Elsevier Editorial System(tm) for Pattern Recognition Letters

Manuscript Draft

Manuscript Number:

Title: Protein Function Prediction as a Graph-Transduction Game

Article Type: SI:AppGTPR

Keywords: Protein function prediction; graph transduction; game theory

Corresponding Author: Dr. Sebastiano Vascon,

Corresponding Author's Institution: Ca' Foscari University of Venice

First Author: Sebastiano Vascon

Order of Authors: Sebastiano Vascon; Marco Frasca, PhD; Rocco Tripodi, PhD; Giorgio Valentini; Marcello Pelillo

Pattern Recognition Letters

Authorship Confirmation

Please save a copy of this file, complete and upload as the "Confirmation of Authorship" file.

As corresponding author I, SEBASTIANO VASCON, hereby confirm on behalf of all authors that:

1. This manuscript, or a large part of it, has not been published, was not, and is not being submitted to any other journal.
2. If presented at or submitted to or published at a conference(s), the conference(s) is (are) identified and substantial justification for re-publication is presented below. A copy of conference paper(s) is(are) uploaded with the manuscript.
3. If the manuscript appears as a preprint anywhere on the web, e.g. arXiv, etc., it is identified below. The preprint should include a statement that the paper is under consideration at Pattern Recognition Letters.
4. All text and graphics, except for those marked with sources, are original works of the authors, and all necessary permissions for publication were secured prior to submission of the manuscript.
5. All authors each made a significant contribution to the research reported and have read and approved the submitted manuscript.

Signature Sebastiano Vascon Date 15/10/2017

List any pre-prints:

A small portion of this work has been submitted as a four-page extended abstract at the Workshop on Machine Learning in Computational Biology which will be held at the NIPS 2017.
The paper will be also submitted to arXiv clearly stating that is under consideration at *Pattern Recognition Letter*

Relevant Conference publication(s) (submitted, accepted, or published):

Justification for re-publication: The extended abstract submitted at the NIPS workshop (which is not supposed to be published in an archival form), is only a small fraction of the entire paper submitted here, and omits altogether all the technical details of the proposed work and presents only a subset of the experimental results.

Research Highlights (Required)

In this work we propose a graph-based method that:

- for the first time models the protein function prediction as a graph-transduction game
- offers a newer perspective to the automatic protein function prediction problem
- exploit at the same time the similarities at level of protein and functionalities
- has better overall performances compared to other state-of-the-art graph-based methods
- has been massively tested on 5 organisms, 3 ontologies with thousands of classes

Pattern Recognition Letters
journal homepage: www.elsevier.com

Protein Function Prediction as a Graph-Transduction Game

Sebastiano Vascon[a,c,**], Marco Frasca[b], Rocco Tripodi[a], Giorgio Valentini[b], Marcello Pelillo[a,c]

[a]*DAIS, Ca' Foscari University, Via Torino 155, 30172, Venezia Mestre, Italy*
[b]*AnacletoLab, Dipartimento di Informatica, Università degli Studi di Milano, Via Comelico 39, 20135 Milano, Italy*
[c]*ECLT, Ca' Foscari University, San Marco 2940, 30124, Venice, Italy*

ABSTRACT

Motivated by the observation that network-based methods for the automatic prediction of protein functions can greatly benefit from exploiting both the similarity between proteins and the similarity between functional classes (as encoded, e.g., in the Gene Ontology), in this paper we propose a novel approach to the problem which is based on the notion of a "graph transduction game." We envisage a (non-cooperative) game, played over a graph, where the players (graph vertices) represent proteins, the functional classes correspond to the (pure) strategies, and protein- and function-level similarities are combined into a suitable payoff function. Within this formulation, Nash equilibria turn out to provide consistent functional labelings of proteins, and we use classical replicator dynamics from evolutionary game theory to find them. To test the effectiveness of our approach we conducted experiments on five different organisms and three ontologies, and the results obtained show that our approach compares favorably with state-of-the-art algorithms.

1. Introduction

The Automatic Function Prediction of proteins (AFP) consists in the computational assignment of the biological functions to the proteins of an organism (Friedberg, 2006). It can be modeled as a multi-label classification task, since each protein may be associated with multiple functions, and represents one of the most challenging problems in the context of computational biology (Cesa-Bianchi et al., 2012; Radivojac et al., 2013; Jiang et al., 2016). The increasing availability of large-scale networks constructed from high-throughput biotechnologies, representing functional similarities between proteins, such as co-expression networks, protein domain similarities, and protein-protein interactions just to mention a few, opened the avenue of a large class of graph-based algorithms, able to learn from the functional similarities between proteins (Sharan et al., 2007).

These methods are able to transfer annotations from previously annotated (labeled) nodes to unannotated (unlabeled) ones through a learning process inherently transductive in nature, by exploiting the so-called guilt-by-association principle (Oliver, 2000), known also as homophily principle, by which proteins topologically close in the graph are likely to share their functions. Starting from simple approaches based on local learning strategies (Mayer and Hieter, 2000), several other methods have been proposed in literature, able to exploit in different ways the overall topology of the functional network. Some examples are represented by label propagation algorithms based on Markov (Deng et al., 2004) and Gaussian Random Fields (Zhu et al., 2003; Zhou et al., 2004; Mostafavi et al., 2008), methods that integrate local learning strategies with simple weighted combination of diverse information (Chua et al., 2007), approaches based on the evaluation of the functional flow in graphs (Vazquez et al., 2003), algorithms based on Hopfield networks (Karaoz et al., 2004; Frasca et al., 2015), methods that exploit relationships between homologous proteins to connect networks of different species (Mitrofanova et al., 2011), while other approaches applied random walk based methods (Lovász, 1996; Kohler et al., 2008) and their kernelized version by exploiting both local and global learning strategies (Re et al., 2012; Valentini et al., 2016).

Despite their large diversity, network-based methods share the common property of using some notion of similarity be-

**Corresponding author: Tel.: +39-041-234-7594; fax: +39-041-234-7589;
e-mail: sebastiano.vascon@unive.it (Sebastiano Vascon), marco.frasca@unimi.it (Marco Frasca), rocco.tripodi@unive.it (Rocco Tripodi), valentini@di.unimi.it (Giorgio Valentini), pelillo@unive.it (Marcello Pelillo)

tween proteins to learn protein functions. The underlying assumption is that *similar* proteins tend to share *the same* functional class, an idea which is reminiscent of the homophily principle widely used in social network analysis (Easley and Kleinberg, 2010) and which lies at the heart of virtually all classification algorithms.

This general approach has well-founded biological motivation (Sharan et al., 2007), but also the similarity between functional classes (i.e. the Gene Ontology – GO terms to be predicted) plays a key role in the prediction of protein functions, as outlined by the recent CAFA2 (Critical Assessment of Functional Annotation) challenge for the AFP problem (Jiang et al., 2016), since GO terms are not independent, but hierarchically related according to a directed acyclic graph (Gene Ontology Consortium, 2013). To our knowledge no network-based method has been proposed in the context of AFP to jointly consider the similarity between the proteins and the similarity between functional classes. We hypothesize that network-based methods could significantly enhance their performance if they were able to contextually learn from both similarity between the examples (the proteins) and the similarity between the GO terms associated with the proteins their selves. This corresponds to the well-known biological principle for which a protein is fully characterized by the entire spectrum of its structural and functional properties, coded as a set of GO terms (Gene Ontology Consortium, 2013).

Motivated by this observation, in this paper we present an application to AFP of a graph transduction model based on game-theoretic principles that conforms to a general classification principle which, assuming the existence of a notion of similarity not only at the object but also at the category level, prescribes that *similar objects* should be assigned to *similar categories*. This is in fact a generalization of the standard homophily principle which suggests instead that similar objects should be placed *in the same* category.

Along the lines set forth in Erdem and Pelillo (2012) within a standard homophily-based transductive setting, which ignored potential category-level similarities altogether, the AFP problem will be abstracted in terms of a multi-player non-cooperative game where the players represent proteins, the functional classes correspond to the (pure) strategies, and protein- and function-level similarities are combined in a suitable payoff function. Within this formulation, the Nash equilibrium concept for non-cooperative games turns out to offer a principled solution to the problem of finding a "consistent" labeling assignment (Hummel and Zucker, 1983; Miller and Zucker, 1991).[1] In order to find Nash equilibria of our AFP games we use (multi-population) replicator dynamics, a well-known class of dynamical systems developed and studied in evolutionary game theory (Weibull, 1995).

Our approach gives us the possibility not only to exploit the contextual information of a protein but also to find the most appropriate functions for the proteins in a determined context. In other words, the proposed model exploits two different kinds of information: structural and semantic. Structural information identifies how the proteins are organized in an organism, semantic information identifies how the functions of the proteins are structured. The integration of these two sources of information in a game theoretic model gives us the possibility to predict the combination of functions that are more suited for the proteins of a given organism. This is the most important methodological contribution of our work, which distinguishes it from existing AFP network-based algorithms.

To assess the effectiveness of the proposed game-theoretic approach, we conducted extensive experiments over different model organisms and using the ontologies of the GO, including thousands of functional classes and predictions for tens of thousands of proteins. We found that our proposed algorithms systematically obtain prediction results that are competitive with respect to state-of-the-art network-based methods for protein function prediction.

[1] See Kleinberg and Tardos (2002) for a different approach based on MRF's.

2. Graph Transduction and Non-Cooperative Games

2.1. Graph Transduction

Graph transduction is a semisupervised learning technique that aims at estimating a classification function defined over a graph of labeled and unlabeled data points. Models based on this technique use a graph to represent the data, with nodes corresponding to labeled and unlabeled points and edges encoding the pairwise similarity among each pair of nodes. This technique works propagating the label information from labeled nodes to unlabeled, exploiting the graph structure.

It was introduced by Vapnik (1998) and motivated by the fact that it is easier than inductive learning, because inductive learning tries to learn a general function to solve a specific problem, while transductive learning tries to learn a specific function for the problem at hand.

Graph transduction consists of a set of labeled objects (x_i, y_i) $(i = 1, 2, ..., l)$, where $x_i \in \mathbb{R}^n$ the real-valued vector describing the object i, and $y_i \in (1, ..., m)$ its label, for $i \in \{1, 2, \ldots, n\}$, and a set of k unlabeled objects $(x_{l+1}, ..., x_{l+k})$. Rather than finding a general rule for classifying future examples, transductive learning aims at classifying only (the k) unlabeled objects exploiting the information derived from labeled ones.

Within this framework it is common to represent the geometry of the data as a weighted graph. For a detailed description of algorithms and applications on this field of research, named graph transduction, we refer to (Zhu, 2005). Formally we have a graph $G = (V, E, w)$ in which V is the set of nodes representing both labeled and unlabeled points, E is the set of edges connecting the nodes of the graph and $w : E \to \mathbb{R}_{\geq 0}$ is a weight function assigning a non-negative similarity value to each edge $\epsilon \in E$. The task of transduction learning is to estimate the labels of the unlabeled points given the pairwise similarity among the data points and a set of possible labels.

In this article, we follow the approach proposed in Erdem and Pelillo (2012) that interprets the graph transduction task as a non-cooperative multiplayer game. This choice is motivated by the fact that this approach is inherently multiclass and for this reason it perfectly adapts to the AFP problem, as

defined in previous section. Furthermore it has a solid mathematical foundation rooted in game theory and it does not impose any constraint on the pairwise similarity function used to weight the graph. Classical graph transduction algorithms are based on the homophily principle (Joachims, 2003; Zhu et al., 2003; Zhou et al., 2004), that simply states that *similar data points* are expected to have the *same class*. We found this assumption too strong for the AFP task and for this reason we extended it using the approaches proposed in (Tripodi and Pelillo, 2017; Tripodi et al., 2016) that is reminiscent of the Hume association principle (Hume, 2000), that states that *similar objects* are expected to have similar properties and hence to belong to *similar classes*. With this approach we are able to exploit two sources of information: the similarity among the data points, as in classical graph transduction approaches and the similarity among their classes. With the latter source of information it is possible to build structural classifiers that produce consistent labeling of the data according to information provided by an ontology where it is encoded the information about the classes and their reciprocal relations. This turns out to be very useful in the context of classification of relation data. Imagine, for example, the case in which you want to classify the functional parts of an object, you do not want to assign to them the same class, just because they are functionally related (e.g.: the wheel and the dumper of a car) but you want to assign to them two coherent (similar) classes, as encoded in a knowledge base. We will see in Sections 3 that this information can be easily embedded in a game-theoretical framework as part of the payoff function but before we need to introduce some concepts of game theory in the next section.

2.2. Game Theory

Game theory (GT) was introduced by Von Neumann and Morgenstern (1944) in order to develop a mathematical framework able to model the essentials of decision making in interactive situations. In its *normal-form* representation, it consists of a finite set of players $I = \{1, .., n\}$, a set of pure strategies for each player $S = \{s_1, ..., s_m\}$, and a utility function $u : S_1 \times S_2 ... \times S_n \rightarrow \mathbb{R}$, which associates strategies to payoffs. Here we assume that all the players have the same set of strategies S, but in the more general formulation this is not mandatory. Each player can adopt a strategy in order to play a game and the utility function depends on the combination of strategies played at the same time by the players involved in the game, not just on the strategy chosen by a single player. An important assumption in game theory is that the players try to maximize their utility u. Furthermore, in *non-cooperative games*, the players choose their strategies independently, considering what other players can play in order to find the best strategy profile to employ in a game.

Nash Equilibria (NE) (Nash, 1951) represent the key concept of game theory and can be defined as those strategy profiles in which each strategy is the best response to the strategy of the co-player and in which no player has the incentive to unilaterally deviate from his decision (the players are in equilibrium). The NE of a game exist in two forms: *i)* pure-strategy and *ii)* mixed-strategy. In a pure-strategy NE each player adopts only one strategy while in the latter case is a probability distribution among the possible strategies. A mixed strategy for a player is defined as a stochastic column vector $\mathbf{x} = (x^1, \ldots, x^m) \in \Delta^m$, where m is the number of pure strategies and each component x^h denotes the probability that a particular player chooses its h-th pure strategy. Each mixed strategy corresponds to a point in the m-dimensional simplex Δ^m defined as,

$$\Delta^m = \Big\{x \in \mathbb{R} : \sum_{h=1}^{m} x^h = 1, x^h \geq 0, \forall h\Big\}, \tag{1}$$

whose corners correspond to pure strategies (pure strategy NE can be seen as an extremal case of mixed-strategies).
In a *two-player game*, a strategy profile can be defined as a pair $(\mathbf{x}_i, \mathbf{x}_j)$ where $\mathbf{x}_i \in \Delta^m$ and $\mathbf{x}_j \in \Delta^m$. The expected payoff for this strategy profile is computed as:

$$\begin{aligned} u(\mathbf{x}_i, \mathbf{x}_j) &= \mathbf{x}_i^T A_{ij} \mathbf{x}_j \\ u(\mathbf{x}_j, \mathbf{x}_i) &= \mathbf{x}_j^T A_{ji} \mathbf{x}_i \end{aligned} \tag{2}$$

where A_{ij} (conversely A_{ji}) is the $m \times m$ payoff matrix of the game between player i and j. Each entry (h, k) of the payoff matrix A_{ij} corresponds to the gain received by player i when he plays strategy h against strategy k.

The strategy space of each player i is defined as a mixed strategy $\mathbf{x}_i$, as defined above. The payoff corresponding to the h-th pure strategy can be computed as:

$$u(x_i^h) = \sum_{j=1}^{n} (A_{ij} \mathbf{x}_j)^h \tag{3}$$

while the expected payoff of the entire mixed-strategy for player i is:

$$u(\mathbf{x}_i) = \sum_{j=1}^{n} \mathbf{x}_i^T A_{ij} \mathbf{x}_j \tag{4}$$

where n is the number of players with whom i plays and $A_{i,\_}$ is their payoff matrix of the game. Given these two functions is possible to find the NE of the game, and to do so we will use a result in the domain of Evolutionary Game Theory (EGT). The EGT, introduced by Maynard Smith and Price (1973), is a branch of game theory which aims to use the notions of GT to model the evolution of behavior in animal conflicts. In EGT we have a set of agents which play games repeatedly with their neighbors and update their beliefs on the state of the system choosing their strategy according to what has been effective and what has not in previous games. This loop is repeated until the system converges, which means that no player need to update its strategies because there is no way to do better.

To find those states, which correspond to the NE of the game, we use the *replicator dynamics* (Weibull, 1995):

$$x_i^h(t+1) = x_i^h(t) \frac{u(x_i^h)}{u(\mathbf{x}_i)} \; \forall h \in S \tag{5}$$

This equation allows better than average strategies to grow at each iteration and we can consider each iteration of the dynamics as an *inductive learning* process, in which the players learn

from the others how to play their best strategy in a determined context (see bottom part of Fig.1). The complexity of each step of the replicator dynamics (Eq.5) is quadratic but there are different dynamics that can be used with our framework to solve the problem more efficiently, such as the recently introduced *infection and immunization* dynamics (Rota Buló et al., 2011) that has a linear-time/space complexity per step and it is known to be much faster then, and as accurate as, the replicator dynamics.

3. Automatic Function Prediction Game

In this section the specific model for the AFP problem is explained in detail. We represent the proteins of an organism as players and their functions as strategies. The games are played between similar players, imposing only pairwise interactions. The payoff matrix is computed using a similarity function among GO terms and is weighted by the structural similarity between the proteins. The payoff function for each player is additively separable and is computed as described in Section 2.

Formulating the problem in this way we can apply equation (5) to compute the equilibrium state of the system, which corresponds to a consistent labeling of the data (Miller and Zucker, 1991). In fact, once stability is reached, all players play the strategy with the highest payoff. Each player arrives to this state not only considering its own strategies but also the strategies that other players are playing.

Our framework (see Fig. 1) require: *a)* the network that describe the interactions among the players, *b)* the similarity between the functions, *c)* the strategy space of the game and *d)* the payoff function.

3.1. Network of interactions:

The network of interactions models the interactions among the players and is represented as a weighted graph $G = (V, E, \omega)$ where the set of nodes $V = \{1, \ldots, n\}$ are the players/proteins and $E \subseteq V \times V$ the affinity between them weighted by the function ω. The edges E of G represents the affinity of the players, highest the value of an edge the more likely the two connected players will play together. The graph G is thus represented with an affinity matrix $\boldsymbol{W} = n \times n$, and its role is to encapsulate the similarities (structural, functional, etc.) between pairs of proteins motivated by the fact that similar or interacting proteins should share common functional annotations, such as the participation to the same biological process, the catalysis of similar biochemical reactions or the location inside the same cellular organelle. The crucial point here is having a good similarity measure $sim(\cdot, \cdot) \rightarrow \mathbb{R}_{\geq 0}$ that represent the closeness of pairs i and j:

$$w_{i,j} = sim_W(i, j) \; \forall i, j \in V \tag{6}$$

In our experiments the networks of interactions have been constructed combining together 8 different protein networks or directly using networks that natively combine different sources of data (Section 4.1).

On top of this network a neighbouring function $\mathcal{N}$ is applied for each player in order to sparsify the net and keeping only the more similar players for each one. The game-theoretic rationale that guided this choice is to select the subset of best matching co-players, while from a labeling perspective task this means to select the set of k neighbours of a point that weighs more in the labeling. Deciding the number of neighbours is often a tedious and stressful task which appears also in other methods, i.e. in k-NN classifier or in k-means clustering. To deal with this problem we decided to use two principles heuristics which are used in similar graph-theoretic methods. Given n the number of nodes in the protein graph, we propose these heuristics for the value k:

GC which stands for Graph Connectivity. The rationale is that by fixing $k = \lfloor log_2(n) + 1 \rfloor$ we guarantees that the underlying graph is statistically connected von Luxburg (2007). Being connected, from a game-theoretic perspective, means that all the players, directly or indirectly through a common neighbour, have the chances to influence the others choices.

k-NN with this heuristics we set $k = \lfloor \sqrt{n} \rfloor$. This rule of thumb is used in k-NN classifier to automatically tune the parameter k Duda et al. (2000). The rationale is that the graph-transduction game and the k-NN classifier are based on the same homophily principles where the labels are propagated from k labeled nodes to the unlabeled ones. If the heuristics holds for k-NN it should also for our method.

Given a value for k, found with the two methods above, the neighbours $\mathcal{N}_i$ of protein i is the set of $j \in \{1...n\}$ s.t. $w_{i,j} \geq \alpha_i$ where α_i is the weight of the k-th most similar element to i [2].

Building the neighbouring set in this way is obviously asymmetric. In order to make it symmetric we use the following policy: given two protein i, j if $j \in \mathcal{N}_i$ while $i \notin \mathcal{N}_j$ then $\mathcal{N}_j = \mathcal{N}_j \cup \{i\}$.

3.2. Function similarity graph

The function similarity graph models the similarity between pairs of GO terms from the used ontology. It is a weighted graph $G = (V, E, \omega)$ with self loop in which $\omega(i, j) \rightarrow \mathbb{R}_{\geq 0}$ weighs the similarity of the GO terms i and j. The graph G is represented as an $m \times m$ matrix $\mathbf{Z}$:

$$Z_{h,k} = sim_Z(h, k) \tag{7}$$

For the details of our implementation see Section 4.1.

3.3. Strategy space

The role of the strategy space $\mathbf{X}$ is to define all the possible associations between the n proteins and the m functions retrieved from an ontology. The space $\mathbf{X}$ is thus modelled as a $n \times m$ matrix in which each row corresponds to a mixed strategy $\mathbf{x}_i$ and each component x_i^h represents the strength of the association between the player (protein) i and the strategy (function) h. The strategy space $\mathbf{X}$ is the starting point of the game and can be initialized in different ways based on the fact that some

[2] $w_{i,:}$ is sorted in descendent order and α_i correspond to the value at position k

prior knowledge exists or not. Here we distinguish the initialization based on the type of protein, *labeled* or *unlabeled*. For the labeled proteins, since their functions are known, we use the following method:

$$x_i^h = \begin{cases} \frac{1}{f_i}, & \text{if } i \text{ has function } h. \\ 0, & \text{if protein } i \text{ does not have function } h. \end{cases} \quad (8)$$

where f_i is the number of terms associated with protein i.

For the testing proteins (the ones with no labels) we propose and evaluate two different initialization methods:

Without priors:. with this initialization all the GO terms have the same probability of being associated to a protein:

$$x_i^h = \frac{1}{m} \quad \forall h = \{1 \cdots m\} \quad (9)$$

With k-priors:. the rationale of this prior is to emphasize the labels assigned to the neighbouring set of a certain protein with the idea that similar protein should be assigned to similar classes. Given a protein i and its set of neighbouring proteins $\mathcal{N}_i$ (with labels), the prior is composed as follow:

$$x_i^h = \frac{1}{m} + \sum_{j \in \mathcal{N}_i} x_j^h \quad \forall h = \{1 \cdots m\} \quad (10)$$

and then $\mathbf{x}_i$ is normalized such that it add up to 1 $\left(x_i^h = \frac{x_i^h}{\sum_{h=1}^m x_i^h}\right)$ and remains in the m-dimensional simplex. The first term $(\frac{1}{m})$ gives the chances also to other functionalities to emerge. If it was set to 0 this possibility would have been lost and the method will focus only on the function that are assigned in the neighborhood.

3.4. Payoff Function

The payoff function has the role of assigning the gain that a certain player i receive when plays a strategy h (in graph-theoretic terms is the compatibility of assigning the function h to the protein i). The rationale is that we want to boost the association between similar players and similar GO terms. What we want for i, when plays with j, is that their labels are mutually affected, including the choice of i and j and also the set of similar labels to the ones associated to both the proteins. The set of similar functions is included with the idea that the correct labels could be received also from similar functions. This turns out to be:

$$u(x_i^h) = \sum_{j \in \mathcal{N}_i} \left(\left(w_{ij} \mathbf{Z} \right) \mathbf{x}_j \right)^h \quad (11)$$

and the expected payoff as,

$$u(\mathbf{x}_i) = \sum_{j \in \mathcal{N}_i} \mathbf{x}_i^T \left(w_{ij} \mathbf{Z} \right) \mathbf{x}_j \quad (12)$$

In this way we weight the influence that each protein receive from its neighbors. According to eq. 12, we assumed that the payoff of protein i depends on: w_{ij}, i.e. the similarity with its neighborhood proteins $j \in \mathcal{N}_i$; $\mathbf{Z}$, the similarities among the functional terms; $\mathbf{x}_j$, the preferences of neighborhood protein $j \in \mathcal{N}_i$ and the preferences $\mathbf{x}_i$ of the protein i itself. With $u(x_i^h)$ and $u(\mathbf{x}_i)$ we can start the dynamics of the game according to equation (5). During each phase of the dynamics, a process of selection allows strategies with higher payoff to emerge and at the end of the process each player chooses its functionalities according to these constraints, which make the labeling consistent (for an example see Fig.1).

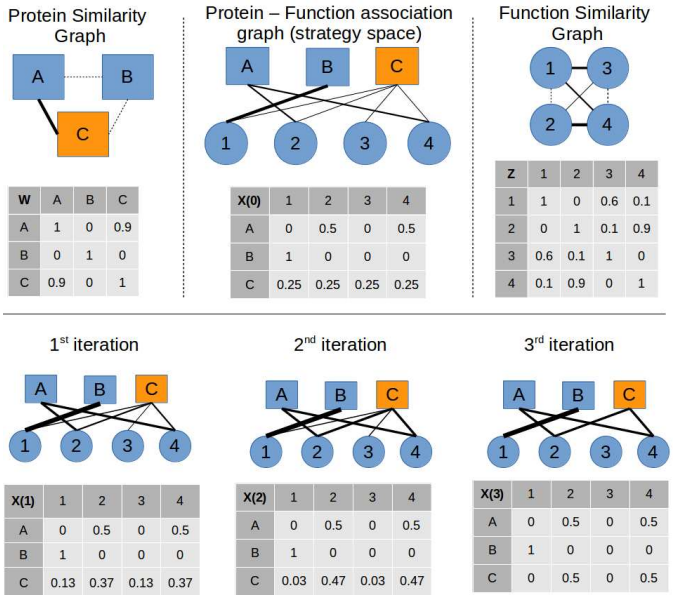

Fig. 1: The picture dissects the payoff function in order to understand what are the single components (three graphs on top) and what is happening to the assignment during the iteration of the dynamical system (eq 5). Consider the following situation: two similar proteins A and C ($A \in \mathcal{N}_C$) in which C has no prior on the functions (eq. 9) while A has the functions $2, 4$ assigned to it (eq. 8). In the first iteration is already possible to note how the labeling for C changes and becomes more similar to A.

4. Experiments

We applied different variants of our *graph transduction game* method (GTG) (see Section 4.4 for more details)

to the prediction of the Cellular Component (CC), Molecular Function (MF) and Biological Processes (BP) ontologies of the GO considering different model organisms, ranging from the human to the fruit fly and the zebrafish, involving thousands of functional classes (see Table 2).

4.1. Data

We constructed five networks representing the functional similarity between proteins. Two networks include phylogenetically related organisms: a) the *DanXen* network encompasses Danio rerio (zebrafish) and Xenopus laevis (a small austral frog); b) the *SacPomDic* network includes Saccharomyces cerevisiae, Schizosaccharomyces pombe and Dictyostelium discoideum (unicellular eukaryotes). The third network (*Dros*) is reserved to Drosophila melanogaster (fruit-fly), the model organism for insects.

Table 1: Data base and type of data used to construct the integrated protein similarity network for *DanXen*, *SacPomDic* and em Dros

Database	Type of data
PRINTS (Attwood et al., 2003)	Motif fingerprints
PROSITE (Hulo et al., 2006)	Protein domains and families
Pfam (Finn et al., 2006)	Protein domain
SMART(Letunic et al., 2006)	Simple Modular Architecture Research Tool (database annotations)
InterPro (Mulder et al., 2007)	Integrated resource of protein families, domains and functional sites
Protein Superfamilies(Gough et al., 2001)	Structural and functional annotations
EggNOG (Muller et al., 2010)	Evolutionary genealogy of genes: Non-supervised Orthologous Groups
Swissprot (Consortium, 2015)	Manually curated keywords describing the function of the proteins at different degrees of abstraction

Such networks are constructed by integrating 8 different sources of information from public databases (Table 1), as briefly described in the following.

At first, we obtained different profiles for each protein by associating for each source of data a binary feature vector, whose elements are 1 or 0 according to the protein annotation for a specific feature (e.g. whether or not a protein includes a specific domain, or a specific motif). Then the protein profiles have been used to construct a set of similarity networks (one for each data type) with edge scores based on the computation of the classical Jaccard similarity coefficient between each possible pair of protein profiles, thus obtaining 8 different protein networks. Finally the networks have been combined by unweighted mean integration (Valentini et al., 2014).

The remaining two networks contain proteins belonging to *Mus musculus* (*Mouse*) and *Homo sapiens* (*Human*) organisms, and have been retrieved from the STRING database, version 10.0 (Szklarczyk et al., 2015). The STRING networks are highly informative networks merging several sources of information about proteins, coming from databases collecting experimental data like BIND, DIP, GRID, HPRD, IntAct, MINT or from databases collecting curated data such as Biocarta, BioCyc, KEGG, and Reactome.

Each of these networks are then used in Sec.3.1 to define the interactions between the players (protein).

As class labels (groundtruth) for the proteins included in our networks we used the Gene Ontology CC, MF and BP experimental annotations extracted from the Swissprot database[3].

In order to enlarge the number of GO terms to be predicted, while preserving at the same the minimum information needed for the functional predictions, we removed only GO terms having less than two annotations, thus resulting in a number of classes ranging from 125 (CC ontology in *DanXen*) to 7309 (BP ontology in *Mouse* – Table 2).

The similarity between the GO terms for each integrated network and each ontology could be in principle computed using semantic similarity measures based e.g. on the the Resnick or Lin measures or other recently proposed variants (Caniza et al., 2014), but to show the applicability of our proposed method we adopted a simple Jaccard similarity measure between the annotations of each GO term. These similarities corresponds to the entries Z_{ij} in Eq.7.

[3] `http://www.expasy.org/` checked 19th May 2016

4.2. State-of-the-art methods compared with GTG

We compared GTG with several classical and state-of-the-art graph-based algorithms just applied to the the AFP problem: *Random Walk* (RW) and *Random Walk with Restart* (RWR), the *guilt-by-association* method (GBA), the *label propagation* algorithm (*LP*), three methods based on Hopfield nets, the *Gene Annotation using Integrated Networks* (GAIN), the *Cost-Sensitive Neural Network* (COSNet) and the *COSNet Multi-functionality-based ranking* (COSNetM), the *Multi-Source k-Nearest Neighbors* (MS-kNN), and the *RAnking of Nodes with Kernelized Score Functions* (RANKS). The compared algorithms are briefly described below.

RW A t-step random walk algorithm (Lovász, 1996) associates a protein $i \in V$ with a score corresponding to the probability that a random walk in G starting from positive nodes ends at node i after t random steps. The iterative procedure to update the probabilities uses at each step a transition matrix $\boldsymbol{T}$ obtained from $\boldsymbol{W}$ by row normalization, i.e. $\boldsymbol{T} = \boldsymbol{D}^{-1}\boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix $D_{ii}|_{i=1}^{n}$, with $D_{ii} = \sum_j w_{ij}$.

RWR After many steps the random walker in the RW algorithm may forget the prior information coded in the initial probability vector (0 for nodes in $V \setminus V_+$ and $1/|V_+|$ for nodes in V_+, where V_+ is the set of positive proteins for the current GO term). Thus, the RWR algorithm at each step allows the walker to move another random step with probability $1-\theta$, or to restart from its initial condition with probability θ.

GBA Family of algorithms relying upon the *guilt-by-association* principle, asserting that similar proteins are more likely to share similar functions (Schwikowski et al., 2000). Usually, the GBA discriminant score of a protein i for a given GO term is obtained as the maximum of the weights connecting i to neighboring proteins associated with that term (that is the positive proteins).

LP The label propagation algorithm, based on Gaussian kernels, iteratively propagates labels from labeled proteins to

Table 2: Number of proteins and GO terms with at least 2 annotations in each protein network.

Network	Proteins	CC terms	MF terms	BP terms
DanXen	6250	125	198	1502
SacPomDic	15836	858	1331	3934
Dros	3195	414	485	2985
Mouse	20648	701	1313	7309
Human	19247	860	1688	6298

the unlabeled ones until convergence (Zhu et al., 2003). During the label propagation the initial known labels are preserved.

GAIN An algorithm assigning labels to unlabeled proteins by minimizing the energy function of a Hopfield net (Hopfield, 1982) associated to the protein network (Karaoz et al., 2004). The net dynamics involves solely the unlabeled proteins, whose activation thresholds are set to 0, and whose initial state is set according to the labeling provided by the current GO term. The equilibrium point reached by the dynamics provides the binary labeling of unlabeled proteins. To provide even a ranking of proteins, in the present work the neuron energy at equilibrium is adopted as ranking score, following the approach presented in Frasca and Pavesi (2013).

COSNet Suitable for unbalanced data like the GO term annotations, this algorithm extends GAIN by substituting the classical Hopfield net with a parametric Hopfield net (Bertoni et al., 2011). The parameters, namely the neuron activation values and thresholds, are automatically learned in order to cope with the labeling imbalance (Frasca et al., 2013).

COSNetM An extension of COSNet exploiting the multifunctional properties of genes (Frasca, 2015).

MS-kNN One of the top-ranked methods in the recent CAFA2 international challenge. MS-kNN integrates several proteins sources/networks by applying the *k-Nearest Neighbours* algorithm (Altman, 1992) to each network independently, and then averages the obtained individual scores (Lan et al., 2013).

RANKS A ranking method adopting a suitable kernel matrix so as to extend the similarity between two proteins also to non neighboring proteins (Re et al., 2012). The score of each protein i for a given GO term is defined through a a local function that takes into account the neighborhood of each protein in the projected Hilbert space, according to the global topology of the underlying network.

For COSNet and RANKS we used the source code publicly available as **R** package (Frasca and Valentini, 2017; Valentini et al., 2016), and for the other methods we used the code provided by the authors or our in-house software implementations. The parameters required by our GTG approach and the other considered methods in this work have been learned through internal tuning on a small subset of training data.

4.3. Experimental setup

To evaluate the generalization performance of the compared methods we applied a 5-fold cross-validation experimental setting. According to the recent CAFA2 international challenge, to compare the results we considered both the "per class" Area Under the Precision Recall Curve (AUPRC), and the "per-example" multiple-label F-score. More precisely if we indicate as $TP_j(t)$, $TN_j(t)$ and $FP_j(t)$ respectively the number of true positives, true negatives and false positives for the protein j at threshold t, we can define the "per-example" multiple-label precision $Prec(t)$ and recall $Rec(t)$ at a given threshold t as:

$$Prec(t) = \frac{1}{n}\sum_{j=1}^{n}\frac{TP_j(t)}{TP_j(t)+FP_j(t)} \qquad Rec(t) = \frac{1}{n}\sum_{j=1}^{n}\frac{TP_j(t)}{TP_j(t)+FN_j(t)} \tag{13}$$

where n is the number of examples (proteins). In other words $Prec(t)$ (resp. $Rec(t)$) is the average multi-label precision (resp. recall) across the examples. The F-score multi-label depends on t and according to CAFA2 experimental setting, the maximum achievable F-score ($Fmax$) is adopted as the main multi-label "per-example" metric:

$$Fmax = max_t \frac{2Prec(t)Rec(t)}{Prec(t)+Rec(t)} \tag{14}$$

To have a fair comparison, the cross validation has been performed by adopting a non-stratified partition of proteins in folds unique for all methods. The AUPRC results have been averaged across folds having at least one annotated protein (otherwise the AUPRC by definition is meaningless).

4.4. GTG variants and settings

In our experiments we applied different variants of the GTG method, depending on the choice of the neighboring function (Section 3.1) and of the priors used to initialize the strategy space (Section 3.3) – see Table 3 for more details.

Table 3: Variants of GTG . The column *name* contains the name used for the particular setting in the paper; *neighbour size* refers to the sec. 3.1; *symmetric* if yes the neighbourhood is symmetrized; *prior* if yes the k-prior defined in sec. 3.3 to initialize the strategy space is used, otherwise no informative prior (uniform distribution) is used.

Name	Neighbour size	Symmetric	Prior
GTG α	GC	No	No
GTG β	GC	Yes	No
GTG γ	GC	Yes	Yes
GTG δ	k-NN	Yes	Yes

5. Results

We performed an extended experimental comparison between GTG α , GTG β , GTG γ and GTG δ methods and nine other state-of-the-art network-based algorithms using 5 different networks (*DanXen*, *SacPomDic*, *Dros*, *Mouse* and *Human*) labelled with terms of the three GO ontologies (BP, MF and CC). In this section we present and discuss the average results across classes (using the AUPRC metric) and across proteins (using the *Fmax* metric) for each network, considering separately the BP, MF and CC ontologies, thus resulting in 15 sets of average results involving thousands of functional classes and tens of thousands of proteins of different model organisms.

Multi-label *Fmax* results are summarized in Table 4. Independently of the model organism and the biological ontology considered, our proposed game theory-based transductive methods largely outperform the other methods (Table 4). In particular GTG γ and GTG δ achieve better results than the other methods (see the last two rows in Table 4). In several cases the relative improvement with respect to the best competing state-of-the-art method is close or larger than 50%: for instance with the MF ontology in *DanXen*, *Dros*, *SacPomDic* and *Mouse* networks, or with the BP ontology in *DanXen*, *Human* and *Mouse*. Also with the other ontologies and the other model organisms considered in this work the improvement with respect to the other network-based methods is impressive.

The only method that attains comparable results (but only limited to the CC ontology in *Human*) is the *MS-kNN* algorithm, one of the top ranked methods in the recent CAFA2 challenge for protein function prediction (Jiang et al., 2016) (Table 4). Note that GTG α and GTG β , which uses an uniform distribution to initialize the strategy space $\mathbf{X}$, usually obtain worse results that the other proposed variants GTG γ and GTG δ that adopt "neighborhood-aware" priors to initialize $\mathbf{X}$ (Section 3.3). Nevertheless, in most cases GTG α and GTG β too achieve comparable or significantly better results than all the other competing methods (Table 4).

Considering the AUPRC per-class metric, our proposed methods and in particular GTG α and GTG β achieve competitive results with respect to the other state-of-the-art network based algorithms, even if the results are not so compelling as with the per-example metric. Indeed average AUPRC results of GTG α better with respect to all the other competing methods (boldfaced in Table 5) are achieved in 11 out of the 15 pairs of network/ontology considered in this experimental comparison, while *GBA*, the second best method, is equal or better than all the other algorithms in 4 out of the 15 network/ontology pairs. Nevertheless we outline that our methods behave largely better with the *Fmax* per-example metric, since both GTG γ and GTG δ achieve better average results in 14 network/ontology pairs (Table 4).

This is not so surprising, since our graph-based transductive approach is conceived for a per-example multi-label learning: for each protein the labels (GO terms) are learned together in the same learning process taking into account the relationships between GO terms coded in the payoff function (eq. 12) used to compute the payoff u_i for each protein $i \in I$ (Section 2). Hence it is quite natural that our approach obtains better results with the hierarchical *Fmax* score, by which we take into account the multi-labels (i.e. the entire set of GO terms) correctly predicted for each protein, while reasonable but not so compelling results are obtained with the AUPRC metric computed on a per-class basis. Moreover, from a biological standpoint, in most cases biologists are more interested in the set of GO terms associated with a specific protein or a set proteins, than in the predictions for a specific term, since the functional and structural characteristics of a given protein are captured by the entire set of functions (GO terms) associated with the protein under study.

We note that for the per-class metric we did not report the classical AUROC (Area Under the Receiver Operating Characteristic curve), but the AUPRC instead. Indeed in the context of the protein function prediction, most of the GO terms are imbalanced, with a number of positive examples very low with respect to the total number of examples (proteins). In this imbalanced setting, from both a machine learning (Davis and Goadrich, 2006) and a bioinformatics standpoint (Saito and Rehmsmeier, 2015) it is well-known the AUPRC provides a more reliable metric to assess the overall performance of the prediction methods.

Summarizing, GTG α results in terms of AUPRC, and in particular GTG γ and GTG δ results in terms of the multi-label *Fmax* score, show that our game-theoretic-based approach can introduce significant improvements in network-based algorithms for AFP problems. The motivation of the success of the proposed approach is likely due to the fact that the game-theoretic model mimics, in a mathematical framework, the driving principle of the "guilt-by-association", and extends it by embedding in the learning process not only the similarities between proteins, but also the similarities between the functional terms of the GO. From a graph-learning standpoint this translates into a network-based semi-supervised approach by which the transductive process contextually learns all the labels (GO terms) associated with a specific protein, thus exploiting at the same time the relationships between both GO terms and proteins. Furthermore the experimental evidence suggests us the following *rule-of-thumb*: if one is interested in optimizing a per-example metric (like Fmax) prior knowledge should be added to the strategy space (see Sec.3.3) and the neighborhood should be symmetric 3.1. To optimize a per-class metric (like the AUPRC) using an uniform distribution in the strategy space and an asymmetric neighbouring system improve the results. In the first case this is explained by the fact that each testing sample is treated independently focusing more on the set of possible functions assigned to the neighbouring proteins. In the latter case we are interested in a (more) global metric, so assuming no prior knowledge for each sample let the protein-function assignment to naturally emerge from the data, thus capturing phenomena that span across the samples.

6. Conclusions

In this paper we have introduced a new game-theoretic perspective to the protein function prediction problem, which is motivated by the observation that network-based methods should take advantage not only of similarity information at the

Table 4: Fmax results across the terms of the CC, MF and BP ontology for *DanXen*, *Dros*, *SacPomDic*, *Human* and *Mouse* integrated protein networks. For each ontology and network the best results are highlighted in bold.

	Danxen			Dros			SacPomDis			Human			Mouse		
	CC	MF	BP	CC	MF	BP	CC	MF	BP	CC	MF	BP	CC	MF	BP
RANKS	0.5418	0.5075	0.4402	0.5483	0.3522	0.3201	0.6893	0.2951	0.4021	0.2804	0.1157	0.1467	0.2970	0.1354	0.1197
RWR	0.2588	0.2860	0.1156	0.1235	0.2237	0.0744	0.0718	0.1263	0.0662	0.0604	0.0374	0.0493	0.0545	0.0453	0.0367
COSNet	0.6055	0.4849	0.4542	0.5698	0.3811	0.2946	0.7128	0.4735	0.3364	0.1089	0.0847	0.0494	0.5694	0.3819	0.2292
COSNetM	0.6031	0.4831	0.4547	0.4405	0.3262	0.1820	0.5857	0.3953	0.2356	0.1953	0.1369	0.1572	0.4006	0.1958	0.1601
GAIN	0.3346	0.1796	0.2603	0.6215	0.1782	0.3642	0.7093	0.1054	0.1930	0.6015	0.5517	0.0828	0.5934	0.1118	0.0808
GBA	0.6572	0.5336	0.3314	0.5152	0.4532	0.2509	0.5002	0.5138	0.3746	0.3072	0.2365	0.1914	0.3285	0.2327	0.1544
LP	0.6678	0.5513	0.4328	0.6473	0.3687	0.4005	0.7244	0.2411	0.3006	0.6225	0.5361	0.263	0.6114	0.2535	0.2475
MS-kNN	0.3517	0.3574	0.2769	0.7120	0.5361	0.5138	0.8173	0.5386	0.5332	**0.6419**	0.5498	0.2276	0.6325	0.4055	0.2123
RW	0.2322	0.2767	0.0943	0.0962	0.1220	0.0562	0.0436	0.0573	0.0261	0.0481	0.0271	0.0374	0.0420	0.0335	0.0282
GTG α	0.6589	**0.5516**	0.3698	0.6315	**0.5762**	0.4037	0.7254	**0.6650**	0.4622	0.5856	**0.5916**	**0.3248**	0.5959	**0.5730**	**0.3108**
GTG β	0.6670	**0.5602**	0.3814	0.6403	**0.5966**	0.4119	0.7313	**0.6126**	0.4427	0.5852	**0.5966**	**0.3278**	0.5939	**0.5832**	**0.3127**
GTG γ	**0.8107**	**0.7188**	**0.6316**	**0.8283**	**0.7627**	**0.5881**	**0.8956**	**0.7953**	**0.6830**	0.6389	**0.6382**	**0.3902**	**0.6531**	**0.6301**	**0.3643**
GTG δ	**0.8138**	**0.7088**	**0.5973**	**0.8184**	**0.7489**	**0.5848**	**0.8989**	**0.7728**	**0.6694**	0.6397	**0.6346**	**0.3804**	**0.6568**	**0.6119**	**0.3521**

Table 5: Mean AUPRC results averaged across the terms of the CC, MF and BP ontology for *DanXen*, *Dros*, *SacPomDic Human* and *Mouse* integrated protein networks. For each ontology and network the best results are highlighted in bold.

	Danxen			Dros			SacPomDis			Human			Mouse		
	CC	MF	BP	CC	MF	BP	CC	MF	BP	CC	MF	BP	CC	MF	BP
RANKS	0.3014	0.266	0.1672	0.2972	0.3038	0.1879	0.2808	0.2183	0.1666	0.3061	0.0988	0.1109	0.2376	0.0933	0.0848
RWR	0.2318	0.2977	0.1399	0.1060	0.2400	0.0979	0.0920	0.2260	0.0880	0.219	0.0630	0.0650	0.157	0.0630	0.0530
COSNet	0.2556	0.2409	0.1469	0.2347	0.2389	0.1398	0.2526	0.1890	0.1240	0.1894	0.0319	0.0452	0.1726	0.072	0.0575
COSNetM	0.2473	0.2400	0.1475	0.2225	0.2363	0.1373	0.2558	0.1860	0.1220	0.1870	0.0317	0.0446	0.1713	0.0716	0.0577
GAIN	0.0216	0.0271	0.0099	0.0332	0.012	0.0145	0.0186	0.0017	0.0044	0.0199	0.0027	0.0022	0.0179	0.0017	0.0024
GBA	0.3213	0.4951	0.2203	0.2746	0.4577	0.1899	0.3074	0.5036	0.2115	0.3314	**0.1129**	**0.1293**	0.2573	**0.1161**	**0.1024**
LP	0.0308	0.0302	0.0187	0.0532	0.0279	0.0359	0.0256	0.0054	0.0112	0.2228	0.0692	0.065	0.1528	0.0563	0.0447
MS-kNN	0.1550	0.1297	0.0833	0.1475	0.1724	0.083	0.2009	0.1496	0.0987	0.1837	0.0109	0.0244	0.1337	0.0105	0.0136
RW	0.1998	0.3903	0.1347	0.0744	0.1476	0.0724	0.0312	0.0903	0.0318	0.1248	0.0402	0.0382	0.0938	0.0383	0.0336
GTG α	**0.4325**	**0.5462**	**0.2698**	**0.3904**	**0.5448**	**0.2379**	**0.5030**	**0.5735**	**0.3131**	**0.3805**	0.1025	0.1194	**0.2739**	0.1076	0.0884
GTG β	**0.4614**	**0.5565**	**0.2747**	**0.4151**	**0.5760**	**0.2448**	**0.5326**	0.5002	**0.2916**	0.3289	0.0933	0.1046	0.2311	0.1017	0.0733
GTG γ	0.3169	0.3534	**0.2357**	**0.2988**	**0.4626**	**0.2283**	**0.3632**	0.3933	**0.2684**	0.2593	0.0692	0.0761	0.1508	0.0632	0.0427
GTG δ	**0.3364**	0.4068	**0.2300**	**0.3213**	0.4554	**0.2349**	**0.4439**	0.4238	**0.2746**	0.2878	0.0721	0.0819	0.1855	0.0674	0.0490

level of proteins, as they usually do, but also of similarities between functional classes, which are available, e.g., in the Gene Ontology. Accordingly, we set up an abstract game whereby proteins (the players) have to choose a strategy (a functional class), in a non-cooperative manner, to get a payoff which is related to *both* protein-level and function-level similarities. It turns out that the Nash equilibria of this AFP game are related to a well-known notion of "consistency" in a contextual labeling problem (Hummel and Zucker, 1983; Miller and Zucker, 1991).

The results of extensive experiments confirm our original intuition that it does pay to incorporate functional-class similarities into network-based prediction algorithms, and demonstrate the power of simple game-theoretic dynamics to address this kind of problems.

References

Altman, N.S., 1992. An introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician 46, 175–185. doi:10.1080/00031305.1992.10475879.

Attwood, T.K., Bradley, P., Flower, D.R., Gaulton, A., Maudling, N., Mitchell, A., Moulton, G., Nordle, A., Paine, K., Taylor, P., et al., 2003. Prints and its automatic supplement, preprints. Nucleic acids research 31, 400–402.

Bertoni, A., Frasca, M., Valentini, G., 2011. COSNet: a cost sensitive neural network for semi-supervised learning in graphs, in: ECML, Springer. pp. 219–234. doi:10.1007/978-3-642-23780-5_24.

Caniza, H., Romero, A., Heron, S., Yang, H., Devoto, A., Frasca, M., Mesiti, M., Valentini, G., Paccanaro, A., 2014. GOssTo: a user-friendly stand-alone and web tool for calculating semantic similarities on the Gene Ontology. Bioinformatics 30. doi:{http://10.1093/bioinformatics/btu144}.

Cesa-Bianchi, N., Re, M., Valentini, G., 2012. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. Machine Learning 88, 209–241. URL: http://dx.doi.org/10.1007/s10994-011-5271-6.

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

Consortium, T.U., 2015. Uniprot: a hub for protein information. Nucleic Acids Research 43, D204–D212. URL: http://nar.oxfordjournals.org/content/43/D1/D204.abstract, doi:10.1093/nar/gku989, arXiv:http://nar.oxfordjournals.org/content/43/

Davis, J., Goadrich, M., 2006. The relationship between precision-recall and roc curves, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, New York, NY, USA. pp. 233–240. URL: http://doi.acm.org/10.1145/1143844.1143874, doi:10.1145/1143844.1143874.

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

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Classification (2Nd Edition). Wiley-Interscience.

Easley, D.A., Kleinberg, J.M., 2010. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press.

Erdem, A., Pelillo, M., 2012. Graph transduction as a noncooperative game. Neural Computation 24, 700–723.

Finn, R.D., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., et al., 2006. Pfam: clans, web tools and services. Nucleic acids research 34, D247–D251.

Frasca, M., 2015. Automated gene function prediction through gene multifunctionality in biological networks. Neurocomputing 162, 48 – 56. doi:http://dx.doi.org/10.1016/j.neucom.2015.04.007.

Frasca, M., Bertoni, A., Re, M., Valentini, G., 2013. A neural network algorithm for semi-supervised node label learning from unbalanced data. Neural Networks 43, 84–98. doi:10.1016/j.neunet.2013.01.021.

Frasca, M., Bertoni, A., Valentini, G., 2015. Unipred: Unbalance-aware network integration and prediction of protein functions. J. Comput. Biol. 22, 1057–1074. doi:doi:10.1089/cmb.2014.0110.

Frasca, M., Pavesi, G., 2013. A neural network based algorithm for gene expression prediction from chromatin structure., in: IJCNN, IEEE. pp. 1–8. doi:10.1109/IJCNN.2013.6706954.

Frasca, M., Valentini, G., 2017. Cosnet: An r package for label prediction in unbalanced biological networks. Neurocomputing 237, 397 – 400. doi:10.1016/j.neucom.2015.11.096.

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

Gene Ontology Consortium, 2013. Gene Ontology annotations and resources. Nucleic Acids Research 41, D530–535.

Gough, J., Karplus, K., Hughey, R., Chothia, C., 2001. Assignment of homology to genome sequences using a library of hidden markov models that represent all proteins of known structure. Journal of molecular biology 313, 903–919.

Hopfield, J., 1982. Neural networks and physical systems with emergent collective compatational abilities. Proc. Natl Acad. Sci. USA 79, 2554–2558.

Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., De Castro, E., Langendijk-Genevaux, P.S., Pagni, M., Sigrist, C.J., 2006. The prosite database. Nucleic acids research 34, D227–D230.

Hume, D., 2000. An enquiry concerning human understanding: A critical edition. volume 3. Oxford University Press.

Hummel, R.A., Zucker, S.W., 1983. On the foundations of relaxation labeling processes. Pattern Analysis and Machine Intelligence, IEEE Transactions on , 267–287.

Jiang, Y., et al., 2016. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biology 17. doi:10.1186/s13059-016-1037-6.

Joachims, T., 2003. Transductive learning via spectral graph partitioning, in: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 290–297.

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

Kleinberg, J., Tardos, E., 2002. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. Journal of the ACM (JACM) 49, 616–639.

Kohler, S., Bauer, S., Horn, D., Robinson, P., 2008. Walking the interactome for prioritization of candidate disease genes. Am. J. Human Genetics 82, 948–958.

Lan, L., Djuric, N., Guo, Y., S., V., 2013. MS-kNN: protein function prediction by integrating multiple data sources. BMC Bioinformatics 14.

Letunic, I., Copley, R.R., Pils, B., Pinkert, S., Schultz, J., Bork, P., 2006. Smart 5: domains in the context of genomes and networks. Nucleic acids research 34, D257–D260.

Lovász, L., 1996. Random walks on graphs: A survey, in: Miklós, D., Sós, V.T., Szőnyi, T. (Eds.), Combinatorics, Paul Erdős is Eighty. János Bolyai Mathematical Society, Budapest. volume 2, pp. 353–398.

von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and Computing 17, 395–416. URL: https://doi.org/10.1007/s11222-007-9033-z, doi:10.1007/s11222-007-9033-z.

Mayer, M., Hieter, P., 2000. Protein networks - guilt by association. Nature Biotechnology 18, 1242–1243.

Maynard Smith, J., Price, G.R., 1973. The logic of animal conflict. Nature 246, 15–18.

Miller, D.A., Zucker, S.W., 1991. Copositive-plus Lemke algorithm solves polymatrix games. Operations Research Letters 10, 285–290.

Mitrofanova, A., Pavlovic, V., Mishra, B., 2011. Prediction of protein functions with gene ontology and interspecies protein homology data. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8, 775–784.

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

Mulder, N.J., Apweiler, R., Attwood, T.K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Buillard, V., Cerutti, L., Copley, R., et al., 2007. New developments in the interpro database. Nucleic acids research 35, D224–D228.

Muller, J., Szklarczyk, D., Julien, P., Letunic, I., Roth, A., Kuhn, M., Powell, S., von Mering, C., Doerks, T., Jensen, L.J., et al., 2010. eggnog v2. 0: extending the evolutionary genealogy of genes with enhanced non-supervised orthologous groups, species and functional annotations. Nucleic acids research 38, D190–D195.

Nash, J., 1951. Non-cooperative games. Annals of mathematics , 286-295.

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

Radivojac, P., et al., 2013. A large-scale evaluation of computational protein function prediction. Nature Methods 10, 221–227.

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

Rota Buló, S., Pelillo, M., Bomze, I.M., 2011. Graph-based quadratic optimization: A fast evolutionary approach. Computer Vision and Image Understanding 115, 984–995.

Saito, T., Rehmsmeier, M., 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE 10, e0118432.

Schwikowski, B., Uetz, P., Fields, S., 2000. A network of protein-protein interactions in yeast. Nature biotechnology 18, 1257–1261.

Sharan, R., Ulitsky, I., Shamir, R., 2007. Network-based prediction of protein function. Mol. Sys. Biol. 8.

Szklarczyk, D., et al., 2015. String v10: proteinprotein interaction networks, integrated over the tree of life. Nucleic Acids Research 43, D447–D452. URL: http://nar.oxfordjournals.org/content/43/D1/D447.abstract, doi:10.1093/nar/gku1003, arXiv:http://nar.oxfordjournals.org/content/43

Tripodi, R., Pelillo, M., 2017. A game-theoretic approach to word sense disambiguation. Computational Linguistics .

Tripodi, R., Vascon, S., Pelillo, M., 2016. Context aware nonnegative matrix factorization clustering, in: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 1719–1724. doi:10.1109/ICPR.2016.7899884.

Valentini, G., Armano, G., Frasca, M., Lin, J., Mesiti, M., Re, M., 2016. RANKS: a flexible tool for node label ranking and classification in biological networks. Bioinformatics 32, 2872–2874. doi:{dx.doi.org/10.1093/bioinformatics/btw235}.

Valentini, G., Paccanaro, A., Caniza, H., Romero, A., 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, 63–78. doi:{http://10.1016/j.artmed.2014.03.003}.

Vapnik, V., 1998. Statistical learning theory. volume 1. Wiley New York.

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

Von Neumann, J., Morgenstern, O., 1944. Theory of Games and Economic Behavior. Princeton University Press.

Weibull, J., 1995. Evolutionary Game Theory. MIT Press.

Zhou, D., et al., 2004. Learning with local and global consistency, in: Adv. Neural Inf. Process. Syst.. volume 16, pp. 321–328.

Zhu, X., 2005. Semi-supervised learning with graphs. Ph.D. thesis. Carnegie Mellon University, language technologies institute, school of computer science.

Zhu, X., Ghahramani, Z., Lafferty, J.D., 2003. Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the 20th International conference on Machine learning (ICML-03), pp. 912–919.