Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

A Rule-based Detection of Functional Modules in Protein-Protein Interaction Networks

Jongmin Park, Jaehun Choi, Jaedong Yang, and Soo-Jun Park

Abstract—**In the protein-protein interaction (PPI) network there are many functional modules, each involving several protein interactions to perform discrete functions. Pathways and protein complexes are the examples of the functional modules. In this paper, we propose a rule-based method for detecting the modules. The rule is expressed in terms of triples and operators between the triples. The former represents conceptual relations reifying the protein interactions of a module, and the latter defines the structure of the module with the relations. Additionally, users can define composite rules by composing the predefined rules. The composite rules make it possible to detect modules that are conceptually similar as well as structurally identical to users' queries. The rules are managed in the XML format so that they can be easily applied to other networks of different species. We also provide a visualized environment for intuitionally describing complexly structured rules.**

I. Introduction

THE PPI network, as an integrated system involving protein interactions, is used as an important mean to systematically understand major events that take place in a cell. In this network, there are many functional modules that consist of several protein interactions to perform discrete biological functions[1]. The modules may be viewed as fundamental building blocks of cellular organization. Prominent examples of the blocks are pathways and protein complexes such as "Apoptosis," "Parkinson's disease" pathway, and "Hemoglobin," "Ribosome" complex, etc. Especially, particular functional modules related to some diseases may be a target for drug discoveries. Also, positive and negative side effects can be predicted by analyzing biological reaction with other proteins and interactions in these modules.

Currently, the PPI network data is quickly extracted by the enhanced experimental techniques, such as "Yeast Two-Hybrid"[2], and protein complexes are identified by high-throughput method such as "TAP-MS"[3] and "HMS-PCI"[4]. However, the methods require much time and expense to detect specific user-interested functional modules due to repeated experiments. Therefore, alternative methods have been developed to predict target functional modules before experiments and to detect known functional modules for reviewing and supporting further analysis. Recently, Bader and Hogue[5] suggest a clustering method for searching dense sub-graphs in which each protein has similar functions with the others. This method relies on the fact that protein complexes generally correspond to dense sub-graphs and its components have similar functions with each other. However, the method cannot find other modules having different functions yet conceptually connected. As another approach, Leser[6] defines a query language, PQL(Pathway Query Language) for detecting user-intended modules. This approach can detect relatively various structures by exploiting a set of constraints described in 'WHERE' clause, such as attributes of node and path expressions between nodes. However, it fails to detect modules that have composite structure consisting of several modules. Another drawback is that it does not allow users to directly define structurally complex modules, because the only way to describe the structure of the module is to manually formulate the query.

To resolve the problems, we propose a rule-based method for detecting functional modules. In this method, the rule can detect modules that agree with user-intention. The rule is expressed in terms of triples and operators between the triples. The former represents conceptual relations reifying the protein interactions of a module, and the latter defines the structure of the module with the relations. By composing the predefined rules, user can define new rules that have a composite structure as well. The composite rule makes it possible to detect modules that are conceptually similar as well as structurally identical to the users' queries. The rules are managed in the XML format so that they can be easily applied to the other networks of different species. We also provide a visualized environment for intuitionally describing complexly structured rules.

II. PPI Network Model

In the proposed method, the PPI network is expressed as N = <P, R>, where 'P' is a set of proteins and 'R' is a set of interaction relations among them. Since a relation $r \in R(N)$ can be represented as two specific proteins and a type of interaction between them, we may define it as follows.

$$r = < p_i, p_j, type_{ij} >,$$
$$where\ p_i, p_j \in P(N)\ and\ type_{ij} \in TYPE$$

J. Park, J. Choi, S. Park are with Electronics and Telecommunications Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea(e-mail:jmpark93;jhchoi;psj@etri.re.kr).

J. Yang is with Division of Electronics & Information Engineering, Chonbuk National University, Duckjin-dong, Jeonju, 561-756, Korea (corresponding author to provide phone: +82-63-270-3388; e-mail: jdyang2000@ paran.com).

In this definition, P(N) represents a set of proteins, R(N) represents a set of relations, and 'TYPE' represents a set of types of interaction respectively. Each protein $p \in$ P(N) has detailed properties such as ID, name, gene, and annotation that is a set of ontology terms. Each *type* ∈ TYPE represents a biological interaction between two proteins such as 'bind', 'activate', 'regulate', 'decrease', 'increase', etc. Also, the PPI network includes many functional modules. In other words, because the module 'M' is the sub-network of the PPI network, it can be represented as the similar formulation as the PPI network's :

$$M =< P', R' >,$$
$$where\ \ p'(M) \subseteq P(N)\ \ and\ \ R'(M) \subseteq R(N)$$

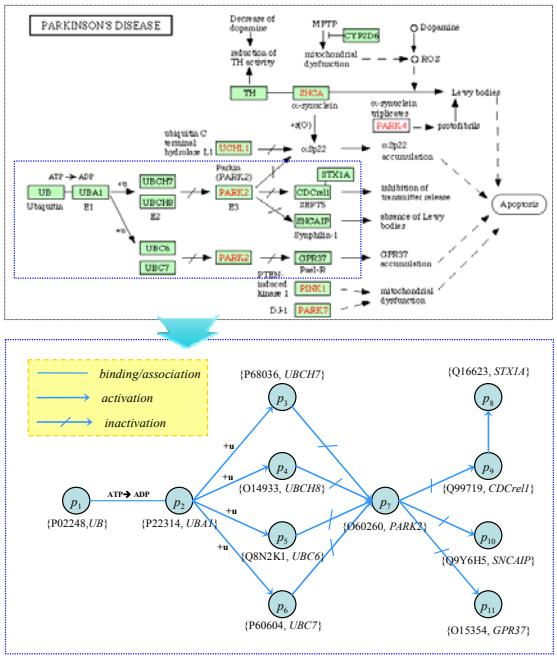

Fig. 1 Functional module 'M_1' transformed from the rectangle part of "Parkinson's disease pathway"

Fig. 1 shows a module 'M_1' together with a set of relations. It is transformed from a rectangle part of "Parkinson's disease pathway" extracted from KEGG[7]. In the 'M_1', gene nodes are dealt with proteins and types of relation between the nodes are used just as it is. Also, duplicated nodes are annotated by a single protein. For example, the node 'UBA1' corresponds to 'p_2' with a identifier 'P22313'. The type of relation between 'p_2' and 'p_3' is $type_{23}$ = 'activation', which is the same as the type of interaction between 'UBA1' and 'UBCH7'. Duplicated nodes 'PARK2' are annotated by 'p_7'.

III. Rule Specification

A rule for detecting functional modules consists of triples and operators between these triples. At first, we define the triple for describing conceptual relations. It is as follows.

$$t =< n_i, n_j, type_{ij} >,$$
$$where\ \ n_i, n_j \in P\ or\ GO\ \ and\ \ type_{ij} \in TYPE$$

In this definition, GO(Gene Ontology)[8] is a set of ontology terms used to describe the 'annotation' property of each protein. For detecting specific relations included in the module, users can define the triple with a protein directly or ontology terms. For example, the triple t = <n_1, n_2, 'inactivation'> can be defined with 'n_1' (ontology term 'protein binding') and 'n_2' (protein 'p_3').

Next, operators are classified into connection operator and logical operator. The former is used to express various types of structural connection between two triples, and the latter expresses logical relationships between them. As connection operators, we define '•' (Arbitrariness) and '*' (Association), and as logical operators, we adopt '¦' (Logical OR) and '&' (Logical AND).

We can express the structure of the rule by a regular grammar. The rule('*RULE*') is expressed as single triple('*t*') or a composite rule ('*COMPOSITE_ RULE*'). The composite rule is composed of predefined rules and operators ('*OPERATOR*') between the rules, or is expressed as a predefined rule with a precedence(parenthesis). A rule used in the composite rule may be a single triple or recursively another composite rule.

RULE → *t* | ***COMPOSITE_RULE***
COMPOSITE_RULE → ***RULE OPERATOR RULE***
| '('***RULE***')'
OPERATOR → '•' | '*' | '&' | '¦'

The composite rules are useful when detecting complex modules composed of separable modules. For example, the following is the composite rule to detect the pathway of "Parkinson's disease." It is the complex module composed of several separable pathways such as "Inhibition of transmitter release," "Absence of lewy body," "GPR37 accumulation," etc.

Inhibition of transmitter release →
<n_1, n_2, $type_{12}$> * (<n_2, n_3, $type_{23}$> | <n_2, n_4, $type_{24}$>) * …
Absence of lewy body → …
(<n_5, n_7, $type_{57}$> | <n_6, n_7, $type_{67}$>) * <n_7, n_{10}, $type_{710}$>
…
Parkinson's disease →
Inhibition of transmitter release |
Absence of lewy body | ...

It is defined by combining the predefined rules such as '*Inhibition of transmitter release*' and '*Absence of lewy body*' used for detecting the corresponding pathways.

IV. Rule Evaluation

Fig. 2 shows the entire process of rule evaluation for detecting functional modules within the network N_1, where $p_{i=1,\ldots,6} \in P(N_1)$, $n_j \in p_i.annotation$, and $n_{j=1,\ldots,6} \in$GO. This figure is used to explain the evaluation process of node,

triples, and operators, which are described in the following steps.

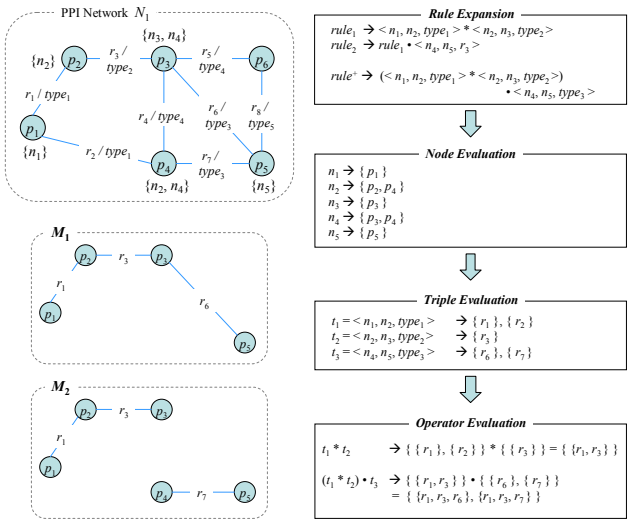

Fig. 2 Entire process of rule evaluation

A. Rule expansion

Target rules for evaluation, '$rule_1$' and '$rule_2$', are expanded into a series of simple rules consisting of pure triples and operators. '$rule^+$' is generated as a real rule for evaluation by expanding the target rules.

B. Node evaluation

Each node 'n_j' included in '$rule^+$' is indexed by matched proteins $\{p_j\}$. According to the properties of nodes defined in a triple, each node may be matched and evaluated with more than one protein –for example, $n_2 \rightarrow \{p_2, p_4\}$.

In case the node is defined as a protein, it is indexed directly, because each protein has a detailed information using SWISS-PROT[9]. Therefore, an identifier coincides with exactly one protein. In case the node is defined as GO terms, the node(n_i) can be matched with several proteins that have conceptually similar 'annotation'(n_j) as well as exactly the same. For examples, if 'n_i' is defined with 'regulation of exocytosis', 'annotation'(n_j) of matched proteins can be defined with more generalized terms of the 'n_i', such as {'exocytosis'/0.74, 'regulation of transport'/0.74,…}, and more specific terms such as {'positive regulation...'/0.74, '…calcium ion- dependent'/0.54,…}.

C. Triple evaluation

A triple is evaluated against a set of modules satisfying each of its conditions. For that reason, evaluation is performed by calculating similarity between the triple and the modules. The similarity 'S' is calculated by combining the similarity 'S_1' between two nodes of the triple and the two proteins of a relation in a module, and the similarity 'S_2' between the relation type of the triple and the relation of each module.

Similarity between nodes in the triple and protein in the relation is evaluated using the index n_j that is made in the previous step, and matched relations are indexed by t_i. For example, in Fig. 2, $t_1 = <n_1, n_2, type_1>$ is evaluated as follows: With $n_1 \rightarrow \{p_1\}$ and $n_2 \rightarrow \{p_2, p_4\}$, two relations, $r_1 = <p_1, p_2, type_1>$ between 'p_1' and 'p_2', $r_2 = <p_1, p_4, type_1>$ between 'p_1' and 'p_4', are identified. Since the type of 'r_1' and 'r_2' are matched with the type of 't_1', $\{r_1\}$ and $\{r_2\}$ match the triple 't_1'.

D. Operator evaluation

The rule is evaluated by applying operations to a set of modules. These operations are determined by operators defined in the rule.

First, '*'(Association) is used to detect modules composed of relations that match two given rules and connected directly between these matched relations. For example, $rule = t_1 * t_2$, where $t_1 = <n_1, n_2, type_1>$ and $t_2 = <n_2, n_3, type_2>$, is evaluated as follows; by cross union between $t_1 \rightarrow \{r_1\}$, $\{r_2\}$ and $t_2 \rightarrow \{r_3\}$, two modules, $\{r_1, r_3\}$ and $\{r_2, r_3\}$, are matched. Among them, $\{r_1, r_3\}$ is selected, because 'r_1' and 'r_3' is connected through protein 'p_2', but there is no connection between 'r_2' and 'r_3'.

Next, '•'(Arbitrariness) is used to detect modules composed of relations that match two given rules. For example, the $rule = (t_1 * t_2) \bullet t_3$ where $(t_1 * t_2) \rightarrow \{r_1, r_3\}$ and $t_3 \rightarrow \{r_6\}$, $\{r_7\}$ is evaluated as follows; by cross union between $t_1 * t_2$ and t_3, two modules, $\{r_1, r_3, r_6\}$ and $\{r_1, r_3, r_7\}$, are obtained. Next, as a final result of '$rule^+$', two modules are suggested; $\{r_1, r_3, r_6\}$ consists of all connected relations, while $\{r_1, r_3, r_7\}$ is composed of two separate sub-graphs (see Fig. 2).

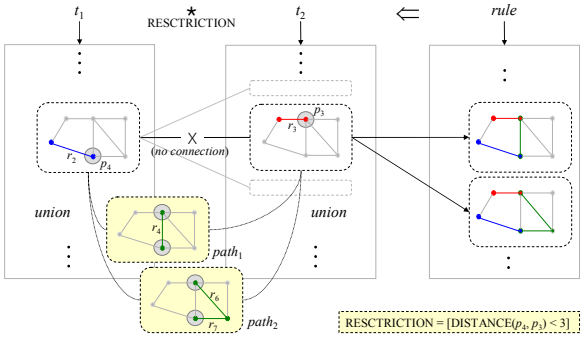

Fig. 3 Evaluation process of $t_1 *_{\text{RESTRICTION}} t_2$

By imposing on the '*' operator constraints of connection between two rules, it is also possible to detect modules that are indirectly connected. For example, $rule = t_1 *_{\text{RESCTRICTION}} t_2$, and RESCTRICTION = [DISTANCE(n_4, n_3) < 3] may detect modules that are connected directly and indirectly between 't_1' and 't_2'. This constraint specifies that 'n_4' and 'n_3' need to be connected by * with the number of intervening nodes less than 3. 'n_4' is matched with two proteins 'p_3' and 'p_4', and 'n_3' is matched only with 'p_3'. Since 't_1' is evaluated only with 'p_4', this constraint is translated into RESCTRICTION = [DISTANCE(p_4, p_3) < 3]. There are two paths satisfying the restriction, $path_1$ and $path_2$; $path_1$

connects two modules via 'r_4', and $path_2$ via 'r_7', 'r_6'. Therefore, restricted cross union between 't_1' ,'t_2', and the paths is calculated as a $\{r_2, r_4, r_3\}$ and $\{r_2, r_7, r_6, r_3\}$. Directly connected module $\{r_1, r_3\}$ is also included in final result.

Logical operators express logical relationships between modules matched with two rules. '|'(OR) means union and '&'(AND) means intersection between two sets of modules.

V. Design and Implementation

There are three ways to define the rule of user-intended functional module in the proposed method. First, users can define the rule in text format as described in this paper. Second, they can define it in visual format by representing triples in terms of nodes and links. The operators may be represented as several different graphical notations. Third, user can write out all definitions related to the rule in the XML format. Both the text form and the visual form of the rule are transformed automatically to the XML format before processing the rule.

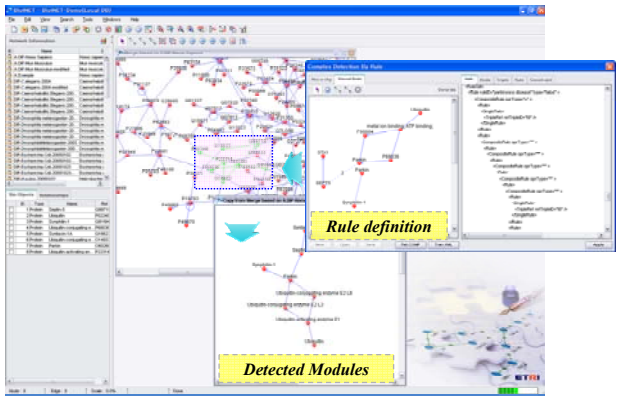

Fig. 4 Rule-based detection in the visual environment

Figure 4 shows the process of rule-based detection in the visual environment. The rule represents a part of "Parkinson's disease" pathway and defines with various properties of protein node, such as ID, protein name, gene name, and protein function using GO terms. Each solid line represents a triple and each connection between two solid line represents the '*' operation. Dotted line between two protein nodes represents the '*' operation with path constraints. Disconnected graph means the '•' operation between two disconnected sub-graphs.

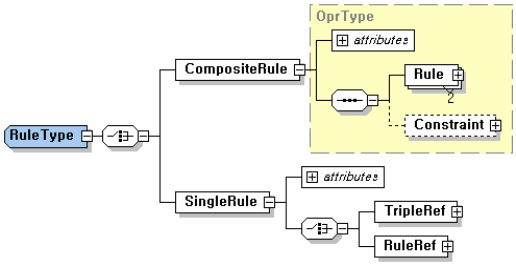

Fig. 5 XML Schema for representing the rule

Figure 5 shows the XML schema for representing the rules and storing the defined rules into a knowledge base. The 'RuleSet' consists of a set of rules used for representing the rules. 'SingleRule' is used for referencing a triple and a predefined rule. 'CompositeRule' is needed for defining a rule with operators between predefined rules.

VI. Conclusions and Future Works

In this paper, we designed and implemented a rule-based method for detecting meaningful functional modules in a large and complex PPI network. On a visualized environment, rules are used to express the biological features of the modules in terms of triples and operators between the triples. The former represents conceptual relations reifying the protein interactions of a module, and the latter defines the structure of the module with the relations. Our method may be applied to other related researches in three directions; first, once representing the general domain knowledge of a known module into a rule, users can easily capture proteins included in the module as well as the concrete structure of their interactions. Second, by detecting complex modules with the corresponding composite rules, to analyze the interaction between the modules is also possible. Third, by applying the rule to the other networks of different species, user can predict similar functional modules in other species.

Although we proposed basic rule operators, additional rule operators are required to naturally express user intention. Additionally, we may need more sophisticated methods to precisely estimate the similarity between triples and protein interaction relations using the other information such as protein's amino acid sequence and protein structure.

References

[1] Ravasz E, Somera AL, Mongru DA, Oltvai ZN, and Barabasi AL, "Hierarchical organization of modularity in metabolic networks.", *Science.*, 2002, 297(5586):1551-5.

[2] Ito T, Chiba T, Ozawa R, Yoshida M, Hattori M, and Sakaki Y, "A comprehensive two-hybrid analysis to explore the yeast protein interactome.", *Proc. Natl Acad. Sci.,* 2001, 98(8):4569-74.

[3] Gavin AC, Bosche M, Krause R, Grandi P, Marzioch M, et al., "Functional organization of the yeast proteome by systematic analysis of protein complexes.", *Nature.* 2002, 415(6868):141-7.

[4] Ho Y, Gruhler A, Heilbut A, Bader GD, Moore L, et al., "Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry.", *Nature.*, 2002, 415:180-183

[5] Bader GD and Hogue CW, "An automated method for finding molecular complexes in large protein interaction networks.", *BMC Bioinformatics*., 2003, 4:2.

[6] Leser U., "A query language for biological networks.", *Bioinformatics.*, 2005, vol.21 Suppl. 2:ii33-ii39.

[7] Kanehisa M, Goto S, Kawashima S, Okuno Y, Hattori M, "The KEGG resource for deciphering the genome.", *Nucleic Acids Res.*, 2004, 32(Database issue):D277-80.

[8] Harris MA, Clark J, Ireland A, Lomax J, Ashburner M, et al., "The Gene Ontology (GO) database and informatics resource.", *Nucleic Acids Res.*, 2004, 32(Database issue):D258-61.

[9] Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, et al., "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003.", *Nucleic Acids Res.*, 2003, 31(1):365-370.