A New Approach for Modelling Gene Regulatory Networks Using Fuzzy Petri Nets

Raed I. Hamed[1*], S. I. Ahson[2], R. Parveen[1]

[1] Department of Computer Science, JMI University, New Delhi-110025, India

[2] Bioinformatics and Computational Biology, Patna University - 800005, India

Summary

Gene Regulatory Networks are models of genes and gene interactions at the expression level. The advent of microarray technology has challenged computer scientists to develop better algorithms for modeling the underlying regulatory relationship in between the genes. Fuzzy system has an ability to search microarray datasets for activator/repressor regulatory relationship. In this paper, we present a fuzzy reasoning model based on the Fuzzy Petri Net. The model considers the regulatory triplets by means of predicting changes in expression level of the target based on input expression level. This method eliminates possible false predictions from the classical fuzzy model thereby allowing a wider search space for inferring regulatory relationship. Through formalization of fuzzy reasoning, we propose an approach to construct a rule-based reasoning system. The experimental results show the proposed approach is feasible and acceptable to predict changes in expression level of the target gene.

1 Introduction

Gene regulatory networks control biological functions by regulating the level of gene expression. Discovering and understanding the complex causal relationships within gene regulatory networks has become a major issue in systems biology, computational biology and bioinformatics. The benefits of characterizing gene interaction are many; for example, the effects of drugs on a regulatory pathway can be found, the development of cancer in a cell can be tracked, etc. DNA microarray experiments today allow to monitor the output of gene regulatory networks by measuring the gene expression levels of thousands of genes [1]. Several methods have been proposed to develop maps of gene interaction, including Bayesian networks [2], dynamic Bayesian networks with hidden Markov model [3], and Boolean networks [4]. More recently, neural networks have also been applied to the problem of gene expression data analysis (eg. [5], [6]). Woolf and Wang [7] introduced a fuzzy logic approach to analyse the activator/repressor relationship of gene interaction using a normalized subset of *Saccharomyces cerevisiae* data [8].

They applied every possible combination of activators and repressors for each gene and the output from the model was compared to the expression levels of the remaining genes. Since gene expression levels are qualitatively classified into low, medium and high states to a varying degree based on a set of membership functions. Genes are then paired into an activator and repressor, and this gene pair determines the predicted target gene expression values based on a set of heuristic rules. Since a fuzzy logic algorithm [7] searches a microarray dataset for regulatory triplets consisting of activator, repressor and target gene.

* To whom correspondence should be addressed. Email: raed.inf@gmail.com

doi:10.2390/biecoll-jib-2010-113

The rule-based system has played an important role in such an expert system. Especially in an uncertain information environment, the rule-based system must have the capability of performing fuzzy reasoning which is based on the fuzzy sets foundation [9]. Several existing work puts emphasis on details of actual implementation of fuzzy system, and lacks formal specifications of fuzzy reasoning.

Modeling and simulation methods provided by systems theory can help improving the level of understanding of biological phenomena, [10]. In particular, Petri nets are becoming the reference modeling formalism for GRNs (see, e.g., [11, 12, 13, 14]): activation and inhibition of gene activity is intrinsically an on/off mechanism, and the dynamics governing proteins concentration are described by hybrid Petri nets (HPNs), while the activation and the deactivation of these dynamics are triggered by discrete switches encoding protein concentration reaching some threshold. Fuzzy Petri net (FPN), which combines fuzzy logic with Petri net, is useful tool in dealing with uncertain and incomplete information.

In this paper, we introduce and motivate a new modeling approach fuzzy Petri net which provides a powerful and intuitive tool for investigating biological processes and systems. We then apply this method to predict changes in expression level of the target gene. Since every triplet of genes (one as the activator, one as an repressor, and one as the target gene) is checked, in our model. With difference from those existing system, we formalize the fuzzy reasoning mechanism with Fuzzy Petri Net [15]. With the FPN's graphical nature and mathematical foundation, we visualize the structure of a rule-based fuzzy reasoning system, and make the model relatively simple and legible.

In an attempt to find new applications and stimulate new research topics, researchers such as [16] and [17] combined fuzzy theory and the basic Petri net to form a new model and define the associated operations of fuzzy Petri net in modeling biological processes. We present a more complete and efficient model for rule-based reasoning system modeled as FPN. The model is developed to simulate the inference process from the antecedent to the consequent propositions. Also, we modify fuzzy Petri nets to model gene regulatory network with fuzzy logic. Our ultimate goal is to develop a model that is similar to, but simpler than, classical fuzzy models (see [1, 7] for instance), and that has the power to perform predicate logic.

The paper is structured as follows. Section 2 introduces a gene regulatory network. Section 3 introduces the definition of Petri nets and fuzzy Petri nets. Section 4 describes the process of changes in expression values. Section 5 describes the FPN model based specifications of fuzzy reasoning rules and procedure of reasoning. Finally, Section 6 concludes the paper and points out directions for future work.

2 Gene regulatory network

Generally, a genetic network can be expressed by a set of nonlinear differential equations with each gene expression level as variables [18]. The accuracy of this design depends on the accuracy of the data in terms of concentrations, rate constants, and expression levels [19]. The expression level of gene *i* at time instant *t*+1 is given by

$$x_i(t+1) = f_i(x(t))$$

where $x_i(t+1)$ is the expression level (mRNA concentration) of gene *i* at time instant $t+1$, $x(t)$ is the vector of expression levels of all genes at time instant t, f_i is the function that determines the expression level of gene i from the previous expression values of all genes. Note that the function f_i is static, without altering during simulation. Although the vector that holds values of all genes is the parameter of the function f_i, it takes in practice only the genes that control the gene i. In most cases, the expression function f_i uses only a couple of genes' values. For instance, the expression function of the gene 3 in Fig. 1 is

$$f_3(x(t)) \equiv f_3(x_2(t), x_1(t))$$

as the gene 3 receives regulation from gene 1, and gene 2. However, this model indicates that the expression level of gene 3 is directly influenced be expression level of gene 1 and gene 2.

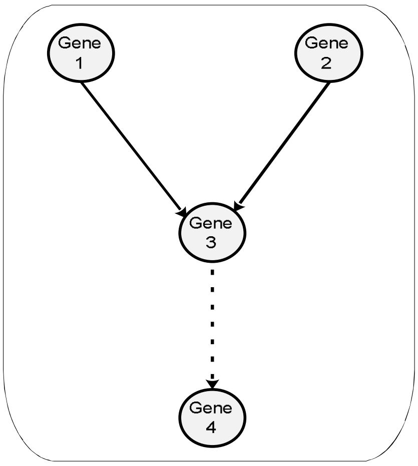

Fig. 1: Generalized GRN model

We can obtain mRNA concentrations of gene $x_i(t)$ at time t, but do not know the expression of $f_i(x(t))$,i.e., we do not know the interaction relation between gene $x_i(t)$. To study the genetic regulatory network, we should obtain the expression of $f_i(x(t))$, according to the microarray data. In fact, it is impossible to find the exact $f_i(x(t))$. However, many people have become aware that the real world is not linear quadratic and that many situations can not be modeled accurately by mathematically tractable equations [20, 21]. Combining with Petri net and knowledge representation, a Fuzzy Petri Nets can be used to depict fuzzy generating rules that can be taken as rules of fuzzy relationships between two propositions [22, 23]. So we will construct a fuzzy system according to the microarray data in different time points, and make the fuzzy system universal approximator of $f_i(x(t))$.

3 A Petri Nets and Fuzzy Petri Nets Review Stage

3.1 Petri Nets

A Petri net is a directed, weighted, bipartite graph consisting of two kinds of nodes, called places (P_i) and transitions (T_j), where arcs are either from a place to a transition or from a transition to a place [24]. Murata has formally defined Petri nets as a 5-tuple [25]: $PN = (P, T, F, W, M_0)$, where $P = \{P_1, P_2, \ldots, P_m\}$ is a finite set of places, $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs, $W : F \rightarrow \{1, 2, 3, \ldots,\}$ is a weight function, and $M_0 : P \rightarrow \{1, 2, 3, \ldots,\}$ is the initial marking. A marking M is an m-vector, $\{M(P_1),\ldots,M(P_m)\}$, where $M(P_i)$ denotes the number of the tokens in place P_i. The incidence matrix $A = [a_{ij}]$ is a $n \times m$ matrix of integers and its typical entry is defined by $a_{ij} = a_{ij}^+ - a_{ij}^-$, where a_{ij}^+ is the weight of the arc from a transition t_i to its output place P_j, and a_{ij}^- is the weight of the arc to a transition t_i from its input place P_j .The reachability set $R(M_0)$ of a Petri net is defined as the set of all possible markings reachable from M_0. Some notations are introduced as follows: $\bullet t_j$ denotes the input places of t_j, $t_j\bullet$ denotes the output places of t_j, $\bullet P_i$ denotes the input transitions of P_i, and $P_i\bullet$ denotes the output transitions of P_i. Because PNs (ordinary Petri nets)cannot deal with vague or fuzzy information such as “very good“ and “healthy” several kinds of Fuzzy Petri Nets (FPNs) have been introduced [15]. They are used for fuzzy knowledge representation and reasoning. A FPN differs from a PN only in markings, the firing rule, and possible-token locations. In this paper, we use the FPNs defined in [26].

3.2 Definition of Fuzzy Petri nets

FPN expanded from a Petri net is a bipartite graph that has place and transition nodes like the Petri net. However, in FPN a token incorporated with a place is associated with a real value between 0 and 1; a transition is associated with a certain factor (CF) between 0 and 1. Fuzzy Petri net is a promising modeling tool for expert systems, and it has shown itself to be suitable for fuzzy knowledge representation and reasoning. In order to capture more information of modeling gene regulatory networks, many authors have developed FPNs, for example, 8-tuple [15], 13-tuple [20], and 9-tuple [26]. As shown in [26], a fuzzy Petri net structure is defined as 9-tuple:

$FPN = (P, T, D, I, O, f, \alpha, \beta, \lambda)$ where:

$P = \{p_l, p_2, ..., p_n\}$ is a finite set of places, corresponding to the propositions of FPRs;

$T = \{t_l, t_2, ..., t_n\}$ is a finite set of transitions, $P \cap T = \varnothing$, corresponding to the execution of FPRs;

$D = \{d_1, d_2, ..., d_n\}$ is a finite set of propositions of FPRs. $P \cap T \cap D = \varnothing$, $|P| = |D|$, d_i (i= 1,2,..., n) denotes the proposition that interprets fuzzy linguistic variables, such as: “low” “medium” “high”, as in our model;

I : is an input incidence matrix;

O : is an output incidence matrix;

$f = \{\mu_1, \mu_2, ..., \mu_m\}$ where μ_i denotes the certainty factor (CF) of R_i , which indicates the reliability of the rule R_i , and $\mu_i \in [0,1]$;

$\alpha : P \rightarrow [0,1]$ is the function which assigns a token value between zero and one to each place;

$\beta : P \rightarrow D$ is an association function, a bijective mapping from places to propositions.

$\lambda : T \rightarrow [0, 1]$ is the function which assigns a threshold λ_i between zero and one to a transition t_i;

By carefully connecting related place and assigning reasonable values of certainty factors to transitions, we can come up with a fuzzy Petri net that can make decisions based on the expertise we gave it during its construction.

There are a significant number of generalizations of Petri nets into the domain of fuzzy sets; most of these approaches are focused on modeling the mechanisms of approximate reasoning. The primary thrust of these attempts is in a proper representation of the semantics of the underlying reasoning mechanisms. For the limitations, fuzzy Petri net may be not suitable to parallel reasoning, such as in [16].

doi:10.2390/biecoll-jib-2010-113

3.3 Definition of pre-set and post-set

1) pre-set: $\forall x \in T \cup P$, ${}^{\bullet}x = \{y \mid (y,x) \in F\}$ is called the pre-set of x ;

2) post-set: $\forall x \in T \cup P$, $x^{\bullet} = \{z \mid (x,z) \in F\}$ is called the post-set of x ;

3.4 Construction of Input Incidence Matrix I and Output Incidence Matrix O

Assuming that there are m places, n transitions in the fuzzy Petri net, we have

Input incidence matrix $I_{m \times n}$:

$$I = (a_{ij})_{m \times n}$$

$$(i = 1,2,\ldots,m;\ j = 1,2,\ldots,n)$$

where

$$a_{ij} = \begin{cases} 1 & p_i \in {}^{\bullet}t_j \ \&\ p_i \notin t_j^{\bullet} \\ 0 & \text{others} \end{cases}$$

a_{ij} = 1 means that there is an arc connecting place p_i to transition t_j;

a_{ij} = 0 means that there is not an arc connecting place p_i to transition t_j;

Output incidence matrix $O_{m \times n}$:

$$O = (b_{ij})_{m \times n}$$

$$(i = 1,2,\ldots,m;\ j = 1,2,\ldots,n)$$

where

$$b_{ij} = \begin{cases} 1 & p_i \in t_j^{\bullet} \ \&\ p_i \notin {}^{\bullet}t_j \\ 0 & \text{others} \end{cases}$$

b_{ij} = 1 represents that there is an arc connecting transition t_j to place p_i;

b_{ij} = 0 means that there is not an arc connecting transition t_j to place p_i;

3.5 The concept of inhibition arc

Pedrycz and Gomide [27] provides a novel scheme for machine learning using fuzzy Petri nets. Their formulation is based on the usual definition of t-and s-norms. A transition t_i fires if its degree of firing exceeds its threshold λi.

$$z = T_{j \neq i}^{n}[(\lambda_j \rightarrow \alpha(p_j))sw_j]$$

where, "→" denoted a fuzzy implication. An important knowledge representation addition pertains to inhibitory arcs (connections) occurring within the net [27]. The role of this arc is to model an inhibitory action coming from a certain input place. In the two-valued version of the net these places while carrying a nonzero number of tokens prevent the associated transitions from firing. In the framework of the fuzzy Petri net the inhibitory action is completed by considering a complement of the marking of the inhibitory place see Fig 2.

This complement $\overline{x_i}$, contributes to the following expression describing the level of firing z.

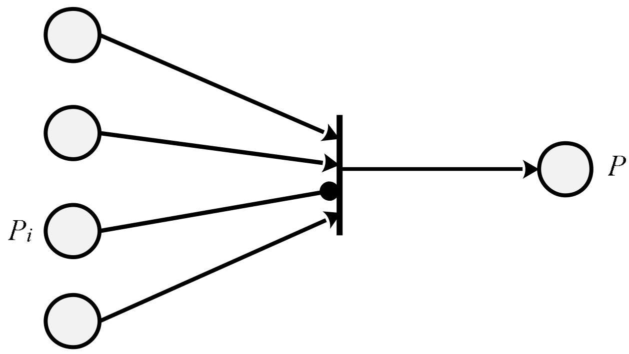

Fig. 2: Illustrating inhibition in a fuzzy Petri net

$$z = T^n_{j \neq i}\{[(\lambda_j \to \alpha(p_j))sw_j]\}t[(\lambda_j \to \overline{\alpha(p_i)})sw_i] = z^+ t z^-$$

where

$$z^- = [(\lambda_i \to \overline{\alpha(p_i)})sw_i] \qquad \text{(Inhibitory component)}$$

and

$$z^+ = T^n_{j \neq i}[(\lambda_j \to \alpha(p_j))sw_j] \qquad \text{(Excitatory component)}$$

Observe that for $w_i = 0$ and $\lambda_i = 1$, the inhibitory effect z^- equals directly $\overline{\alpha(p_i)}$ hence $\alpha(p_i) = 1$ completely prohibits the transition from firing. For lower level of marking $\alpha(p_i)$ this prohibition effect becomes limited. So to understand the inhibitory mechanism, and make that possible to implement our model to predict changes in target expression level, we set $w_i = 0$ and $\lambda_i = 0$.

3.6 Firing Principles of Transitions

A transition can be fired under the condition that the degrees of the truth of all its input places are not null and greater than certain threshold values. We follow the common firing principle in [28]. The degree of truth of an output place is equal to the minimum of the degrees of the input places multiplying the certainty factor (CF) of the transition. Once transition t_j meets its firing conditions, the degrees of truth of the places under the state marking $M_{(k)}$ are computed by:

$$M_{(k)}(p_i) = \begin{cases} \text{Min}\,\{ M_{(k)}({}^{\bullet}t_j) \} \times u_j & p_i \in t_j^{\bullet} \;\&\; p_i \notin {}^{\bullet}t_j \\ \\ M_{(k)}(p_i) & \text{others} \end{cases}$$

where

$t_j \in T,\ j = 1,2,...,n$;

$p_i \in P,\ i = 1,2,..., m$;

$M_{(k)}(p_i)$ denoted the degree of truth of the p_i under the state marking $M_{(k)}$;

k denoted the times of iteration;

u_j denoted the certainty factor (CF) of the *jth* rule.

Firing fuzzy production rules can be considered as firing transitions.

3.7 Rule representation for fuzzy reasoning

Production rules (PRs) are suitable to express expert knowledge. In most cases, collecting data in a precise way is difficult; FPRs are thus adopted, which have the ability of process uncertain or incomplete knowledge [16, 26, 29]. For these reasons, inference rules are obtained in the form of FPRs, enhancing reasoning capacity. FPNs are built on the basis of FPRs.

3.7.1 Fuzzy Production Rules (FPRs)

Let R be a set of fuzzy production rules:

$R = \{R_1, R_2,..., R_m\}$, and a fuzzy production rule R_i is shown as follows [30]:

R_i : If d_j then d_k , $(CF = \mu_i)$

IF *all propositions in the antecedent dj have value true* ***THEN*** *the propositions in the consequent d_k are true.*

where

$d_i = \{ d_{j1}, d_{j2},..., d_{j\,n}\}$, represents the antecedent part which comprises of one or more propositions connected by either “AND” or “OR” in the rule;

$d_k = \{ d_{k1}, d_{k2},..., d_{kn} \}$ represents the consequent part which comprises of one or more propositions connected by “AND” in the rule;

μ_i denotes the certainty factor (CF_i) of the rule R_i . Generally, FPRs are classified into four types as follows:

Type 1: If d_j, then d_k, (CF = μ),

Type 2: IF d_{jl} and d_{j2} and … and d_{jn} THEN d_k (CF = μ),

Type 3: IF d_{j1} or d_{j2} or… or d_{jn}, THEN d_k (CF = μ),

Type 4: IF d_j THEN d_{k1} and d_{k2} and … and d_{kn} (CF = μ),

doi:10.2390/biecoll-jib-2010-113

FPN models of the 4 types of composite *fuzzy* production rules are shown in Fig. 3**.** Places (drawn as circles) represent entities or concentrations of a protein, mRNA, complex of proteins, metabolites, etc. Transitions (drawn as bars) represent biological processes like enzymatic reactions, transport, etc. Arcs represent dependencies of places and transitions or define which and how places are affected whenever a transition fires.

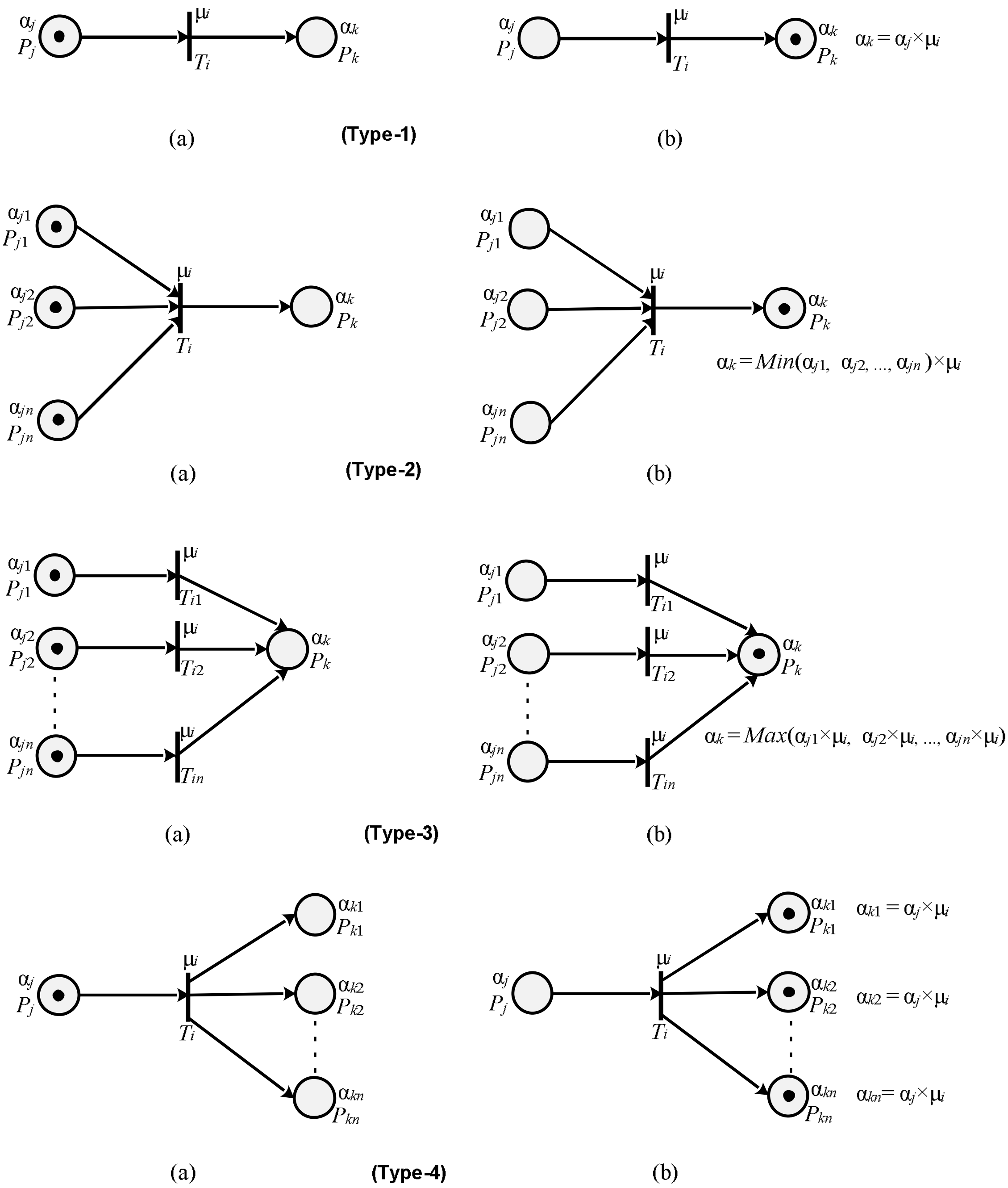

Fig. 3: Fuzzy Petri net models of composite fuzzy production rules. Firing of Transitions in FPN. (a) Before firing transitions. (b) After firing transitions.

4 Process of changes in expression values

The fuzzy predict changes in expression values of the target gene are based on fuzzy logic control theory. It consists of the following four steps:

4.1 Defining the membership functions for the input and output

Fuzzification consists of defining the membership function for the input and output as well as mapping from crisp data to fuzzy membership. In the process of changes in expression values we search a microarray dataset for regulatory triplets consisting of activator, repressor and target gene [7, 31]. Gene expression levels are first qualitatively classified into low, medium, and high states to a varying degree based on a set of fuzzy membership functions. Genes are then paired into an activator and repressor, and this gene pair determines the predicted target gene expression profile based on a set of heuristic rules. In this process we set the activator and repressor as input, with changes in target gene expression level as output. The fuzzy membership functions of activator and repressor are described in Fig.4 .In order to measure these input and output metadata universally, we normalize them into the same standard scale of [0, 1]. The activator and repressor gene are classified into three sets, respectively. The value of input data may belong to 1, 2 or 3 sets with corresponding membership degree. For example,

$\mu_{activator\_expression=low}$ (0.375) = 0.25, $\mu_{activator\_expression=medium}$ (0.375) = 0.75, means the activator expression, 0.375 belongs to medium with confidence value (truth degree) of 75% while 25% belongs to low.

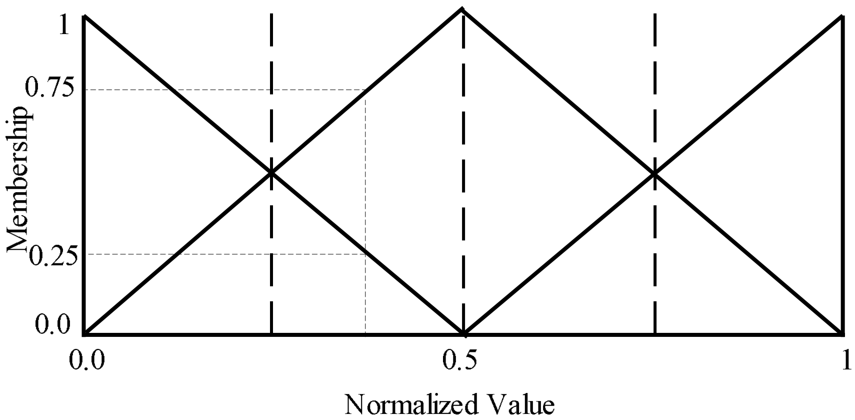

Fig. 4: The fuzzy input membership functions as a function of a normalized gene expression level.

The fuzzy membership function of the output, i.e. changes in target gene expression level is defined in Fig.5. It is represented with five levels or five sets with respect to fuzzy theory, namely Very low, Low, Medium, High, and Very high.

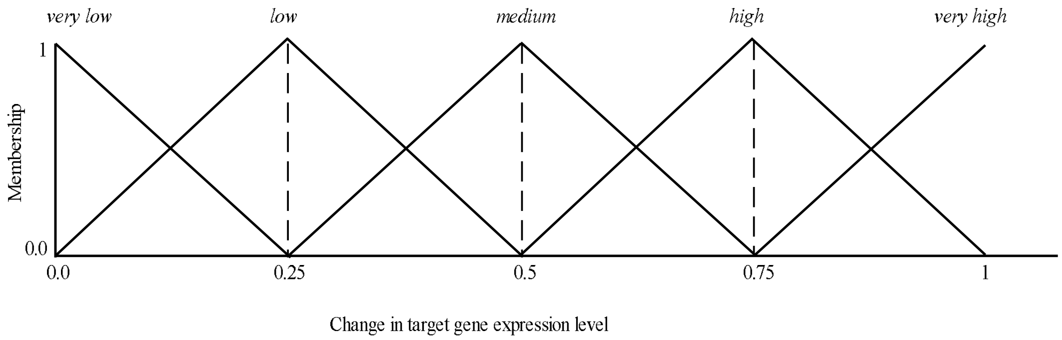

Fig. 5: The fuzzy membership function of output.

4.2 Fuzzification

Fuzzification consists of defining the membership function for the input and output as well as mapping from crisp values to fuzzy membership. Mapping a particular activator and repressor expression to the fuzzy membership correspondingly. By using the membership functions defined above, we translate the input crisp values of activator and repressor expression into a set of linguistic values and assign a membership degree for each linguistic value.

4.3 Reasoning with fuzzy reasoning rules

In our algorithm, input genes are drivers. The heuristic rules are constructed accordingly based on an activator/repressor regulatory logic. For example, if the activator and the repressor genes are qualitatively classified low, then the predicted change in the target expression level is considered Medium. In another case, if the activator is high and repressor is low, then the predicted change at target is a High equivalent to the activation input as the repressor is below threshold expression level. Similar heuristics are applied to construct the decision matrix shown in Fig. 6.

		Repressor		
		Low	**Medium**	**High**
Activator	Low	**Target Medium**	**Target Low**	**Target Low**
	Medium	**Target High**	**Target Medium**	**Target Low**
	High	**Target High**	**Target High**	**Target Medium**

Fig. 6: Fuzzy decision matrix for predicting change in expression level of the target gene in an activator/ repressor regulatory relationship.

The reasoning engine performs decision-making based on the fuzzy logic reasoning rules with first order predicate logic. Each rule can be defined as an If-Then clause, which determines the linguistic value of output according to the linguistic values of input. Those fuzzy reasoning rules are shown in Fig. 7.

1. If activator is "Low" and repressor is "Low" then target shift is "Medium"
2. If activator is "Low" and repressor is "Medium" then target shift is "Low";
3. If activator is "Low" and repressor is "High" then target shift is "Low"
4. If activator is "Medium" and repressor is "Low" then target shift is "High";
5. If activator is "Medium" and repressor is "Medium" then target shift is "Medium"
6. If activator is "Medium" and repressor is "High" then target shift is "Low";
7. If activator is "High" and repressor is "Low" then target shift is "High"
8. If activator is "High" and repressor is "Medium" then target shift is "High";
9. If activator is "High" and repressor is "High" then target shift is "Medium"

Fig. 7: Rules for reasoning

4.4 Defuzzification

In order to represent the global output variable in fuzzy Petri nets (for example, the change in target gene expression level has high, medium, or low) a defuzzification method is required to produce a non-fuzzy output (crisp value). In Fig. 8, firing transition T_1, T_2,…,T_9 deposits token in places P_7, P_8, and P_9 which can be defined as a defuzzification token and can be expressed mathematically by the "*centre of gravity*" method [32]. We adopt the "*center of gravity*" method as the defuzzification of the output predict change in target gene expression level

$$Change\_\ target\_\ expression\_level = \frac{\sum_{i=1}^{n} \mu[i] \times y_i}{\sum_{i=1}^{n} \mu[i]}$$

where:

• $\mu[i]$ is the height of output area from the i-th rule,

• y_i is the gravity's horizontal coordinate of output area from the i-th rule,

• n is the total number of matching rules for given values of each input dimension.

After get the crisp value of the output, we map it into its fuzzy membership and get the linguistic value whose membership degree has the highest level for change in target gene expression level.

5 FPN model to predict changes in target expression values

In this paper, we introduce a novel fuzzy Petri net model to predict changes in expression values and infer causal relationship between genes.

Suppose the available activator expression value is 0.78, and the repressor expression value is 0.3. For these values we will map the normalized expression values to the fuzzy input membership function defined in Fig. 4, the activator expression value, 0.78 is between "*Medium*" and "*High*" $\mu_{Activator\_expression=medium}$ (0.78)=0.44, $\mu_{Activator\_expression=high}$ (0.78)=0.56. The repressor expression value, 0.3 is between "*Low*" and "*Medium*", $\mu_{Repressor\_expression=low}$ (0.3) =0.38, $\mu_{Repressor\_expression=medium}$ (0.3) =0.62.

5.1 Constructing Petri Net model

Fig. 8 shows how we realize the steps of fuzzy inference using the FPN structure as $FPN = (P, T, D, I, O, f, \alpha, \beta, \lambda)$, where

$P= \{p_1, p_2 , p_3 , p_4 , p_5 , p_6 , p_7 , p_8 , p_9\}$,

$T_= \{ t_1 , t_2 , t_3 , t_4 , t_5 , t_6 , t_7 , t_8 , t_9 \}$,

$$I = \begin{Bmatrix} 1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1 \\ 1\,0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \end{Bmatrix} \quad O = \begin{Bmatrix} 0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,1\,0\,0\,0 \\ 1\,0\,0\,0\,1\,0\,0\,0\,1 \\ 0\,0\,0\,1\,0\,0\,1\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0 \end{Bmatrix}$$

The linguistic meaning for propositions together with certainty factor (i.e. CF=μ_i) are listed in the table 1. Because we used our algorithm to search for Activator_Repressor_Target triplets, we expected to find the confidence degree of each initial rule usually depends on those experiences of experts.

Therefore, these initial confidence degrees have been defined before the reasoning begins. Besides, the linguistic meaning of each proposition may have an effect on the confidence degree.

We set the confidence degree of each rule as shown in Fig. 8. Therefore, the confidence degree vector CF=μ_i is

μ_i = {0.8, 0.6, 0.8, 0.7, 0.9, 0.95, 0.8, 0.99, 0.9}.

As we computed the initial truth degree for each place (i.e. $\alpha(p_i) = y_i$), the truth degree vector is

$\alpha = \{0, 0.44, 0.56, 0.38, 0.62, 0, 0, 0, 0\}^T$.

The initial marking vector is

$M_0 =\{1,1,1,1,1,1,0,0,0\}^T$.

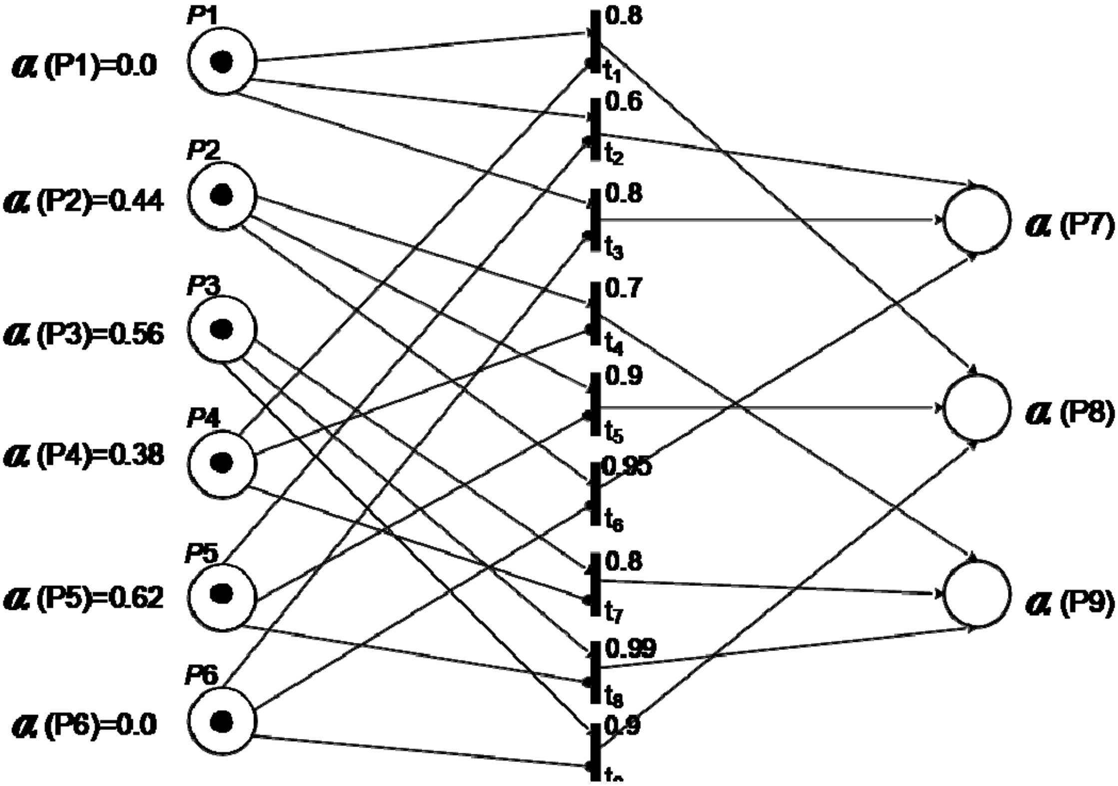

Fig. 8: A FPN model

Tab. 1: propositions and certainty factor for FPN Model

Propositions	Places	Initial truth degree	Initial marking	Certainty factor
Activator _Low	P_1	0	1	0.8
Activator_Medium	P_2	0.44	1	0.6
Activator _High	P_3	0.56	1	0.8
Repressor_Low	P_4	0.38	1	0.7
Repressor_Medium	P_5	0.62	1	0.9
Repressor_High	P_6	0	1	0.95
Target_Low	P_7	0	0	0.8
Target_Medium	P_8	0	0	0.99
Target_High	P_9	0	0	0.9

5.2 FPN reasoning

The execution of FPN will change the truth degree vector (i.e. $\alpha(p_i) = y_i$) and the marking vector M_0. The procedure of execution of FPN is shown as follows.

$\alpha(p_i)$ = {0, 0.44, 0.56, 0.38, 0.62, 0, 0, 0.396, 0.392}^{T}.

$M_1 = \{0,0,0,0,0,0,1,1,1\}^T$.

After the execution mentioned above, the change in target is got. Transfer the linguistic values into a crisp value according to "center of gravity", see section 4.

$$Change\_Target\_Expression\_Level = \frac{0 \times low + 0.396 \times medium + 0.392 \times high}{0 + 0.396 + 0.392}$$

$$Change\_Target\_Expression\_Level = \frac{0 \times 0.25 + 0.396 \times 0.5 + 0.392 \times 0.75}{0 + 0.396 + 0.392} \approx 0.62$$

According to Fig. 5, the change target expression level, 0.62 is between "*Medium*" and "*High*". Apparently, the change in target gene expression level belongs to "*Medium*", because $\mu_{Change\_target\_expression\_level = medium}$ is much larger than $\mu_{Change\_target\_expression\_level = high}$. Hence, we decide the final change in target expression level as "*Medium*".

6 Conclusions

In this work, a fuzzy Petri nets GRN model is proposed for searching activator/ repressor regulatory relationship between gene triplets in the microarray data. The model predicts changes in expression levels in the target gene caused due to possible regulation based on input expression levels. The genes that fit the model are more likely to exhibit activator/ repressor relationship. We propose a novel approach of fuzzy reasoning to predict changes in expression levels. The major features of our approach are: (1) Since the FPN is a graphical tool, we are able to give a description of the typical procedure of fuzzy reasoning; (2) visualize the structure of a rule-based fuzzy reasoning system; (3) with the mathematical foundation of FPN, we construct the reasoning steps for FPN reasoning; and (4) Finally, we describe the FPN based modeling to predict changes in expression levels in the target gene to validates the feasibility of FPN model. With the definition of FPN and the procedure of FPN reasoning, some flaws of FPN based reasoning should be pointed out. In a FPN model, both the truth degree of a proposition and the confidence degree of a rule should be determined beforehand. The determination of these two degrees usually relies on experiences of experts, which induce some uncertainty in the reasoning. Further biological experiments are needed to determine the validity of the genetic interactions suggested by the model.

For future work, we plan to integrate the proposed approach with neural network for modeling gene regulatory network.

Acknowledgements

The authors would like to thank the reviewers for their best comments that help improve the manuscript. Many thanks for my professor Syed I. Ahson for improving this paper. And I also give my thanks to my colleagues for their good advices to this paper. The work is partially supported by the national nature foundation under Contact Number 5576.

References

[1] R. Ram, M. Chetty and T. I. Dix. Fuzzy Model for Gene Regulatory Network. *IEEE Congress on Computational Intelligence*, WCCI 06, Vancouver Canada, 5599-5604, 2006.
[2] P. Spirtes, C. Glymour, R. Scheines, S. Kauffmann, V. Aimale and F. Wimberly. Constructing Bayesian Network Models of Gene Expression Network from Microarray Data. *Proc. Atlantic Symp. Computational Biology, Genome Information Systems and Technology*, 2000.
[3] E. Perrin, L. Rolaivola, A. Mazurie, S. Bottani, J. Mallet and F. D'Alche-Buc. Gene Network Inference using Dynamic Bayesian Networks. *Bioinformatics*, 19(2):138-148, 2003.
[4] S. Laing, S. Fuhrman and R. Somogyi. REVEAL, A general reverse engineering algorithm for inference of genetic network architectures. *Proc. Pacific Symp. Biocomputing*, 3:18-29, 1998
[5] A. Narayanan, E. C. Keedwell, J. Gamalielsson and S. Tataneni. Single Layer Artificial neural network for gene expression analysis. *Proc. Neuro computing Conf.*, 61: 217-240, 2004.
[6] E. Keedwell and A. Narayanan. Discovering Gene Networks with a Neural - Genetic Hybrid. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2 (3): 231-242, 2005.
[7] P. J. Woolf and Y. Wang. A fuzzy logic approach to analyzing gene expression data. *Physiological Genomics*, 3: 9–15, 2000.
[8] R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrieleian, D. Landsman, D. J. ockhart and R. W. Davis. A genome wide transcriptional analysis of the mitotic cell. *Mol. Cell*, 2: 65-73, 1998.
[9] L. Zadeh. Fuzzy Sets as Basis for a Theory of Possibility. *Fuzzy Sets and Systems*, 1:3-28, 1978.
[10] E. D. Sontag. Molecular systems biology and control. *European Journal of Control*, 11(4-5): 396-435. 2005.
[11] H. Matsuno, A. Doi, M. Nagasaki and S. Miyano. Hybrid Petri net representation of gene regulatory network. *Pacific Symposium on Biocomputing*, (in press), 2000.
[12] H. Matsuno, S. Fujita, A. Doi, M. Nagasaki and S. Miyano. Towards Biopathway Modeling and Simulation. *In Proceedings of ICATPN*, 3-22, 2003.
[13] S. Fujita, M. Matsui, H. Matsuno and S. Miyano. Modeling and simulation of fission yeast cell cycle on hybrid functional Petri net. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E87-A(11): 2919-2928, 2004.
[14] H. Matsuno, T. Inouye, Y. Okitsu, Y. Fujita and S. Miyano. A new regulatory interactions suggested by simulations for circadian genetic control mechanism in mammals, Proc. *The 3rd Asia-Pacific Bioinformatics Conference*, Imperial College Press, 171-180, 2005.
[15] S. M. Chen, J. S. Ke and J. F. Chang. Knowledge representation using fuzzy Petri nets. *IEEE Transactions on Knowledge Data Engineering*, 2: 311-319, 1990.
[16] E. Tazaki and Y. Katsumi. A fuzzy Petri net model for approximate reasoning and its application to medical diagnosis. *In Proc. of 1992 IEEE int. Con. On Systems, Man, and Cybernetics*, 1:627-631, 1992.
[17] W. Lukas and Z. Ralf. Intuitive Modeling of Dynamic Systems with Petri Nets and Fuzzy Logic. *German Conference on Bioinformatics*, (136): 106-115, 2008.
[18] J. Shen, Y, Wang and Z. Liu. Designing Genetic Regulatory Network with Fuzzy Logic. *Springer-Verlag Berlin Heidelberg*, LNAI 5227: 990–995, 2008.

[19] T. Maeshiro, S. Nakayama, H. Hemmi and K. Shimohara. An evolutionary system for the prediction of gene regulatory networks in biological cells. *SICE, 2007 Annual Conference*, 17-20: 1577 – 1581, 2007.
[20] H. Ressom, D. Wang, R. S. Varghese, R. Reynolds. Fuzzy Logic based genetic regulatory network. *IEEE International Conference on Fuzzy Systems*, 1210-1215, 2003.
[21] R. R. Chines and M. L. Shaw. From fuzzy logic to expert systems. *Itdorm Sci*, 36: 5-15, 1985.
[22] M. L. Garg, S. I. Ahson and P. V. Gupta. A fuzzy Petri net for knowledge representation and reasoning. *Information Processing Letters*, 39: 165-171, 1991.
[23] S. I. Ahson. Petri net models of fuzzy neural networks. *IEEE. Trans. On. SMC*, 25(6): 926-932. 1995.
[24] J. L. Peterson. Petri Net Theory and the Modeling of Systems. Englewood Cliffs, NJ: Prentice-Hall, 1981.
[25] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE* 77(4): 541-580, 1989.
[26] J. Yuan, H. Shi, C. Liu and W. Shang. Improved basic inference models of fuzzy Petri nets. *IEEE Transactions on Intelligent Control and Automation*, 1488-1493, 2008.
[27] W. Pedrycz and F. Gomide. A generalized fuzzy Petri net model. *IEEE Transsactions on Fuzzy Systems*, 2(4): 295-301, 1994.
[28] H. Chun, L. Ping and W. Hui. Improved Modeling Algorithm of Fuzzy Petri Net for Fuzzy Reasoning. *IEEE international conference on systems, Man and Cybernetics* 5: 4992-4997, 2003.
[29] M. Gao and M. Zhou. Fuzzy reasoning Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics-Part A:* Systems and Humans, 33(3): 314-324. 2003.
[30] J. Sun, S. Qin and Y. Song. Fault diagnosis of electric power systems based on fuzzy Petri nets. *IEEE Transactions on Power Systems*, 19(4): 2053-2059, 2004.
[31] D. S. Yeung and E. C. Tsang. A multi-level weighted fuzzy reasoning algorithm for expert systems. *IEEE Trans Syst Man Cybern* 28(2): 149-158, 1998.
[32] Y. Jian, L. Jintao, S. Hongzhou, G. Xiaoguang and Z. Zhenmin. A Fuzzy Petri Net Model towards Context-Awareness Based Personalized Recommendation. *IEEE Transactions*, FSKD (3): 325-330, 2008.

doi:10.2390/biecoll-jib-2010-113