Learning functional linkage networks with a cost-sensitive approach

Alberto BERTONI [a], Marco FRASCA [a,1]
Giuliano GROSSI [a] and Giorgio VALENTINI [a]
[a] *University of Milan, Computer Science Dept., Via Comelico 39/41, 20135 Milan, Italy*
{bertoni,frasca,grossi,valentini}@dsi.unimi.it

Abstract. Assigning functional classes to unknown genes or proteins on diverse large-scale data is a key task in biological systems, and it needs the integration of different data sources and the analysis of functional hierarchies. In this paper we present a method based on Hopfield neural networks which is a variant of a precedent semi-supervised approach that transfers protein functions from annotated to unannotated proteins. Unlike this approach, our method preserves the prior information and takes into account the imbalance between positive and negative examples. To obtain more reliable inferences, we use different evidence sources, and integrate them in a Functional Linkage Network (FLN). Preliminary results show the effectiveness of our approach.

1. Introduction

Functional linkage networks (FLNs) are well-established tools to represent the functional relationships between proteins. A FLN is a graph where each node corresponds to a protein, and an edge connects two proteins if any experimental or computational procedure states that these proteins might share a common biological characteristic. The main purpose of these networks is to find functional classes, i.e. Gene Ontology (GO) terms [1], for partially annotated or unannotated proteins.

A FLN can be constructed by various evidence sources, such as physical or genetic interactions, correlated gene expression data from microarray data, or correlated phylogenetic profiles. The single source is able to capture only a subset of the underlying biological characteristics of proteins, and the resulting FLNs have different reliability and coverage according to the selected source.

The connections between nodes in a single source graph often are "putative", i.e. some experiments suggest that two nodes can be related, but we have no assurance of this relationship. However, if a putative functional linkage between two proteins is established by independent experiments, than the confidence of a functional relationship between these proteins increases. As consequence, some researchers have proposed *integrated FLNs*, which are FLNs constructed combining several data sources, with the objective to obtain more robust and reliable functional linkages. Different approaches

[1]Corresponding Author: Marco Frasca, University of Milan, Computer Science Dept., Via Comelico 39/41, 20135 Milan, Italy; e-mail: frasca@dsi.unimi.it

have been proposed: Marcotte et al [2] describe a *conjunctive* integration, in which the integrated graph includes the edges confirmed in each single source graph; this approach is likely to generate a high false-negative rate. Analogously, the *disjunctive* integration tends to generate a high false-positive rate. Jiang et al [3] consider only the connections that have a "good" evidence in each single data source, and than they use a decision tree to establish if a connection between two protein must be maintained.

The selection of the integration method is crucial for the prediction phase, and it is dependent on the data sources and on the *decision rule* adopted. The decision determines the mechanism by which functional annotations are transferred to a node from its neighborhood. Several decision rules have been proposed in literature, such as the simple *guilt by association* rule [4], which transfers to a node the functional annotations of the neighbors with linkage weight greater than a fixed threshold. This rule does not consider the frequencies of the annotations in the neighborhood of nodes. The *neighborhood weighting rule* overtakes this limit considering either the frequencies of the annotations or the weights of the relevant connections [5]. Other rules infer functional annotations by extending the neighborhood of nodes at a distance greater than one edge [6].

Applying the decision rule to a node determines a functional consistency condition in its neighborhood, but it does not guarantee the functional consistency of the whole network. Some proposed methods attempt to achieve a *global* consistency by minimizing the number of locally inconsistent assignments [7,8,9]. Karaoz et al [9] propose a global FLN represented by an Hopfield network [10]: starting from a state (the prior knowledge), an equilibrium point that maximizes the global consistency is reached, and then the final state of the network is used to infer the functional annotations. This method allows to obtain inferences with a good reliability and a low computational effort. The algorithm has good performances when the "positive examples" rate for a given functional property is approximately equal to that of the negatives. When negative examples overcomes positives the prediction capabilities undergo a significant decay.

In this paper we propose two variants of Karaoz's approach in order to improve the performances in these cases: GAINu which "preserves" the prior information, and Cost-Sensitive GAIN (CS-GAIN) which takes into account the imbalance between positive and negative examples. In Sec. 2 we give some preliminary definitions necessary for the subsequent discussion; Sec. 3.1 contains the detailed description of the Karaoz's method, while in Sec. 3.2 and Sec. 3.3 the two variants of this method are introduced. Finally in Sec. 4 we present some preliminary results with the model organism *S. cerevisiae*.

2. Basic Definitions

The prior information about the protein network is represented by an undirected weighted graph $G(V, w)$, where $V = \{1, 2, \ldots, n\}$ is the set of nodes (proteins) and $w : V \times V \longrightarrow \mathbb{R}^+$ is a function which associates to each edge $\{i, j\}$ a non negative weight w_{ij}, with $w_{ij} = w_{ji}$ and $w_{ii} = 0$ for each $i, j \in V$,where w_{ij} is the evidence that two proteins share the same biological function. The weight is zero if there is no evidence that the two proteins share the same function.

To define functional annotations for nodes we use the Gene Ontology (GO) functional hierarchy [1]. GO covers three domains: *cellular component*, *molecular function* and *biological process*. Concepts are represented by terms and each term is used to de-

scribe gene products, through the mechanism of *annotations*. An annotation is an association between a term and a gene product suggested by suitable experimental or indirect (i.e. computational) methods. The hierarchical structure of GO is described by a directed acyclic graph, whose nodes are terms. The annotations are "preserved": if the term u is an ancestor of v, then the set of gene products annotated with v is a subset of the set of gene products annotated with u.

For each term c and a proteins i, we define:

$$ann_c(i) = \begin{cases} +1 & \text{if} \quad i \text{ is annotated with } c \\ -1 & \text{if} \quad i \text{ is not annotated with } c \end{cases}$$

Given the protein network $G(V, w)$, we write $V = V_a \sqcup V_u$, where

$$\begin{aligned} V_a &= \{i \mid \exists c \text{ s.t. } ann_c(i) = +1\} \\ V_u &= \{i \mid \forall c,\, ann_c(i) = -1\} \end{aligned}$$

are respectively the set of annotated and unannotated proteins in V.

3. Methods

In this section we firstly recall the dynamic FLN presented by Karaoz et al [9], then we introduce a method GAINu for preserving the prior knowledge, and a variant CS-GAINu that takes into account the different relevance of "positive" and "negative" examples in unbalanced GO classes.

3.1. GAIN.

The method proposed by Karaoz et al, called GAIN, consists, after the selection of a GO term c and of the original graph *G*, in defining a binary classifier based on a discrete asynchronous Hopfield network [10] depending on c and *G*. In this network each node *i* has a discrete state, denoted by x_i, which can be 1, 0 and -1. The state 0 corresponds to an uncertain state, and the purpose of this approach is to change these states in -1 or 1 during the network evolution. A final state +1 means that the protein associated to the node can be annotated with *c*. This method uses all annotated nodes as examples to train the network and to transfer functional annotations. The nodes *i* for which $ann_c(i) = 1$ are the positive examples, those for which $ann_c(i) = -1$ are the negative ones. The energy function of the network is

$$E(x) = -\frac{1}{2}\left[xWx^T - x\theta^T\right] \tag{1}$$

where W is the weight matrix, $\theta = (\theta_1, \ldots, \theta_n) \in \mathbb{R}^n$ is the vector of the activation thresholds, which is usually set to zero. At the beginning, the state of the network is:

$$x_i(0) = \begin{cases} ann_c(i) & \text{if } i \in V_a \\ 0 & \text{if } i \in V_u \end{cases}$$

The network update is asynchronous, i.e. only one node at a time is updated, and sequentially the update involves all network nodes.The activation rule of node i at time $t+1$ is given by:

$$x_i(t+1) = \mathrm{Sgn}\left(\sum_{j=1}^{i-1} w_{ij}x_j(t+1) + \sum_{j=i+1}^{n} w_{ij}x_j(t) - \theta_i\right)$$

where 'Sgn' is the *signum* function, and its argument computes the weighted sum of the states at time *t* of the neighbors of node *i*.

The main purpose is that, at the end of the execution of the algorithm, two nodes connected by an edge with a high weight have the same state; an edge that connects two nodes in the same state is called *consistent*. The goal is to maximize the weighted sum of the consistent edges. This can be done minimizing the energy function; in fact, as the weights w_{ij} are non negative, each consistent edge gives a positive contribution to the absolute value of *E* , while an inconsistent edge gives a negative contribution. Minimizing *E* maximizes the weighted sum of consistent edges, and Hopfield network with asynchronous dynamics is a local optimizer of the energy function.

3.2. GAINu.

A first observation about GAIN is that each iteration of the algorithm also updates nodes which are already annotated, and the network evolution can change the state of these nodes. In order to avoid to change the prior information, we consider a model where only unannotated nodes are updated. We called this variant GAINu, where *u* stands for "unannotated". Fixed a term c, suppose that, up to a permutation, $\{1, 2, \ldots, h\}$ are the unannotated nodes, $\{h+1, h+2, \ldots, n\}$ are those annotated, and $z = \{z_1, z_2, \ldots, z_h\}$, $y = \{y_{h+1}, y_{h+2}, \ldots, y_{h+l}\}$, with $l = n - h$, are their states respectively. Let $\theta^z = (\theta_1, \ldots, \theta_h)$ be the initial activation thresholds for unannotated nodes. Denoting with *W* the weight matrix, we have

$$W = \begin{pmatrix} W_u & W_{ua} \\ W_{ua}^t & W_a \end{pmatrix}$$

where W_u is the *h x h* matrix of the weights between each pair of unannotated nodes, W_a is the *l x l* matrix of the weights between each pair of annotated nodes, W_{ua} is the *h x l* matrix of the weights of edges between unannotated and annotated nodes and W_{ua}^t its transpose. By considering y fixed in the dynamics, the energy function is:

$$E(z) = -\frac{1}{2}\left(zW_u z^T - z\overline{\theta}^T\right) \tag{2}$$

where $\overline{\theta} = \theta^z - 2W_{ua}y^T$. At the beginning, the state of the network is 0; the activation rule for node i at time $t+1$ is

$$z_i(t+1) = \mathrm{Sgn}\left(\sum_{j=1}^{i-1} w_{ij}z_j(t+1) + \sum_{j=i+1}^{h} w_{ij}z_j(t) - \overline{\theta}_i\right)$$

with $\overline{\theta}_i = \theta_i - 2\sum_{j=h+1}^{n} w_{ij}y_j$.

Updating at each iteration only unannotated nodes makes GAINu faster than GAIN, and this is an important advantage when using large scale data.

In summary, both GAIN and GAINu use some positive and negative examples to propagate their information to the whole network, but they do not take into account the different proportions of positive and negative examples.

3.3. CS-GAINu

A common situation in the GO ontology is that positive examples are very few, sometimes with a ratio positive/negatives less than 1/10. This situation affects considerably methods based on neural networks, because the network evolution in this case tends to the stable point $x^* = \{-1\}^n$. This behavior may be more or less evident depending on the network topology, and on the presence of different connected components in the network. To prevent this behavior we propose a model that modifies GAINu, called CS-GAINu, in which the influence of positive examples is increased. This approach is also motivated by the fact that usually positive examples derive from an accurate study, while negative examples, except for rare cases, are simply proteins not annotated for the term under investigation. For increasing the influence of positive nodes, we propose first of all to consider states $\{\gamma, -1\}$, with $\gamma > 1$. Then we return to states $\{1, -1\}$ through an affine transformation which preserves the dynamics:

$$z_i \in \{\gamma, -1\} \longrightarrow v_i \in \{1, -1\}, \qquad \forall i \in \{1, 2, \ldots, h\}$$

with $z_i = av_i + b$, and $a = \frac{\gamma+1}{2}$, $b = \frac{\gamma-1}{2}$. The energy function of this model is

$$E(v) = -\frac{1}{2}\left[vW_u v^T - v\widehat{\theta}^T\right] \tag{3}$$

where $\widehat{\theta} = (\widehat{\theta}_1, \ldots, \widehat{\theta}_h)$, $\widehat{\theta}_i = \frac{1}{a}\big(\theta_i - 2\Delta_i - (\gamma - 1)D_i\big)$, $\Delta_i = \sum_{j=h+1}^{n} w_{ij}y_j$, $D_i = \sum_{j=1}^{h} w_{ij}$ and h is the number of unannotated nodes. The update rule for node i at time $t + 1$ is

$$v_i(t+1) = \mathrm{Sgn}\left(\sum_{j=1}^{i-1} w_{ij}v_j(t+1) + \sum_{j=i+1}^{h} w_{ij}v_j(t) - \widehat{\theta}_i\right).$$

When $\gamma = 1$ we have $\widehat{\theta}_i = \overline{\theta}_i$, which are exactly the activation thresholds in GAINu. Δ_i contains the information relative to annotated nodes in the neighborhood of node i, and it reflects the initial imbalance between positive and negative examples. A value of γ that reduces the absolute value of the thresholds $\widehat{\theta}_i$ allows to balance the Δ_i contribution. Moreover, γ value must allow the network to avoid the trivial states $\{1\}^h$ and $\{-1\}^h$. Therefore, we select the value of γ which minimizes

$$f(\gamma) = ||\widehat{\theta}||^2 = \sum_{i=1}^{h} \left(\frac{2}{\gamma+1}\right)^2 (-2\Delta_i - (\gamma-1)D_i)^2$$

where the initial thresholds are set to 0. The value of gamma which minimizes $f(\gamma)$ is

$$\gamma = 1 + 2\frac{M_{\Delta\Delta} - M_{D\Delta}}{M_{DD} - M_{D\Delta}}$$

where $M_{\Delta\Delta} = \sum_{i=1}^{h} {\Delta_i}^2$, $M_{D\Delta} = \sum_{i=1}^{h} D_i\Delta_i$ and $M_{DD} = \sum_{i=1}^{h} {D_i}^2$. If negative examples overwhelm those positive, $M_{D\Delta}$ is negative and so γ is greater than 1. In the case in which the number of positives is larger than negatives we use the precedent approach by setting γ = 1.

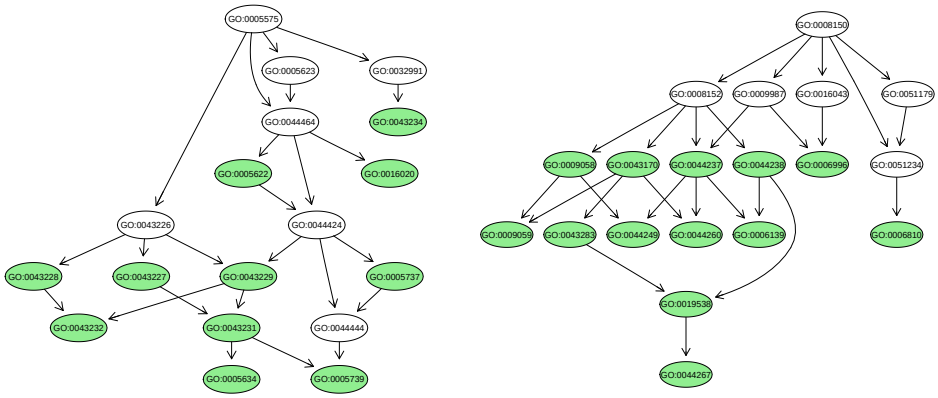
4. Experimental Results

4.1. Data

The experimental context refers to a public dataset used by Chua et al in [11] and available on-line at *http://srs2.bic.nus.edu.sg/∼kenny/integration*. The dataset is made up by the *Saccharomyces Cerevisiae* yeast proteins from SGD with at least one GO annotation. Protein association information are obtained from six different data sources: *sequence homology* (BLAST), *protein-protein interactions* (BIOGRID), *Pfam domains* (SwissPfam), *pubmed abstract* (NCBI Entrez Pubmed), *Predicted interactions* (STRING) and *Gene expression data* (Eisen , Rosetta Compendium). Expression data weights represent the absolute Pearson's correlation with 0.7 threshold. As first naive approach, we have integrated this sources in two steps:

- from the four non expression data sources, we construct an integrated graph *G* so that an edge (u, v) is in G if (u, v) is at least in one of the four source graphs;
- in *G* we maintain edges connecting pairs of proteins having at least one of the two expression data sources, and their weights are represented by the correlation of the corresponding expression values.

We obtain a final graph of 1081 yeast proteins. In [11] to annotate proteins are considered only the GO *informative* terms, which are terms with at least 30 annotations and with children having less than 30 annotations. As our dataset contains only 1081 yeast proteins, there are informative terms with no annotations for these proteins, and we discarded them from our experiments. Moreover we need terms which give us the opportunity to test our methods in different conditions of positive/negative ratio, in order to verify the effectiveness in overcoming the initial imbalance of positive and negative examples. Therefore we consider the GO terms in *Cellular Component* and *Biological Process* ontologies which have at least 800 yeast annotations, excluding the root and its children which are too generic. We obtain 11 and 13 functional terms respectively in CC and BP ontologies, as shown in Table 1. Either CC terms or BP ones result in very close locations in the corresponding ontology DAG, as we can see in Figure 1. In addition to structural relationships, such terms also have also semantic relationships, in the sense that they share part of their annotations.



(a) (b)

Figure 1. The two subgraphs representing (gray nodes) the selected GO Cellular Component (a) and Biological Process (b) terms.

Table 1. Functional terms of GO Cellular Component and Biological Process ontologies used as prediction terms.

	Term	*Description*	*Ontology*
1)	GO:0043232	intracellular non-membrane-bounded organelle	CC
2)	GO:0043228	non-membrane-bounded organelle	CC
3)	GO:0005739	mitochondrion	CC
4)	GO:0016020	membrane	CC
5)	GO:0043234	protein complex	CC
6)	GO:0005634	nucleus	CC
7)	GO:0005737	cytoplasm	CC
8)	GO:0043231	intracellular membrane-bounded organelle	CC
9)	GO:0043227	membrane-bounded organelle	CC
10)	GO:0043229	intracellular organelle	CC
11)	GO:0005622	intracellular	CC
12)	GO:0009059	macromolecule biosynthetic process	BP
13)	GO:0006810	transport	BP
14)	GO:0006996	organelle organization	BP
15)	GO:0044249	cellular biosynthetic process	BP
16)	GO:0009058	biosynthetic process	BP
17)	GO:0044267	cellular protein metabolic process	BP
18)	GO:0044260	cellular macromolecule metabolic process	BP
19)	GO:0019538	protein metabolic process	BP
20)	GO:0006139	nucleobase, nucleoside, nucleotide and nucleic acid metabolic process	BP
21)	GO:0043283	macromolecule metabolic process	BP
22)	GO:0043170	macromolecule metabolic process	BP
23)	GO:0044238	primary metabolic process	BP
24)	GO:0044237	cellular metabolic process	BP

4.2. Results

We have tested GAIN, GAINu and CS-GAINu performing 10-folds cross validation on the 1081 yeast proteins and validating each method using the BP and CC terms previously selected. Proteins are randomly divided into 10 equal-sized subsets. Each time the annotations for proteins in a fold are hidden (their initial state set to 0) and predicted using as training data the annotations for proteins in the other nine folds.

As the sequence of updated nodes in the network is randomly defined, the convergence trajectories of the network are not deterministic; therefore to have more reliable results we iterated 10-folds cross validation ten times for both GAINu and CS-GAINu, and five times for GAIN (to reduce the major computational burden of the Karaoz's algorithm), averaging the results. Prediction capabilities are evaluated for all methods by using the F-score, that is the harmonic mean of precision p and recall r, with $p = TP/(TP + FP)$ and $r = TP/(TP + FN)$. TP is the number of positive examples correctly predicted, FP is the number of negative examples wrongly predicted as positive, and FN is the number of positive examples predicted as negatives. Table 2 summarizes the results achieved with the three methods. For terms with a great

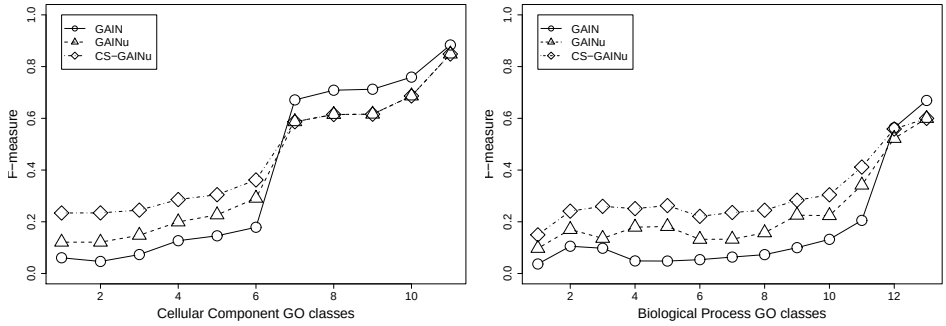
Table 2. Performance comparison between GAIN, GAINu and CS-GAINu.

GOTerms	*Positives*	*F-score*		
		GAIN	*GAINu*	*CS-GAINu*
1) GO:0043232	163	0.061	0.121	0.234
2) GO:0043228	163	0.047	0.121	0.234
3) GO:0005739	202	0.073	0.148	0.245
4) GO:0016020	232	0.127	0.199	0.286
5) GO:0043234	253	0.146	0.227	0.305
6) GO:0005634	334	0.179	0.291	0.362
7) GO:0005737	636	0.671	0.587	0.587
8) GO:0043231	654	0.709	0.615	0.615
9) GO:0043227	654	0.712	0.616	0.616
10) GO:0043229	717	0.759	0.686	0.686
11) GO:0005622	884	0.884	0.848	0.848
12) GO:0009059	88	0.037	0.097	0.149
13) GO:0006810	213	0.105	0.170	0.241
14) GO:0006996	201	0.098	0.136	0.260
15) GO:0044249	176	0.049	0.179	0.251
16) GO:0009058	203	0.048	0.182	0.263
17) GO:0044267	181	0.054	0.132	0.221
18) GO:0044260	197	0.064	0.133	0.236
19) GO:0019538	208	0.073	0.157	0.245
20) GO:0006139	248	0.100	0.225	0.283
21) GO:0043283	267	0.132	0.223	0.304
22) GO:0043170	365	0.205	0.342	0.412
23) GO:0044238	533	0.563	0.522	0.559
24) GO:0044237	605	0.669	0.600	0.600

imbalance towards negatives, which represents the majority of GO terms, GAIN has aw-

ful performances, GAINu performs better than GAIN and CS-GAINu shows a better discriminative capability than the other two approaches. Even when the initial imbalance towards negatives is small CS-GAINu has better performances than GAINu, as for example for GO term *GO:0044238*. When the number of positives is larger than negatives (a quite rare case in the GO), GAIN obtains slightly higher F-scores due to the major possibility for GAIN to propagate evidence across the network. Figure 2 summarizes this results.

A further advantage of GAINu and CS-GAINu is their low computational time, which in our experiments is ten times lower than the time spent by GAIN.



(a) (b)

Figure 2. Comparison between GAIN, GAINu and CS-GAINu on Cellular Component (a) and on Biological Process (b) terms.

5. Conclusions.

In this paper we present a new method for protein function prediction based on Hopfield neural networks, called CS-GAINu, and some preliminary results about its application to the functional classification of proteins in the model organism *S.Cerevisiae*. This method is a variant of GAIN, a semi-supervised approach able to transfer protein functions from annotated to unannotated proteins. Unlike GAIN, CS-GAINu takes into account the imbalance between positive and negative examples to prevent that a high imbalance adversely affects the functional predictions.

Preliminary results provide a first confirmation of the effectiveness of this approach to manage the unbalance between positive and negative annotations that characterize the Gene Ontology, and encourage us to test this method with other datasets using larger sets of functional terms.

Moreover we are experimenting a new approach based on CS-GAINu that takes into account the existing hierarchical and semantic relationships between functional terms to perform hierarchical multi-label predictions at genome-wide level.

References

References

[1] The Gene Ontology Consortium (2000): Gene Ontology: tool for the unification of biology. *Nature Genetics* , 2000, **25** 25–29, doi:10.1038/75556. PMID 10802651

[2] Marcotte E. M., Pellegrini M., Thompson M.J., Yeates T.O., Eisemberg D.: A combined algorithm for genome-wide prediction of protein function. *Nature*, 1999, **402** 83–86

[3] Jiang T., Keating A.E.: AVID: an integrative framework for discovering functional relationships among proteins. *Bioinformatics*, 2005, **6** 136

[4] Oliver S.: Guilt-by-association goes global. *Nature*, 2000, **403** 601–603

[5] McDermott J., Bumgarner R., Samudrala R. : Functional annotation from predicted protein interaction networks. *Bioinformatics*, 2005, **21(15)** 3217–3226

[6] Hishigaki H., Nakai K., Ono T., Tanigami A., Takagi T.: Assessment of prediction accuracy of protein function from protein - protein interaction data. *Yeast*, 2001, **18** 523–531

[7] A. Vazquez, A. Flammini, A. Maritan, A Vespignani: Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, 2003, **21**, 697 - 700

[8] T.M. Murali, C. J. Wu, S. Kasif: The art of gene function prediction. *Nature Biotechnology*, 2006, **24** 1474 - 1475

[9] Karaoz U., Murali T.M., Letovsky S., Yu Zheng, Chumming Ding, Cantor R.C. : Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc Natl Acad Sci USA*, 2004, **101** 2888–2893

[10] J. J. Hopfield: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 1982, vol. 79 no. 8, pp. 2554-2558

[11] H. N. Chua, W. K. Sung and L. Wong: An efficient strategy for extensive integration of diverse biological data for protein function prediction. *Bioinformatics*, 2007, **23**:3364-3373

[12] Lee I., Date S.V., Marcotte E.M:: A Probabilistic Functional Network of Yeast Genes. *Science*, 2004, **306(5701)**: 1555-1558