Italy in 2005 and the Ph.D. degree in Computer Science from Milan University, Italy in 2012. He is currently a research fellow at Computer Science department of Milan University. His research interests include gene function and expression prediction, with main focus on network-based model for the analysis of biomolecular networks.