Labeling network motifs in protein interactomes for protein function prediction

Jin Chen Wynne Hsu Mong Li Lee
School of Computing
National University of Singapore
Singapore 119077
{chenjin, whsu, leeml}@comp.nus.edu.sg

See-Kiong Ng
Institute for Inforcomm Research
21 Heng Mui Keng Terrace
Singapore 119613
sknq@i2r.a-star.edu.sg

Abstract

Biological networks such as the protein-protein interaction (PPI) network have been found to contain small recurring subnetworks in significantly higher frequencies than in random networks. Such network motifs are useful for uncovering structural design principles of complex biological networks. However, current network motif finding algorithms models the PPI network as a uni-labeled graph, discovering only unlabeled and thus relatively uninformative network motifs as a result.

Our objective is to exploit the currently available biological information that are associated with the vertices (the proteins) to capture not only the topological shapes of the motifs, but also the biological context in which they occurred in the PPI networks for network motif applications. We present a method called LaMoFinder to label network motifs with Gene Ontology terms in a PPI network. We also show how the resulting labeled network motifs can be used to predict unknown protein functions. Experimental results showed that the labeled network motifs extracted are biologically meaningful and can achieve better performance than existing PPI topology based methods for predicting unknown protein functions.

1 Introduction

Motifs in a network are small connected subnetworks that are found to be repeatedly occurring in the network in frequencies that are significantly higher than in random networks. Many complex networks in the real world, such as the gene regulatory network and the protein-protein interaction network, have recently been found to contain such topological patterns of local connections [16]. Analysis of network motifs in these naturally occurring networks has led to many interesting results. For example, it was shown that conserved network motifs allow protein-protein interaction predictions [2], and that they can be used to discover the underlying network decomposition [9]. As such, network motifs have been gaining increasing attention as a useful concept to uncover structural design principles of complex networks [15, 16, 19].

Current approaches in finding network motifs typically consist of two major subtasks:

- **Task 1.** Find which classes of isomorphic subgraphs occur frequently in the input network;
- **Task 2.** Verify which of these subgraph classes are also displayed at a much higher frequency than in random graphs.

The first subtask discovers network motifs that are *frequent* or *repeated* in the network, while the second subtask ensures that they are also *unique*. Clearly, network motif discovery is a computationally challenging problem, but scientists have begun to devise methods for detecting motifs in large networks. For example, the MFINDER by Kashtan *et. al* [10] supported the detection of network motifs consisting of up to eight vertices, while the latest NeMoFinder by Chen *et. al* [5] has enabled the discovery of network motifs with sizes ranging all the way to meso-scale, since many of the relevant processes in biological networks have been shown to correspond to the meso-scale (5-25 genes or proteins) [18].

However, the current PPI network motif finding methods are based on a standard graphical model of protein-protein interactions (PPI) as *uni-labeled networks*. In this model, a species' "interactome" is defined as a network of interactions between the n proteins found in the species (*i.e.* its "proteome"), represented as a graph in which all the vertices (*i.e.* proteins) are *uniquely labeled* with $v_1, \ldots, v_n$. As a result, the network motifs generated by the current motif finding algorithms are "unlabeled", capturing only the topological shapes of the motifs, and not the biological context in which they occurred. While these network motifs have been shown to be somewhat competent for certain biological applications such as protein interaction prediction [2],

1-4244-0803-2/07/$20.00 ©2007 IEEE.

such purely statistical patterns are not informative enough for the more sophisticated biological applications of network motifs that have been envisaged by researchers; for example, in protein function prediction using a dictionary of network motifs and their functional information to predict the functions of unknown proteins [3].

Since the current uni-label model treats each protein in a PPI network as a unique and anonymous entity, it inadvertently ignores any other useful biological information that we may have already known about some of the proteins. In reality, the biologists usually would have performed experimental studies on some of the proteins to determine their biological functional roles and the cellular sublocalization. In fact, there are ongoing systematic efforts to annotate the various proteins in a species' proteome with the known biological information using the Gene Ontology or GO (Section 2). This means that the underlying PPI network is actually a partially labeled network, with many of the vertices (i.e. proteins) being already annotated with known functional and cellular sublocalization labels. In order to exploit the availability of such useful biological information associated with the proteins in network motif applications, we introduce a third subtask to the problem of network motif mining:

- **Task 3.** Assign biological labels to the vertices in the network motifs such that the resulting labeled subgraphs also occur frequently in the underlying labeled input network.

The task of labeling the network motifs (formally defined in Section 3) turns out to be computationally expensive, due to the sophisticated GO scheme by which the proteins are annotated. There is often missing information even in the most well-studied model organism. As a result, not all the proteins in the PPI network are annotated with biological information. When they are, many of the proteins would be multiply-labeled since they have complex biological roles. Moreover, the biological labeling scheme is hierarchical, introducing a further element of complexity. As such, even if both the motif size and the number of the motifs are small, it is almost impossible to hand-label the motifs. In fact, the number of possible motifs' labels increases exponentially as we graduate to meso-scale network motifs.

In this paper, we propose an algorithm, LaMoFinder, which stands for *La*beled *Mo*tif *Finder*, to label the network motifs discovered in a biological network (Section 3). Such enrichment of the network motifs enables them to become biologically meaningful for the more sophisticated biological applications such as protein function prediction envisaged by researchers. We apply LaMoFinder to label network motifs mined from the large whole-genome *S. cerevisiae* (Yeast) PPI network for knowledge discovery applications. Our evaluation results show that our labeled network motifs are biologically meaningful (Section 4) and can achieve better performance than existing topology-based methods for predicting unknown protein functions using PPI (Section 5).

2 Gene Ontology

The Gene Ontology (GO) project [1] is a collaborative effort initiated since 1998 to construct and use ontologies to facilitate the systematic annotation of genes and their products (e.g. proteins) in a wide variety of organisms. The resulting GO ontologies have now been accepted as the *de facto* language for the description of attributes of genes and gene products, with a rapidly growing number of model organism databases and genome annotation groups contribute annotation sets using GO terms to GO public repository.

The GO ontologies provide a systematic language for the description of attributes of biological entities in 3 key domains that are shared by all organisms, namely molecular function, biological process and cellular component. In each of these domains, the corresponding GO ontology is structured as a directed acyclic graph (DAG) to reflect the complex hierarchy of biological terminologies. Mathematically, suppose $T = \{t_1, t_2, \ldots, t_n\}$ is a set of GO terms, we say term t_i is a direct child of term t_j, if and only if t_i is an instance ("is-a" relationship) or a component ("part-of" relationship) of t_j ($t_i, t_j \in T$).

To properly model the biological information in different genomes, we also need to take into account that not all the GO terms are equally informative within a certain genome due to their biological differences [12]. In other words, for each genome, we assign genome-specific weights to the GO terms based on the method suggested by Lord et. al [12]: the weight of a GO term is defined as the ratio of the number of occurrences of the GO term and any of its descendants' terms in the genome to the total number of terms occurrences in the genome. We denote it as $w(t)$, $\forall t \in T$. By this definition, the root node has a weight of 1.

Figure 1 shows an illustrative example of a subset of GO. In addition, Table 1 shows its protein annotation list. We observe that G04 is a child of G02 following the "is-a" relationship. G06 is a child of G03 following the "part-of" relationship. In addition, the weight of G04 is 0.42 because 245 out of 585 proteins are annotated with G04 or its decedents. Note that it is possible for a child term to have multiple parents in GO. In Figure 1, G05 has G02 and G03 as its parents.

Zhou et al [21] define a GO term as an informative functional class (FC) if the GO term has at least 30 proteins directly annotated with it. In Figure 1, G04, G05, G06, G09, and G10 are informative FC. In this work, we are interested in a subset of the informative FC, namely the informative FC with no ancestors that are informative. We call them

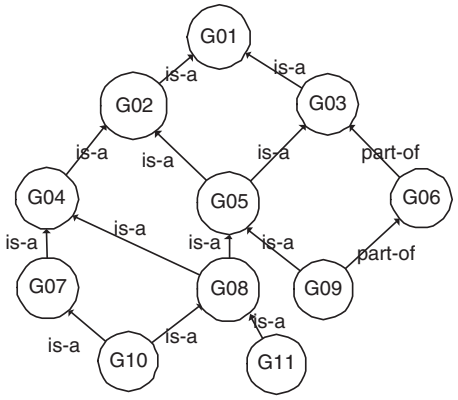

Figure 1. An illustrative example of the complicities of GO.

GO term t	Num. of proteins annotated with t	Num of proteins annotated with t and its decedents	GO term weight $w(t)$
G01	0	585	1.00
G02	0	415	0.71
G03	20	475	0.81
G04	100	245	0.42
G05	70	280	0.48
G06	150	250	0.43
G07	10	100	0.17
G08	25	135	0.23
G09	100	100	0.17
G10	90	90	0.15
G11	20	20	0.03
SUM	585		

Table 1. Example: Weights and the numbers of occurrences of GO terms in Figure 1 .

the *border* informative FC. Border informative FC are used to avoid the generation of labels that would be too general. In our example, G09 and G10 have informative ancestors G05. Hence they are not excluded from the border informative FC.

Having introduced the background of GO annotations, we now illustrate some of the difficulties in labeling network motifs with GO annotations. Figure 2 shows an unlabelled network motif g that has been discovered in a PPI network. The occurrences of g in the PPI network G are shown in Figure 3 and protein GO annotations are shown in Table 2. The task is to label the vertices of g such that the labeling scheme is consistent with some occurrences of g. In other words, the labels must be the same, or more general than the annotation of the corresponding vertex in the occurrence.

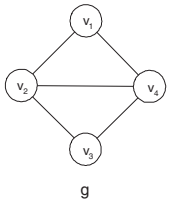

Figure 2. Example: network motif g.

For example, suppose we label the vertices $\{v_1, v_2, v_3, v_4\}$ as $\{G04, G08, G04, G05\}$ in Figure 3. For occurrence o_1, suppose vertices $\{v_1, v_2, v_3, v_4\}$ are mapped to $\{p_1, p_2, p_3, p_4\}$. We observe that G04 is one of the annotation of p_1 (see Table 2). For p_2, although G08 is not in any of the p_2's annotation, we realize that G10 is in fact a descendant of G08. In other words, assigning

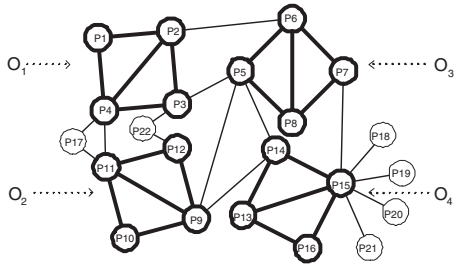

Figure 3. Example: 4 occurrences (shown with thick lines) of the network motif g (Figure2) in a PPI network G.

G08 to v_2 is appropriate since it is more general than the annotation of p_2 (G10). Similarly, p_3's annotation of G08 is a descendant of G04 and p_4's annotation of G09 is a descendant of G05. We can conclude that the labeling scheme $\{G04, G08, G04, G05\}$ is consistent with the occurrence o_1.

From this example, we realize that the task of labeling network motifs from biological networks needs to consider the following issues:

1. **Multiple and hierarchical labeling.**

   Biologically, many proteins are involved in multiple cellular processes and they are therefore labeled with more than one GO term, e.g., the proteins in yeast are currently annotated with an average of 9.34 GO terms. Therefore, the number of labeling schemes that are consistent with an occurrence increases exponentially with network motif size. This leads to the scalability issue.

Protein	GO annotation	Protein	GO annotation
p_1	G04, G09, G10	p_9	G11, G10
p_2	G10, GO3	p_{10}	G03, G05, G07
p_3	G08	p_{11}	G05
p_4	G09, G07	p_{12}	G09
p_5	G03	p_{13}	G11
p_6	G10	p_{14}	G04, G05
p_7	G03	p_{15}	G04
p_8	G05	p_{16}	G04, G09

Table 2. Example: GO annotations for proteins in occurrences o_1, o_2, o_3 and o_4.

2. **Symmetric vertices.**

   Symmetric vertices are vertices that can be interchanged without affecting the topological structure of the network. For example, the network motif g in Figure 2 has two sets of symmetric vertices, $\{v_1, v_3\}$ and $\{v_2, v_4\}$. The existence of these sets of symmetric vertices implies that we need to enumerate all possible mappings between the motif vertices and the occurrence vertices in order to obtain all the possible labeling schemes. Time complexity increases exponentially with the size of the symmetry set. Furthermore, testing whether a graph has any axial symmetry is an NP-complete problem [13]. This also increases the complexity of the labeling work.

The LaMoFinder method to be described in Section 3 is specifically devised to address the above challenges effectively.

3 LaMoFinder

We model a biological network as a graph $G = (V, E)$ where each vertex in V represents a biological entity (e.g., a protein for PPI networks, or a gene for gene regulatory networks), and each edge in E between two vertices v_A and v_B indicates that there exists a biological relation detected between the corresponding proteins/genes A and B. To simplify discussion, we will focus on PPI networks, although our algorithm can be applied to any biological networks.

A network motif g is a frequently occurring non-random subgraph pattern in a network G [16]. By definition, g is a connected, unlabeled subgraph that is repeated and unique in G. For each g, there exists a set of occurrences of this network motif in G, denoted as D_g.

Problem Definition. Let $T = \{t_1, t_2, \ldots, t_n\}$ be the set of GO terms which will be assigned to the vertices of network motifs as labels. Each GO term in T is a border informative FC or a descendant of a border informative FC. A labeling scheme L of g is said to conform to an occurrence o ($o \in D_g$) if the assigned labels for all vertices of g are either the same or more general than the label of the corresponding vertices in o. Our goal is to find all possible labeling schemes for the vertices of a network motif g such that they conform to at least σ occurrences in D_g.

A naive approach is to pick an occurrence at random and use its labels as a possible labeling scheme. It then proceeds to determine the number of occurrences that conform to this labeling scheme. If the number of occurrences is less than σ, it picks a combination of vertices at random and generalizes their labels one level up the function hierarchy. With the generalized vertex labels, the total number of occurrences that conforms to the labeling scheme is recomputed. If the number exceeds σ, the scheme is output. The process is repeated till all occurrences have participated in at least one labeling scheme. Clearly, this approach is not scalable. As the network motif size increases, the number of possible vertices combination to generalize increases exponentially. A better approach is needed.

We design a heuristic network motif labeling algorithm called LaMoFinder. Instead of enumerating all possible vertices and their sets of possible generalized labels, we start with the set of occurrences and try to group the occurrences based on their degree of similarity to each other. As the occurrences are grouped, we determine the least general labeling scheme that conforms to all the occurrences in the group. Here, the least general labeling scheme refers to selecting the lowest GO terms that is able to encompass all the occurrences.

In Figure 4, suppose we group o_1 and o_2 and assume that $\{p_1, p_2, p_3, p_4\}$ are matched with $\{p_{12}, p_9, p_{10}, p_{11}\}$. The corresponding annotations for $\{p_1, p_2, p_3, p_4\}$ are $\{(G04, G09, G10), (G10, G03), (G08), (G09, G07)\}$; while the corresponding annotations for $\{p_{12}, p_9, p_{10}, p_{11}\}$ are $\{(G09), (G11, G10), (G03, G05, G07), (G05)\}$. Then the least general labeling scheme is $\{(G05, G09), (G08, G10), (G04, G05), (G05)\}$.

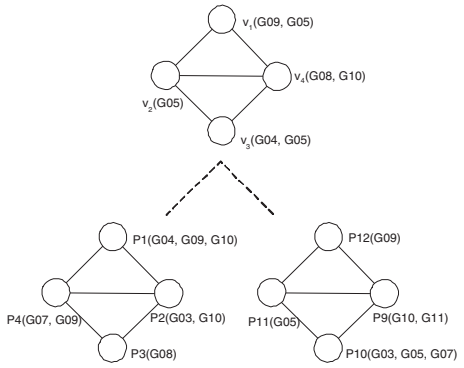

Figure 4. Example: The labeling of two occurrences

Two issues immediately surface. The first issue concerns the computation of similarity measures between occurrences. To address this problem, we derive a similarity measure for occurrences based on the GO term similarities. The second issue concerns the grouping criteria. This is dealt with in subsection 3.2.

3.1 Similarity Measure for Occurrences

As we use GO terms as labels, we first compute the similarity value between any two GO terms. Based on the GO term similarity, we will compute the similarity value between occurrences.

GO Term Similarity. Given two GO terms t_a and t_b and their corresponding weights $w(t_a)$ and $w(t_b)$, we adopt an enriched GO term comparison method [11] to assign a term similarity score for t_a and t_b, denoted as $ST_{(t_a,t_b)}$.

Recall that GO allows multiple parents for each term. Two terms may share one or more common parents via different paths. For example, in Figure 1, G08 and G09 have 2 common parents (G05 and G01). We denote the GO term of the lowest common parent (in our example, this corresponds to G05) as t_{ab}. Then the similarity between GO terms t_a and t_b is defined as:

$$ST(t_a, t_b) = \frac{2 \times \ln w(t_{ab})}{\ln w(t_a) + \ln w(t_b)} \quad (1)$$

where $w(t_x)$ is the weight of GO term t_x in T. As $1 \geq w(t_{ab}) \geq w(t_a)$ and $1 \geq w(t_{ab}) \geq w(t_b)$, $ST(t_a, t_b)$ varies between 1 and 0.

Occurrence Similarity. The similarity between any two occurrences o_i and o_j of a network motif g is determined from the similarities between the corresponding vertices of o_i and o_j. The computation of the occurrence similarity has two complications.

The first complication arises from the fact that each vertex of an occurrence may have multiple labels. For any two vertices v_i and v_j, let T_{v_i} and T_{v_j} be the set of GO terms annotated to v_i and v_j respectively, we define the similarity score $SV_{i,j}$ for vertices v_i and v_j as:

$$SV(v_i, v_j) = 1 - \prod_{t_a \in T_{v_i}, t_b \in T_{v_j}} (1 - ST(t_a, t_b)) \quad (2)$$

where $ST(t_a, t_b)$ denotes the similarity between GO term t_a and t_b computed with Equation 1. Note that $SV(v_i, v_j)$ is close to 1 as long as there is at least one good GO term match among the lists of GO terms in T_{v_i} and T_{v_j}. In other words, two vertices are considered similar if they share at least one biological feature.

The second complication arises due to the presence of two or more symmetric vertices. In our example, occurrence o_1 has symmetric vertices $\{p_1, p_3\}$ and $\{p_2, p_4\}$ and occurrence o_2 has symmetric vertices $\{p_{12}, p_{10}\}$ and $\{p_9, p_{11}\}$. Let $I_1 = \{v_{11}, v_{12}, \ldots, v_{1t}\}$ be one set of symmetry vertices in o_1 and $I_2 = \{v_{21}, v_{22}, \ldots, v_{2t}\}$ be the corresponding set of symmetry vertices in o_2. We denote $pair(I_1, I_2)$ as the possible pairings of the vertices between the two sets I_1 and I_2. In our example, $pair(\{p_1, p_3\}, \{p_{12}, p_{10}\}) = \{\{(p_1, p_{12}), (p_3, p_{10})\}, \{(p_1, p_{10}), (p_3, p_{12})\}\}$.

Let $\wp_a = \{I_{a1}, I_{a2}, \cdots, I_{ak}\}$ be the set of all sets of symmetric vertices in the occurrence o_i; $\wp_b = \{I_{b1}, I_{b2}, \cdots, I_{bk}\}$ be the set of all sets of symmetric vertices in the occurrence o_j. We define the similarity score of the occurrences o_i and o_j, $SO(o_i, o_j)$, as:

$$SO(o_i, o_j) = \frac{1}{|V|} \sum_{a,b=1}^{k} \left(max\{ \sum_{pair(I_a, I_b)} SV(v_\alpha, v_\beta)\} \right) \quad (3)$$

where $|V|$ is the number of vertices in the network motif, and $(v_\alpha, v_\beta) \in pair(I_a, I_b)$, $I_a \in \wp_i$ and $I_b \in \wp_j$.

For example, if we want to compute the pairwise similarity scores for the occurrences o_1 and o_2, we need to find the sets of symmetric vertices. This problem has been proven to be NP-complete by J. Manning in [13]. Several heuristics are known to be polynomial in general. Here, we make use of the heuristics provided in the graph algorithm library PIGALE (*http://pigale.sourceforge.net/*). Table 3 shows the occurrence similarity between o_1 and o_2.

occurrence o_1	occurrence o_2	SV score
p_1(G04, G09,G10)	p_{12}(G09)	1.00
p_1(G04, G09,G10)	p_{10}(G03, G05,G07)	0.99
p_2(G03, G10)	p_9(G10, G11)	1.00
p_2(G03, G10)	p_{11}(G05)	0.76
p_3(G08)	p_{10}(G03, G05,G07)	0.80
p_3(G08)	p_{12}(G09)	0.45
p_4(G07, G09)	p_{11}(G05)	0.69
p_4(G07, G09)	p_9(G10, G11)	0.99
SO score		0.87

Table 3. Example: Similarity score between occurrences o_1 and o_2

3.2 Grouping Occurrences

Having worked out the details to compute the similarity of occurrences, the next issue concerns the grouping of the occurrences such that we can find all the possible labeling schemes that encompass the σ number of occurrences.

One possible solution is to use the popular clustering algorithm such as the k-means clustering algorithm to find clusters of size σ. For each cluster, we derive the labeling scheme by assigning to the vertex of the network motif one GO term that conforms to all the occurrences of that vertex.

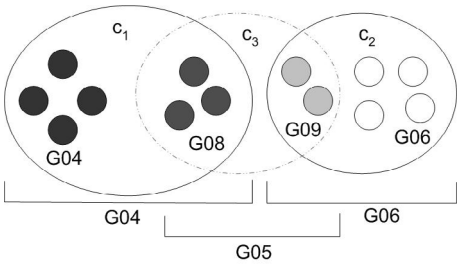

Figure 5. Example: Clusters and their labeling schemes. Each node in the figure represents an occurrence.

Unfortunately, this approach does not work well due to the hierarchical structure of the GO ontology. Consider Figure 5. We observe that if we use k-means clustering, all the occurrences will be grouped into non-overlapping clusters. We can find 2 labeling schemes c_1 and c_2 with threshold $\sigma = 5$. However, a closer examination shows that there are in fact 3 possible labeling schemes. This example shows that non-overlapping clusters may miss some valid and significant labeling schemes.

In order to discover all the possible labeling schemes for the unlabeled network motifs, we adopt an agglomerative hierarchical clustering method to cluster the occurrences based on the occurrence similarity measures in Section 3.1.

In the hierarchical clustering process, each occurrence is initially a cluster by itself. At each iteration, pairs of the most similar clusters are joined to form a new cluster. The least general labelling scheme of the cluster is derived. If a cluster does not have any occurrence to combine with, it proceeds to the next step. The clustering process stops when the labeling scheme has assigned more than half of the vertices with labels that belong to the border informative FC. If the number of occurrences within the cluster exceeds σ, the cluster's labels are saved as a labeling scheme.

If a protein in an occurrence of a motif in the interactome does not have a GO annotation, the generalized label of this node in the motif will be determined by the GO annotations of corresponding proteins in the other occurrences. If none of the corresponding proteins has a GO annotation, we label the node with label "unknown".

The details of LaMoFinder are given in Algorithm 1 and Algorithm 2. LaMoFinder continuously combines the clusters of occurrences until all the labeled network motifs are obtained. In the worst case, LaMoFinder takes $O(|D|^2)$ computational time in the pairwise similarity computation, where D is the size of the occurrence set of network motif g. The unavoidable NP problem, graph symmetry, could be solved with an existing heuristic method that has $O(n^3)$ time complexity, where n is the number of the vertices of g.

o_1	o_2	common label
G04, G09, G10	G09	G02, G09, G05
G03, G10	G10, G11	G03, G10, G08
G08	G03, G05, G07	G03, G05, G04
G07, G09	G05	G02, G05

Table 4. Example: The minimum common father labels of vertices in occurrence o_1 and o_2

Algorithm 1 LaMoFinder

Input: G - PPI network;
T - the set of GO terms;
g - a network motif of G;
D - occurrence set of g in G;
σ - Frequency threshold;
Output: L - Labeled network motif set;
$L \leftarrow \emptyset$;
$C \leftarrow D$;
$C' \leftarrow \emptyset$;
$\Upsilon \leftarrow getSymmetry(g)$;
while $|C| \neq 1$ and $C \neq C'$ **do**
  $C' \leftarrow C$
  for each cluster $c_i,\ c_j\ \in\ C$ **do**
    $Sim \leftarrow getSimilarity(c_i, c_j, \Upsilon)$;
  end for
  $C \leftarrow Cluster(C', Sim, \Upsilon)$;
end while
for each cluster $c \in C$ **do**
  if $size(c) \geq \sigma$ **then**
    $L \leftarrow c$;
  end if
end for
return L;

4 Experiment Results

We implemented LaMoFinder in C++ and carried out experiments on a 3.0GHz single processor Pentium PC with 1GB memory. For evaluation, we applied LaMoFinder on an experimentally-derived (yeast-two-hybrid) interaction data for *Saccharomyces cerevisiae* (yeast) downloaded from the BIND database. The interactome comprises of 7903 Y2H interactions between 4401 of the yeast proteins. After removing redundant links and self-links, the resulting

Algorithm 2 $Cluster(C', Sim, \Upsilon)$

Input: C' - set of clusters of occurrences of g;
Sim - set of pairwise similarity scores of clusters in C;
Υ - Symmetry vertices set in g;
Output: C - the new set of the clusters;
$C \leftarrow \emptyset$;
for each $c_i \in C'$ **do**
  if less than half of vertices in c_i are border informative FC **then**
    $c_i' \leftarrow c_i$'s closest cluster in C'
    $C \leftarrow Combine(c_i, c_i', \Upsilon)$;
  end if
end for
return C;

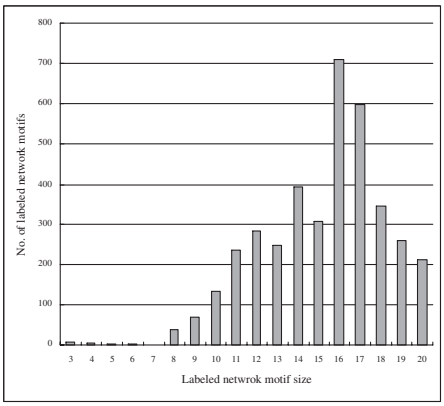

Figure 6. Labeled network motif distribution.

PPI network has 7095 edges connecting 4141 vertices.

We utilized the NeMoFinder algorithm in [5] to discover 1367 network motifs from the PPI network. Motifs of sizes up to 20 were discovered by NeMoFinder. All the motifs have frequencies of at least 100 times in the PPI network, with a uniqueness value of more than 0.95 (against random networks).

The GO annotations for the yeast proteins were downloaded from the Gene Ontology database [1]. 3554 out of the 4141 yeast proteins are found to have at least one GO biological annotation. There are 3 different branches of GO annotations (function, process and location). We call LaMoFinder 3 times to label the network motifs based on the 3 branches of GO annotations before using them for protein function prediction (Section 5).

4.1 Meso-scale labeled network motifs

We set the labeled network motif frequency threshold to 10, requiring each labeled network motif to have at least 10 occurrences in the PPI network.

Out of the 1367 unlabeled network motifs, LaMoFinder is able to extract a total of 3842 labeled network motifs from the PPI network. Figure 6 shows that the number of labeled network motifs varies with motif size. We observe that the majority of the labeled network motifs are meso-scale. For example, 18.5% labeled network motifs have 16 vertices, and 15.6% labeled network motifs have 17 vertices. This is in accordance to the observation that many relevant processes in biological networks are at the meso-scale (5-25 genes or proteins) level [18].

4.2 Biologically meaningful motifs

We asked a biologist to peruse the different classes of labeled network motifs to verify if there are any motifs discovered by LaMoFinder that would be biologically meaningful.

We first check whether LaMoFinder is able to discover biologically meaningful uni-labeled network motifs, since scientists have observed a notable functional homogeneity in large motifs [20]. Figure 7 shows a uni-labeled motif g_1 discovered by LaMoFinder that is indeed verified to be commonly found in protein splicing complexes. Here we use size-5 network motifs to simplify discussion. The nodes represent actual proteins in the occurrences, and edges represent detected physical interactions between the proteins. Each labeled network motif has at least 10 occurrences in the interactome of Yeast.

Next, we verify whether LaMoFinder is able to discover non-uni-labeled motifs where the vertices have different but biologically related labels. For example, in Figure 7, the network motif g_2 is labeled with 3 different function labels. Our biologist has ascertained that g_2 is indeed a biologically meaningful motif because it depicts a interesting biological possibility that a protein with function *"carbohydrate utilization"* can be regulated (via *"mRNA transcription"*) by its indirect neighbor with function *"regulation of carbohydrate utilization"*.

Finally, we test the biological validity of the network motifs that are labeled with multiple types of GO terms, since we have labeled our motifs with both functional labels as well as cellular localization labels in this work. In fact, just like we have shown in the above non-uni-labeled example, the resulting complex network motifs can reveal interesting biological insights. For example, the third labeled motif g_3 shown in Figure 7 illustrates how a parallel-labeled motif can reveal from the PPI network such insightful information as how proteins with different functions may operate in different cellular localizations. The upper triangle of g_3 shows a protein triplet labeled with the same function, sug-

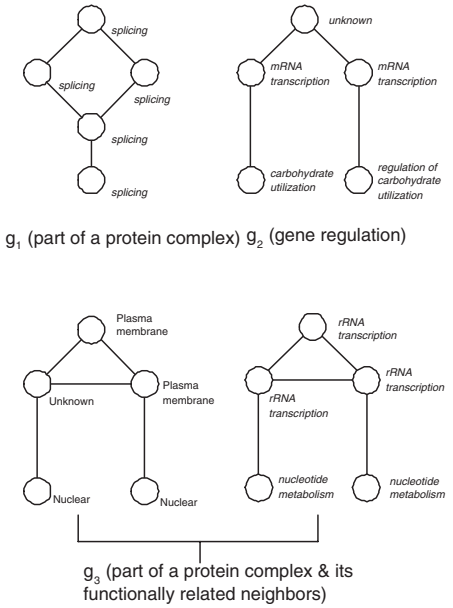

Figure 7. Example labeled network motifs. Italic labels are functional annotations, and leading capital labels refer to cellular location annotations.

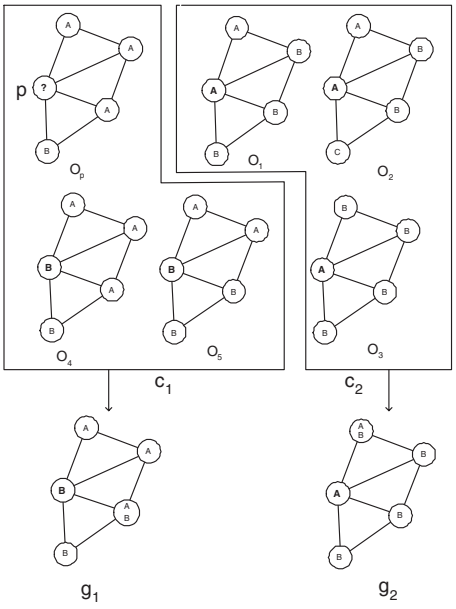

Figure 8. Example: predicting function of protein p from labeled motif g_1.

gesting that they are likely to form a protein complex for the purpose of, in this case, rRNA transcription. The other two vertices in the motif depict its functional neighbors that are necessary for this biological process to occur. On closer examination at the parallel cellular sublocalization labels of this motif, we can postulate the various locations in which this complex biological process typically take part.

The above findings illustrate that using LaMoFinder to label network motifs can reveal interesting insights to help biologists better understand the underlying biological processes.

5 Application: Protein Function Prediction

Determining protein functions experimentally is an expensive process. As such, even in yeast, the historically most well-studied model organism, only about 60% of yeast proteins have been functionally annotated to-date. Scientists have recently envisaged the accurate prediction of protein functions using a dictionary of network motifs and their functional information [3]. In this section, we describe how this can be achieved with network motifs that have been functionally labeled by LaMoFinder.

5.1 Prediction with Labeled Motifs

Suppose we have a labeled network motif $g_{labeled}$ and its set of occurrences O in a PPI network G. We observe that

1. Any protein p in an occurrence $o_i \in O$ is topologically similar to its corresponding proteins in the occurrences $O - \{o_i\}$; and
2. All the proteins in o_i other than p are functionally similar to their corresponding proteins in the occurrences in $O - \{o_i\}$.

Therefore, we propose to predict unknown protein functions by using labeled network motifs as follows:

Given a protein p whose function is unknown, and p is located in an occurrence of a labeled network motif $g_{labeled}$, we can predict the functions of p by using the functions of proteins that are topologically similar to p in the occurrences of $g_{labeled}$.

For example, Figure 8 shows an unknown protein p in occurrence o_p. The occurrence o_p is in the cluster of occurrences c_1 which has the labeled motif g_1. We can actually predict that protein p has the function B from the corresponding vertex in the labeled motif g_1.

A straightforward method to predict protein functions using network motifs is to build a dictionary of network motifs and their functions, as suggested in [3]. However,

a network motif is likely to have multiple functions, as we have seen in the many non-uni-labeled motifs discovered by LaMoFinder. In order to measure the relation between network motif and protein function more precisely, we define the concept of labeled network motif strength (LMS).

Let g be a network motif, $g_{labeled}$ be a labeled network motif of g. Let $D_{g_{labeled}} = \{o_1, \ldots, o_m\}$ be the set of occurrences of $g_{labeled}$. We say that $g_{labeled}$ is the labeled network motif for a protein p if and only if p is a vertex in o_i ($o_i \in D_{g_{labeled}}$ and $1 \leq i \leq m$).

We can rank the labeled network motifs in terms of their contribution to the PPI network with respect to their individual frequencies and uniqueness. For a labeled network motif $g_{labeled}$, the frequency value is the number of occurrences in G that conforms to $g_{labeled}$. The uniqueness of $g_{labeled}$ is the number of times g's frequency is equal or greater than its frequency in randomized networks, over the total number of randomized networks [16]. For simplicity, we assume that the labeled network motifs are independent of each other. For a labeled network motif $g_{labeled}$, we define the labeled network motif strength $LMS(g_{labeled})$ as:

$$LMS(g_{labeled}) = \frac{s(g_{labeled}) \times |g_{labeled}|}{max_k} \quad (4)$$

where $|g_{labeled}|$ is the frequency of $g_{labeled}$; $s(g_{labeled})$ is the uniqueness value of $g_{labeled}$; max_k is the maximal value of $s(g_{labeled}) \times |g_{labeled}|$ of all size-k labeled network motifs.

Given a set of labeled network motifs for protein p, denoted as LG_p, let v be the corresponding vertex of p in a labeled network motif $g_{labeled}$ ($g_{labeled} \in LG_p$), and $x_1, \ldots, x_k$ be the k functions of v. Then the likelihood that protein p has function x is given by:

$$f_x(p) = \frac{1}{z} \sum_{g_{labeled} \in LG_p} (\delta^{g_{labeled}}(v, x) \times LMS(g_{labeled})) \quad (5)$$

where $\delta^{g_{labeled}}(v, x)$ returns the frequency of function x on vertex v in $g_{labeled}$. $\delta^{g_{labeled}}(v, x)$ is 0 if x is not a function of v. z is a normalization parameter to ensure that $f_x(p)$ is between 0 and 1.

5.2 Results

Previous works have shown that simple topological methods [6, 17] could outperform sequenced-based methods, especially in the case of functional similarity without sequence homology. Hence, we expect that using topologically similar proteins will further improve the precision of function prediction. We compare our method with some of the well-known topological associative analysis methods that have been recently shown to be useful in the inference of unknown protein function:

1. The neighbor counting (NC) approach [17] labels a protein with the function that occurs frequently in its neighbors. The k most frequent functions are assigned as the k most likely functions for that protein.

2. Chi-Square (Chi^2) approach is a statistical approach proposed by Hishigaki et al [8] that makes use of Chi-Square statistics to take into account the frequency of each function in the dataset.

3. PRODISTIN [4] uses the Czekanowski-Dice distance between each pair of proteins as a distance metric and clusters the proteins using the BIONJ algorithm.

4. The MRF approach proposed by Deng *et. al* [7] is a global optimization method based on Markov Random Fields and belief propagation to compute a probability that a protein has a function given the functions of all other proteins in the interaction dataset.

All the above prediction methods are based on the functional information of nearby proteins in the network. The proposed use of meso-scale labeled network motifs will enable, for the first time, the exploitation of remote but topologically similar proteins for the functional prediction of unknown proteins.

To facilitate comparison, we use the same PPI dataset employed by the other methods. The PPI dataset was download from MIPS [14] and it comprises 1877 proteins and 2448 physical interactions after removing 120 pairs of self-interactions. We apply NeMoFinder followed by LaMoFinder to discover a set of labeled network motifs for this MIPS dataset. Then, we use a leave-one-out strategy to recognize top 13 functional categories [1] of yeast proteins. Figure 9 shows the precision and recall of the various methods. The proposed labeled network motif prediction method shows improved accuracy.

6 Conclusion

Many biological networks such as the PPI network have been found to contain small recurring subnetworks in significantly higher frequencies than in random networks [16]. Scientists have believed that such overabundant topological modules in the network can be useful for uncovering the structural design principles of complex biological networks. However, current network motif finding algorithms invariably models the PPI network as a uni-labeled graph, limiting themselves to only discovering unlabeled (and uninformative) network motifs. As a result, the currently available biological information that are associated with the vertices

[1] Based on the hierarchical structure of GO, for precision and recall computation, we generalized all function annotations to the top 13 key functions in Yeast.

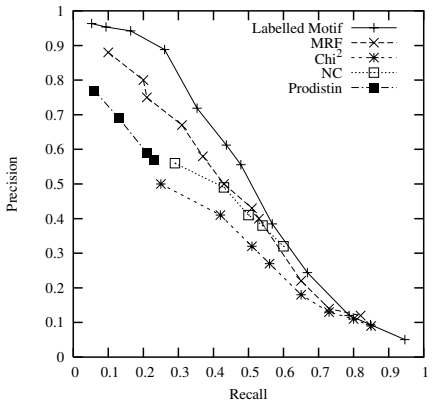

Figure 9. Precision vs. Recall for labeled network motif functional prediction

(the proteins) cannot be exploited for further knowledge discovery applications.

In this work, we have proposed a method called LaMoFinder to annotate network motifs with the biological information associated with the proteins in the PPI network. Our method was specifically devised to handle the large labeling space as well as the sophisticated scheme (GO) in which the proteins were annotated. As a result, we have captured not only the topological shapes of the motifs, but also the biological context in which they occurred in the labeled network motifs.

We also demonstrated how the network motifs labeled by LaMoFinder can be used to predict the functions of unknown proteins in the PPI network. Our superior performance against other current prediction methods confirmed that the network motifs have indeed been adequately enriched by LaMoFinder for the more sophisticated biological applications such as protein function prediction. For further work, we plan to look into mining labeled and directed network motifs, as many real-world networks can also be modelled with directed graphs.

References

[1] The gene ontology (go) project in 2006. *Nucleic Acids Res*, 34(Database issue):322–326, 2006.

[2] I. Albert and R. Albert. Conserved network motifs allow protein-protein interaction prediction. *Bioinformatics*, 20(18):3346–3352, 2004.

[3] U. Alon. Biological networks: the tinkerer as an engineer. *Science*, 301:1866–67, 2003.

[4] C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Guenoche, and B. Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biol.*, 5(1):R6, 2003.

[5] Jin Chen, Wynne Hsu, Mong Li Lee, and See-Kiong Ng. Dissecting genome-wide protein-protein interactions with repeated and unique network motifs. *SIGKDD*, 2006.

[6] M. Deng, F. Sun, and T. Chen. Assessment of the reliability of protein-protein interactions and protein function prediction. *PSB*, 2003.

[7] M. Deng, K. Zhang, S. Mehta, T. Chen, and F. Z. Sun. Prediction of protein function using protein-protein interaction data. *J. Comp. Biol.*, 10(6):947–960, 2003.

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

[9] S. Itzkovitz, R. Milo, and N. Kashtan. Coarse-graining and self-dissimilarity of complex networks. *Phys. Rev. E*, 71(016127), 2005.

[10] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.

[11] Dekang Lin. An information-theoretic definition of similarity. *ICML*, pages 296–304, 1998.

[12] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 2002.

[13] J. Manning. Geometric symmetry in graphs. *Ph.D thesis, Purdue University*, 1990.

[14] H. W. Mewes, D. Frishman, U. Guldener, et al. Mips: a database for genomes and protein sequences. *Nucleic Acids Res*, 30(1):31–34, 2002.

[15] R. Milo, S. Itzkovitz, and N. Kashtan. Superfamilies of designed and evolved networks. *Science*, 303(5663):1538–1542, 2004.

[16] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[17] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnol*, 18:623–627, 2003.

[18] V. Spirin and L.A. Mirny. Protein complexes and functional modules in molecular networks. *PNAS*, 100(21):12123–12128, 2003.

[19] S. Wernicke and F. Rasche. Fanmod: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, 2006.

[20] S. Wuchty, Z.N. Oltvai, and A.L. Barabasi. *Nature Genetics*, 25:176–179, 2003.

[21] X. Zhou, M. C. Kao, and W. H. Wong. Transitive functional annotation by shortest-path analysis of gene expression data. *Proc. Natl. Acad. Sci. U S A*, 99(20):12783–88, 2002.