Prediction of Protein Interaction with Neural Network-Based Feature Association Rule Mining

Jae-Hong Eom and Byoung-Tak Zhang

Biointelligence Lab., School of Computer Science and Engineering,
Seoul National University,
Seoul 151-744, South Korea
{jheom, btzhang}@bi.snu.ac.kr

Abstract. Prediction of protein interactions is one of the central problems in post–genomic biology. In this paper, we present an association rule-based protein interaction prediction method. We adopted neural network to cluster protein interaction data, and used information theory based feature selection method to reduce protein feature dimension. After model training, feature association rules are generated to interaction prediction by decoding a set of learned weights of trained neural network and by mining association rules. For model training, an initial network model was constructed with public *Yeast* protein interaction data considering their functional categories, set of features, and interaction partners. The prediction performance was compared with traditional simple association rule mining method. The experimental results show that proposed method has about 96.1% interaction prediction accuracy compared to simple association mining approach which achieved about 91.4% accuracy.

1 Introduction

It is known that protein–protein interactions (PPIs) are fundamental reactions in the organisms and play important roles by determining biological processes. Therefore, comprehensive description and analysis of PPIs would significantly contribute to the understanding of biological phenomena and problems. After the completion of the genome sequence of yeast (*Saccharomyces cerevisiae*), researchers have undertaken the task of functional analysis of the yeast genome comprising more than 6,300 proteins [1], and abundant interaction data have been produced by many research groups. Thus, fresh methods to discover novel knowledge from the interaction data through the analysis of these data are needed.

A variety of attempts have been tried to predict protein functions and interactions with various data such as gene expression, PPI data, and literature analysis. Analysis of gene expression data through clustering also adopted to predict functions of unannotated proteins based on the idea that genes with similar functions are likely to be co-expressed [2, 3]. Park *et al.* [4] analyzed interactions between protein domains in terms of the interactions between structural families of evolutionarily related domains. Iossifov *et al.* [5] and Ng *et al.* [6] inferred new interaction from existing interaction data. Even though there are many other approaches for analyzing and predicting protein interactions, however, many approaches to protein interaction analysis suffered from high dimensional property of data which have thousand of features [7].

I. King et al. (Eds.): ICONIP 2006, Part III, LNCS 4234, pp. 30 – 39, 2006.

In this paper, we propose an adaptive neural network based feature association mining method for PPI prediction. We used additional association rules for interaction prediction those are generated by decoding set of learned weights of neural network. We presumed that association rules decoded from neural network would make the prediction procedure more robust for unexpected error factors by accounting relatively robust characteristic of neural networks (e.g., error factors would be false positive or negative interactions those are provided to the prediction model).

Basically, we use adaptive resonance theory (ART) [8] as an adaptive neural network clustering model to build prediction model. We used ART-1 [9], modified version of ART [10], to cluster binary vectors. The advantage of using ART-1 algorithm for grouping of feature abundant interaction data is that it adapts the changes in new protein interactions without losing key information learned from other interactions trained previously. We assumed 'protein–protein interaction' of yeast as 'feature–to–feature' association of each interacting proteins. To analyze PPIs with respect to their interaction class with their feature association, we use as many features as possible from several major public databases such as (Munich Information Center for Protein Sequences) MIPS and SGD (Saccharomyces Genome Database) [11, 12] to build rich feature vector for each protein interaction. We used the same approach of Rangarajan *et al*. [13] for clustering model design and we also use the same feature selection filter of Yu *et al*. [14] to reduce computational complexity and improve the overall learning performance by eliminating non-informative features.

This paper is organized as follows. In Section 2, we introduce feature selection filter and describe overall architecture of ART-1 based protein interaction clustering model. In Section 3, we present detailed neural network training method with PPI data and the decoding method of association rules extracted from trained network. In Section 4, we present the representation scheme of protein interaction for neural network input, association mining, and experimental results. Finally, concluding remarks and future works are given in Section 5.

2 Feature Dimension Reduction and Protein Cluster Learning

Feature Dimension Reduction by Feature Selection

A set of massive features for each protein and interacting pairs are built by utilizing several public protein databases [11, 12, 15, 16, 17]. Generally, feature selection is necessary when dealing with such high dimensional (feature dimension) data. In our study, set of features having no information of its association with other proteins are removed by applying feature selection. To filter out non-informative features we applied entropy and information gain-based measure, *symmetrical uncertainty* (*SU*-value), as a measure of feature correlation [18]. The procedures of the correlation-based feature dimension reduction filter of Eom *et al*. [7] used for our application.

Enriching Protein Features by Neural Network-Based Cluster Learning

We use ART-1 neural network to group the class of PPIs by their 13 functional classes and the class of interacting counterparts. In our ART-1 based clustering, a protein interaction is represented as a prototype vector that is a generalized representation of a

set of features of each interacting proteins. The degree of similarity between the members of each cluster can be controlled by changing the value of the vigilance parameter ρ. We analyzed the cluster formed by using the ART-1 technique by varying the vigilance parameter between the values 0.2 and 0.8. Figure 1 represents the architecture of ART-1 based clustering model and the PPI_i stand for each protein interaction and it includes set of features of two interacting proteins. The overall procedure for clustering protein interactions with the ART-1 based clustering model is described in the Appendix. The basic layout of this procedure is identical with the work of Rangarajan *et al.* [13]. The set of weights of trained neural network were decoded as a form of association rule with the 'weight-to-rule' decoding procedures described in Figure 3 to enrich the protein features.

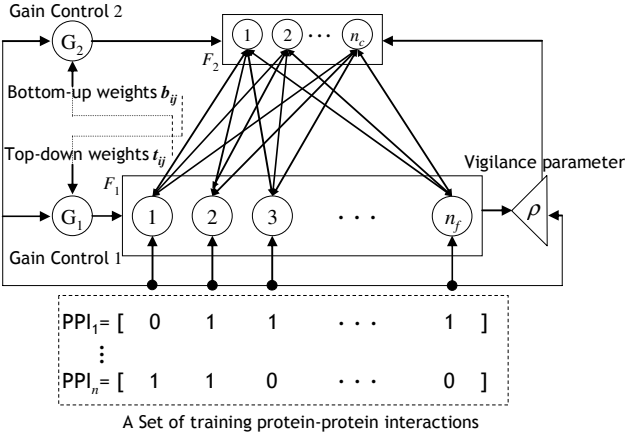

Fig. 1. The schematic architecture of neural network (ART-1) based clustering model (More detailed model constructions are described in [19])

3 Rule Extraction from Trained Neural Network

Learning Feature Associations with Neural Network

A supervised artificial neural network (ANN) uses a set of training examples or records. These records include N attributes. Each attribute, A_n (n = 1, 2, … , N), can be encoded into a fixed length binary substring $\{x_1 \ldots x_i \ldots x_{m(n)}\}$, where $m(n)$ is the number of possible values for an attribute A_n. The element x_i = 1 if its corresponding attribute value exists, while all the other elements = 0. Then, the proposed number of input nodes, $\boldsymbol{I}$, in the input layer of ANN can be given by $I = \sum_{n=1}^{N} m(n)$.

The input attributes vectors, $\boldsymbol{X}_m$, to the input layer can be rewritten as $\boldsymbol{X}_m = \{x_1 \ldots x_i \ldots x_I\}_m$, m = (1,2,…, M) where M is the total number of input training patterns. The output class vector, $\boldsymbol{C}_k(k = 1, 2, \ldots, K)$, can be encoded as a bit vector of a fixed length K as follows $C_k\{\psi_1 \ldots \psi_k \ldots \psi_K\}$ where K is the number of different possible classes. If the output vector belongs to class$_k$ then the element ψ_k is equal to 1 while all the other elements in the vector are zeros. Therefore, the proposed number of

output nodes in the output layer of ANN is *K*. Accordingly the input and the output nodes of the ANN are determined and the structure of the ANN is shown in Figure 2. The ANN is trained on the encoded vectors of the input attributes and the corresponding vectors of the output classes. The training of ANN is processed until the convergence rate between the actual and the desired output will be achieved. The convergence rate can be improved by changing the number of iterations, the number of hidden nodes (***J***), the learning rate, and the momentum rate.

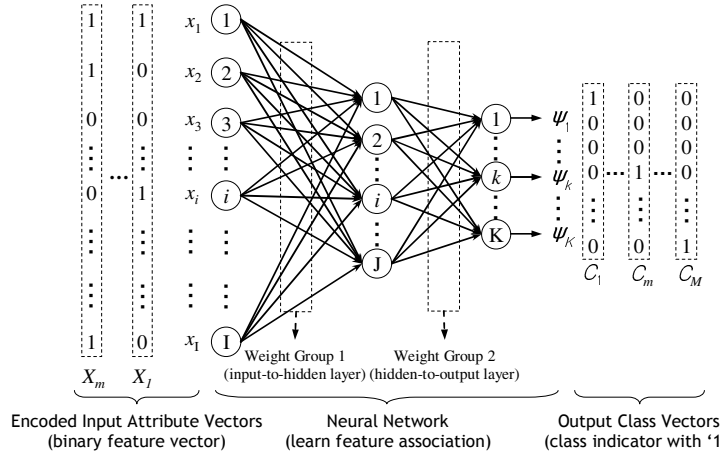

Fig. 2. The structure of the artificial neural network for feature association learning. The two weight groups (WG1 and WG2) are decoded into association rule after network training.

By ANN training, two groups of weights are obtained. The first group, $(WG1)_{i,j}$, is the weights between the input node *i* and the hidden node *j*. The second group, $(WG2)_{j,k}$, is the weights of the hidden node *j* and output node *k*. A sigmoid is used for the activation function of the hidden and output nodes. Then, the total input to the *j*–th hidden node (IHN_j) and the output of the *j*–th hidden node (OHN_j) are given by

$$IHN_j = \sum_{i=1}^{I} x_i (WG1)_{i,j} \ , \ OHN_j = \frac{1}{1+e^{-\left[\sum_{i=1}^{I} x_i (WG1)_{i,j}\right]}}. \tag{1}$$

Nextly, the total input to the *k*–th output node, ION_k, is given by

$$ION_k = \sum_{j=1}^{J} (WG2)_{j,k} \frac{1}{1+e^{-\left[\sum_{i=1}^{I} x_i (WG1)_{i,j}\right]}}. \tag{2}$$

Then, the final value of the *k*–th output node, ψ_k, is given by

$$\psi_k = \left\{ \frac{1}{1+e^{-\left[\sum_{j=1}^{J} WG2_{j,k} \left(\frac{1}{1+e^{-\left[\sum_{i=1}^{I} x_i (WG1)_{i,j}\right]}} \right)\right]}} \right\}. \tag{3}$$

The function, $\psi_k = f(x_i, (WG1)_{i,j}, (WG2)_{j,k})$ is an exponential function in x_i since $(WG1)_{i,j}$, $(WG2)_{j,k}$ are constants and its maximum output value is equal to one. Then, we can say that "An input vector, $\boldsymbol{X_m}$, belongs to a $class_k$ *iff* $\boldsymbol{\psi_k} \in \boldsymbol{C_m} = 1$ and all other elements in $\boldsymbol{C_m} = 0$."

With the given parameters,
- A: set of attributes. - α: set of attributes (conditional), β: set of result attributes (result).
- n: the number of total attribute, γ: the length of feature n.
- G: set of the best b chromosome, g: a chromosome in G.
- b: the number of total chromosome ($|G| = b$).
- μ: the number of total rule found by association rule mining.

Repeat Step 1 to Step 5, for all g in G.

1. **Create** temporary empty rule t: $\{\alpha\} \rightarrow \{\beta\}$, and **Set** $\alpha = \beta = \varphi$.
2. **Divide** best chromosome into $2n$ segments.
   (Each segment in 1 to n is corresponds to each attribute of A_n for condition of rule).
   (Each segment in n+1 to $2n$ is corresponds to each attribute of A_n for result of rule).
3. For all i, $i = 1$ to n.
   3.1 For all j, $j = 1$ to γ.
      3.1.1 If the corresponding bit of conditional chromosome is equal to '1',
         $\alpha \leftarrow \alpha \cup A_j$.
   3.2 **Connect** all feature in α with operator 'AND'.
4. For all i, $i = n$+1 to $2n$.
   4.1 For all j, $j = 1$ to γ.
      4.1.1 If the corresponding bit of result chromosome is equal to '1',
         $\beta \leftarrow \beta \cup A_j$.
   4.2 **Connect** all feature in β with operator 'AND'.
5. For all k, $k = 1$ to μ.
   5.1 If any $R(k) \equiv t$ then $R \leftarrow R - R(k)$ else $R \leftarrow R \cup t$.

Return final rule set R
(R = rules mined by association mining + rules decoded by top b chromosome decoding).

Fig. 3. The rule decoding procedures from the selected best chromosome

Deriving Association Rules from Trained Network with GA-Based Decoding

To extract relations (rules) among the input attributes, $\boldsymbol{X_m}$ relating to a specific $class_k$ one must find the input vector, which maximizes $\boldsymbol{\psi_k}$. This is an optimization problem and can be stated as $\psi_k(x_i)$ by considering binary data feature vector $\boldsymbol{x}$. In $\psi_k(x_i)$, x_i are binary values (0 or 1). Since the objective function $\psi_k(x_i)$ is nonlinear and the constraints are binary, it is a nonlinear integer optimization problem. Genetic algorithm (GA) can be used to solve this optimization problem by maximizing the objective function $\psi_k(x_i)$. In this paper, we used conventional generational-GA procedures with this objective function $\psi_k(x_i)$ to find the best chromosome which provided as an input of neural network and produce best network output (i.e. highest prediction accuracy).

After we obtain best chromosomes which produces best network output, we decoded these chromosome into the form of association rules (here, we call this association rule as '*neural feature association rule*' because they are extracted from trained neural network). To extract a rule for $class_k$ from the best chromosomes selected by GA procedures, we decoded them with several procedures presented in Figure 3.

4 Experimental Results

Protein Interopaction as Binary Feature Vector

An interaction is represented as a pair of two proteins that directly binds to each other. This protein interaction is represented by binary feature vector of interacting proteins and their associations. Figure 4 describes this interaction representation processes. Interactions prepared through these processes are provided to the neural network-based clustering and to the prediction model to group each protein interaction class and learn the association of features which generalize the interactions. Then, the constructed cluster prototype is used to predict the classes of protein interactions presented in the test step. The 13 functional categories of interacting protein from MIPS [11] which is known for the most reliable curated protein interaction database in current literature are used to evaluate the category classes clustering accuracy.

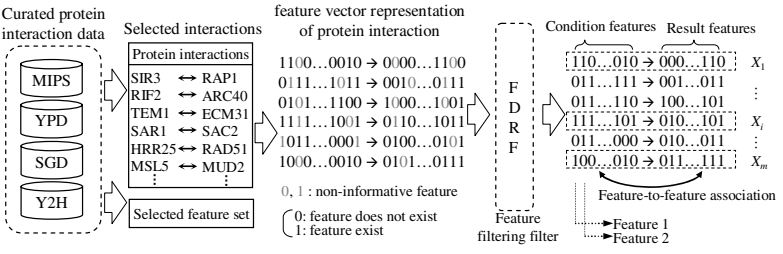

Fig. 4. The feature vector representation of protein interactions. Each interaction is represented as a binary feature vector (whether the feature exists or not). The feature dimension reduction filter (FDRF) marks those features as 'don't care' which have *SU* value less than given *SU* threshold δ to remove non-informative features so as to improve the performance of clustering model. The marked features are regarded when train clustering model. The resulting vectors of interactions are provided to the neural network learning model as network input, describe in Figure 2, for model training, testing, and generation of neural feature association rules.

Data Sets

Each *yeast* protein has various functions or characteristics which are called 'feature.' In this paper, set of features of each protein are collected from public genome databases [11, 12, 15, 16, 17]. We use similar features of protein interaction of Oyama *et al.* [20] which include EC numbers (from SWISS-PROT), SWISSPROT/PIR keywords, PROSITE motifs, bias of the amino acids, segment cluster, and amino acid patterns, etc. A major protein pairs of the interactions are also obtained from the same data source of Oyama *et al.* [20]. These dataset include various experimental data such as

YPD and Y2H by Ito *et al*. [16] and Uetz *et al*. [17]. Additionally, we used SGD to construct more abundant feature set [12]. Table 1 shows the statistics of each interaction data source and the number of features before and after the application of FDRF.

Table 1. The statistics for the dataset

Data Source	# of interactions	# of initial features	# of filtered features
MIPS [11]	10,641	6,232 (total)	1,293 (total)
YPD [15]	2,952		
SGD [12]	1,482		
Y2H (Ito *et al*.) [16]	957		
Y2H (Uetz *et al*.) [17]	5,086		

Experiment Procedures

First, we predicted the classes of new PPIs with neural network for their 13 functional categories obtained from MIPS [11]. The accuracy of class prediction is measured whether the predicted class of interaction is correctly corresponds to the class of MIPS. After this step, we constructed feature association rule from this trained neural network with similar procedure with Figure 3.

Next, we trained another neural network with PPI data represented as binary feature vector according to the method in Figure 4. After the model training, we extracted again feature association rules from the model with the procedure in Figure 3. Then we predicted test PPIs with these two set of association rules and measured the prediction accuracy of each approaches with 10-fold cross-validation.

Results

Table 2 show the interaction prediction performance of various combination of associantion mining, information theory based feature filtering, and exploitation of rules derived from trained neural network.

Table 2. The comparison of prediction accuracies of the proposed methods. The effect of the FDRF-based feature selection and neural network-based are shown in terms of prediction accuracy. For filtered interaction vectors by FDRF, the feature association-based prediction model with neural association rule (☆) shows the best performance (*Asc*: association rule based prediction. *FDRF* + *Asc*.: prediction based on association rule mined from filtered feature vectors. *Asc*. + *N-Asc*.: rule based prediction with association rule and the rule derived from trained neural network. *FDRF* + *Asc*. + *N-Asc*.: combination of all methods).

Prediction method	Number of interactions			Accuracy (|P|/|T|)
	Training set Size	Test set (T)	Predicted correctly (P)	
Asc. (△)	4,628	463	423	91.4 %
FDRF + Asc. (▽)	4,628	463	439	94.8 %
Asc. + N-Asc. (◇)	4,628	463	432	93.3 %
FDRF + Asc. + N-Asc. (☆)	4,628	463	445	96.1 %

In Table 2, simple association mining approach (△) achieved the lowest performance. The number of total feature used in this approach was 6,232. This is quite high feature dimension. So, we can guess that it may includes lots of non-informative and redundant features and these features may affect the prediction accuracy in negative way by interfering correct rule mining. This assumption confirmed by investigating the result of second approach, FDRF + Asc. (▽), association mining with non-informative and redundant feature filtering. This feature filtering approach improved overall prediction performance about 3.4% than the first approach. But the third approach, Asc. + N-Asc. (◇), prediction with the rules from association rule mining and the rule derived from trained neural network only improved overall prediction performance about 1.9% than the first approach.

This result can be explained again with the feature dimension problem. In this third approach, there also exist redundant and non-informative garbage features which decrease the prediction performance. But in this approach, eventhough there still lots of garbage features, the over all performance improved about 1.9%. This is the effect of the rule exploitation derived from trained neural network. This inference can be confirmed again by investigating the result of fourth approach, FDRF + Asc. + N-Asc (☆), prediction with the rule from association mining and the rule derived from trained neural network along with feature filtering. Non-informative and redundant features are filtered out in this approach. Consequently, this approach improved over all prediction accuracy up to about 4.7%. These results are outperform other several approaches including *k*-NN (86.4%), support vector machine (93.3), structure and sequence conservation-based prediction (88.5%), and generative stochastic model with MCMC estimation (94.8%) in prediction accuracy [21].

Thus, we can say that both the information theory-based feature filtering and the exploitation of the rule derived from trained neural network and conventional association rule mining methods are helpful for improving overall performance of feature-to-feature association-based PPI prediction. By considering these experimental results, the proposed approaches will be useful as a data preprocessing and prediction methods especially when we handle the data which have many features.

5 Conclusions

We presented neural network based protein interaction learning and association rule mining method from feature set and trained neural network model for PPI prediction task. Also we applied information theory-based feature selection procedure to improve the performance of trained feature association learning model. The proposed method (combination of all methods) achieved accuracy improvement about 4.7%. From the experimental results, it is suggested that the neural network-based feature association learning model could be used for more detailed investigation of the PPIs by learning the hidden patterns of the data having many features and implicit associations among them. From the results, we can conclude the proposed method is suitable for efficient analysis of PPIs through learning their hidden 'feature associations.'

However, to overcome the false positive rates of current public interaction database is one of the important issues for more reliable prediction. The computational complexities caused by using neural network and GA is another issues to resolve for efficient predictions. Also, more biological features such as pseudo amino acid

composition or protein localization facts will be also helpful for improving overall prediction accuracy and should be considered in the future works.

References

1. Goffeau, A., Barrell, B.G., *et al*.: Life with 6000 genes. *Science* **274** (1996) 546–67.
2. Eisen, M.B., Spellman, P.T., *et al*.: Cluster analysis and display of genomewide expression patterns. *Proc. Natl. Acad. Sci. USA* **95** (1998) 14863–68.
3. Pavlidis, P. andWeston, J.: Gene functional classification from heterogeneous data. In *Pro. of the 5th Int'l Conf. on Comp. Molecular Biology* (2001) 249–55.
4. Park, J., *et al*.: Mapping protein family interactions: intramolecular and intermolecular protein family interaction repertoires in the PDB and yeast. *J. Mol. Biol.* **307** (2001) 929–39.
5. Iossifov, I., Krauthammer, M., *et al*. Probabilistic inference of molecular networks from noisy data sources. *Bioinformatics* **20**(8) (2004) 1205–13.
6. Ng, S.K., Zhang, Z., *et al*. Integrative approach for computationally inferring protein domain interactions. *Bioinformatics* **19**(8) (2003) 923–29.
7. Eom, J.-H., *et al*.: Prediction of implicit protein–protein interaction by optimal associative feature mining. *Lecture Notes in Computer Science* **3177** (2004) 85–91.
8. Carpenter, G.A. and Grossberg, S.: A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision*, *Graphics and Image Processing* **37** (1987) 54–115.
9. Barbara, M.: ART1 and pattern clustering. In proceedings of the 1988 connectionist models summer 1988. Morgan Kaufmann (1988) 174–85.
10. Heins, L.G. and Tauritz, D.R.: Adaptive resonance theory (ART): an introduction. *Internal Report* 95-35, Dept. of Comp. Sci., Leiden University, Netherlands (1995) 174–85.
11. Mewes, H.W., Amid, C., *et al*.: MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Res.* **32**(1) (2004) D41–44.
12. Christie, K.R., Weng, S., *et al*.: Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms. *Nucleic Acids Res.* **32**(1) (2004) D311–14.
13. Rangarajan, S.K., Phoha, V.V., *et al*.: Adaptive neural network clustering of web users. *IEEE Computer* **37**(4) (2004) 34–40.
14. Yu, L. and Liu, H.: Feature selection for high dimensional data: a fast correlation-based filter solution. In *Proceeding of ICML-03* (2003) 856–863.
15. Csank, C., Costanzo, M.C., *et al*.: Three yeast proteome databases: YPD, PombePD, and CalPD (MycoPathPD). *Methods Enzymol*. 350 (2002) 347–73.
16. Ito, T., Chiba, T., *et al*.: A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl Acad. Sci. USA* **98** (2001) 4569–4574.
17. Uetz, P., Giot, L., *et al*.: A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae. *Nature* **403** (2000) 623–627.
18. Press, W.H., Flannery, B.P. *et al*.: Numerical recipes in C: The Art of Scientific Computing (2nd Ed.). *Cambridge University Press*, Cambridge (1992) 633–634.
19. Eom, J.-H., *et al*.: Adaptive Neural Network based Clustering of Yeast Protein-Protein Interactions. *Lecture Notes in Computer Science* **3356** (2004) 49–57.
20. Oyama, T., Kitano, K., *et al*.: Extraction of knowledge on protein–protein interaction by association rule discovery. *Bioinformatics* **18** (2002) 705–14.
21. Eom, J.-H. and Zhang, B.-T.: Prediction of implicit protein–protein interaction using optimal associative feature rule, *Journal of Korea Information Science Society: Software and Application* **33**(4) (2006) 365–77.

Appendix

The procedures of ART-1 based protein interaction clustering.

Given array of input protein interaction vectors ***PPI*** and vigilance parameter ρ,

1. **Initialize**
   1.1 **Set** the value of gain control G_1 and G_2,
   $$G_1, G_1 = \begin{cases} 1 \; if \; input \; PPI_I \neq 0 \; and \; output \; from \; F_2 \; Layer = 0 \\ 0 \; for \; all \; other \; cases \end{cases}$$
   1.2 **Set** all nodes in F_1 layer and F_2 layer to 0.
   1.3 **Set** all weight of top-down weight matrix, $t_{ji} = 1$.
   1.4 **Set** all weight of bottom-up weight matrix,
   $b_{ij} = \left(1/(n_f + 1)\right)$ (n_f = the size of the input feature vector).
   1.5 **Set** the vigilance parameter ρ (0.2 to 0.7).

2. **Repeat** Step 2.1 to 2.7, for all protein-protein interaction vector PPI_I.
   2.1 **Read** randomly chosen interaction vector
   $PPI_I = (P_1, P_2, \cdots, P_{I=n_f})$, where $P_1 = 0$ *or* 1.
   2.2 **Compute Input** y_j for each node in F_2, $y_j = \sum_{i=1}^{n_f} P_i \times b_{ij}$.
   2.3 **Determine** ***k***, $y_k = \sum_{j=1}^{\#\ of\ nodes\ in\ F_2} \max(y_j)$.
   2.4 **Compute Activation**, $X_k^* = (X_1^*, X_2^*, \cdots, X_{i=5,240}^*)$ for the node k in F_1
   Where, $X_i^* = t_{ki} \times P_i \; (i = 1, \cdots, n_f)$.
   2.5 **Calculate similarity** δ, between X_i^* and PPI_I: $\delta = \dfrac{\left\| X_k^* \right\|}{\left\| PPI_I \right\|} = \left(\sum_{i=1}^{n_f} X_i^* \Big/ \sum_{i=1}^{n_f} P_i \right)$.
   2.6 **Update weight** of top-down weight matrix with PPI_I and node *k*.
   If $\delta > \rho$, update top-down weight of node *k*, $t_{ki}(new) = t_{ki} \times P_i$ where $i = 1, \cdots, n_f$.
   2.7 **Create a new node** in F_2 layer
   2.7.1 **Create** a new node *l*.
   2.7.2 **Initialize** top-down weight t_{li} to the current input feature pattern.
   2.7.3 **Initialize** button-up weight for the new node *l*.
   $$b_{il}(new) = \left(X_i^* \Big/ (0.5 + \sum_{i=1}^{n_f} X_i^*) \right), \text{ where } i = 1 \cdots n_f .$$