Modeling Gene Networks using Fuzzy Logic

Artur Gintrowski

Silesian University of Technology, Institute of Electronics,
Division of Microelectronics and Biotechnology
16 Akademicka Street, 44-100 Gliwice, Poland
artur.gintrowski@gmail.com

Abstract

Recently, almost uncontrolled technological progress allows so called high-throughput data collection for sophisticated and complex experimental biological systems analysis. Especially, it concerns the whole cellular genome. Therefore it becomes more and more vital to suggest and elaborate gene network models, which can be used for more complete interpretation of large and complex data sets. The presented paper concerns modeling of interactions in yeast genome. With the reference to previously published papers concerning the same subject, our paper presents a significant improvement in calculation procedure leading to very effective reduction of time of calculation.

1998 ACM Subject Classification I.5.1 Models, I.5.4 Applications

Keywords and phrases Fuzzy network, gene expression, time optimization

Digital Object Identifier 10.4230/OASIcs.MEMICS.2010.32

1 Introduction

Immediate technological evolution allows the analysis of more and more composite biological systems. The creation of elaborate gene network models involves widely developed analyses, which allows the better utilization of biological interpretation of medical data packages, particularly data concerning gene expression measurements. An example of a very effective modeling of gene network was previously presented by Sokhansanj et al. in BMC Bioinformatics [1]. This algorithm, which takes advantage of the theory of fuzzy sets, allows the creation of a model of intergenetic interactions. Input data for the described fuzzy system are gene expression measurements obtained as a result of a biological experiment using GeneChip microarray technology.

The huge advantage of the described method is that it receives the exact model of interactions in the result of the analysis. However, the time of account is its main defect. In the article, we present a detailed description of the algorithm with modifications on the significant correction of the speed of calculation (over 95% time reduction) to obtain almost identical results in comparison with the original exhaustive algorithm.

2 Original Algorithm Idea

In this study, measurement data of gene expression, obtained during the whole cellular cycle, are used. Results are collected in the I matrix, in which the following rows expressions

Sixth Doctoral Workshop on Math. and Eng. Methods in Computer Science (MEMICS'10)—Selected Papers.
Editors: L. Matyska, M. Kozubek, T. Vojnar, P. Zemčík, D. Antoš; pp. 32–39
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of individual G_i genes are included:

$$I = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,N} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,N} \\ \vdots & & \ddots & \\ e_{M,1} & e_{M,2} & \cdots & e_{M,N} \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_M \end{bmatrix} \tag{1}$$

$$G_i = [e_1, e_2, \ldots, e_N] \tag{2}$$

2.1 Data Preparation

Raw data are processed non-linearly at the beginning according to (Eq. 3). This transformation conducts the standardization of the input data in the interval $< -1; 1 >$, as it is possible that data are collected from different microarray experiments.

$$\widehat{I} = \frac{\arctan(I)}{\frac{\pi}{2}} \tag{3}$$

Data prepared this way are entered into the fuzzy system, which initially affects their fuzzification for sets with low, medium, and high expression respectively (Eq. 4,5 and 6). In the destination of further calculations, fuzzy data arrays F_L, F_M and F_H are concatenated in the third dimension of the F matrix (Eq. 7).

$$F_{L_{i,j}} = \begin{cases} -\widehat{e}_{i,j} & \widehat{e}_{i,j} < 0 \\ 0 & \widehat{e}_{i,j} > 0 \end{cases} \tag{4}$$

$$F_{M_{i,j}} = 1 - |\widehat{e}_{i,j}| \tag{5}$$

$$F_{H_{i,j}} = \begin{cases} 0 & \widehat{e}_{i,j} < 0 \\ \widehat{e}_{i,j} & \widehat{e}_{i,j} > 0 \end{cases} \tag{6}$$

$$F = \{F_L, F_M, F_H\} \tag{7}$$

2.2 Fuzzy Rules

The database in the described fuzzy system consists of three basic fuzzy rules in the following form:

- if *input* is ***LOW*** then *output* is ***ExpressionLevel***
- if *input* is ***MEDIUM*** then *output* is ***ExpressionLevel***
- if *input* is ***HIGH*** then *output* is ***ExpressionLevel***

where: $ExpressionLevel \in \{LOW,\ MEDIUM,\ HIGH\}$.

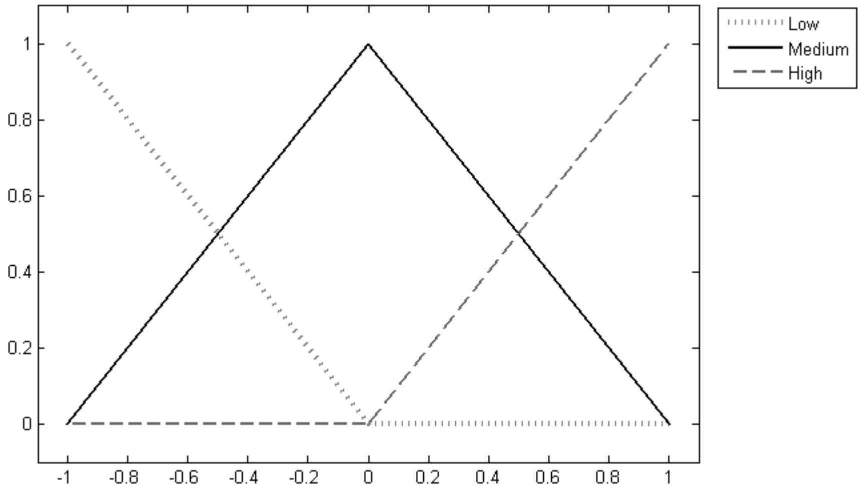

Figure 1 Linear functions used to fuzzify gene expression data

With a view to enable calculations, a relevant notation of the record is provided in the form of the r vector:

$$r = [l_1, l_2, l_3] \tag{8}$$

where l_1 is the output expression if input is *LOW*, l_2 is the output expression if input is *MEDIUM*, l_3 is the output expression if input is *HIGH* and $l_i \in \{1, 2, 3\}$ (output expression is 1 – low, 2 – medium or 3 – high).

Example: The following exemplary rule database

- if *input* is *LOW* then *output* is *HIGH*
- if *input* is *MEDIUM* then *output* is *MEDIUM*
- if *input* is *HIGH* then *output* is *LOW*

can be described using the following vector:

$$r_{example} = [3, 2, 1]$$

2.3 Iteration Issue

The fuzzy system is used repeatedly to model the initial vector of the expression, based on the chosen input genes and the chosen linguistic fuzzy rules.

For every choice of initial genes and combination of linguistic rules (combination of vectors r), the result of a vector being compared with the original vector of the expression of the initial gene is obtained. The optimum combination of input genes and linguistic rules is chosen using a certain rate of error described hereinafter. This guarantees gaining the highest resemblance

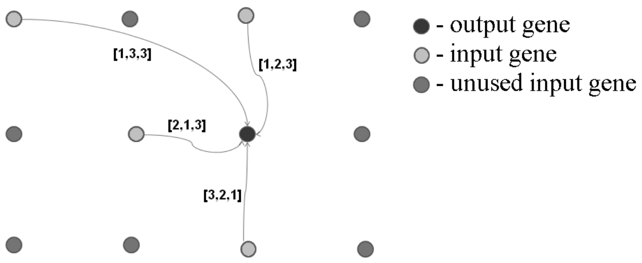

Figure 2 Example of the connections to an output gene in the fragment of the network modeled using four input genes per output

between the vector obtained as a result of the modeling and the original vector of the initial gene created.

Due to the huge amount of iterations, limiting the number of genes considered in the analysis to be initial genes is intentional. An optimum number of four initial genes is suggested.

Determining the huge number of iterations discarded is worthwhile. In the analysis of the discussed microarray of the 12 genes of yeast, conducting the analysis of the effect of the 11 remaining genes is necessary for each of them. To establish the recommended number of four entries, conducting calculations for the following number of the combinations of input genes is necessary:

- C_{11}^1 combinations for tests of influence of one gene on the network output
- C_{11}^2 combinations for two inputs
- C_{11}^3 combinations for three inputs
- C_{11}^4 combinations for four inputs.

These values are calculated in (Tab. 1). In the case of testing the influence of one gene, for each output the test of the 27 combinations of linguistic rules is necessary. For two input genes, there are $27^2 = 729$ combinations of rules; and for three input genes, there are $27^3 = 19683$ combinations. The case with four input genes gives $27^4 = 531441$ combinations. The total number of iterations included in the elaborated software during the single yeast microarray analysis gives

$$L_{oryg} = 12 \cdot C_{11}^1 \cdot 27 + C_{11}^2 \cdot 27^2 + C_{11}^3 \cdot 27^3 + C_{11}^4 \cdot 27^4 = 2143963404.$$

They require compound accounts from the disposed implementation programme on time optimization executable for over two billion cases of the described procedures. The final modification of the conclusion process contributes to the progress through the introduction of additional ratios of error in the analysis of the combinations of input genes. This most significantly limits the number of selected fuzzy rules.

2.4 Inference Engine

The described fuzzy system concludes each iteration of the algorithm, that is, for each combination of the fuzzy input genes F_C (see Eq. 7) as well as the combination of linguistic

Table 1 Number of analyzed combinations of rules for each combination of input genes

$C_{11}^{1} = \binom{11}{1} = \frac{11!}{(11-1)!\cdot 1!} = \frac{11!}{10!} = 11$
$C_{11}^{2} = \binom{11}{2} = \frac{11!}{(11-2)!\cdot 2!} = \frac{11!}{9!\cdot 2!} = 55$
$C_{11}^{3} = \binom{11}{3} = \frac{11!}{(11-3)!\cdot 3!} = \frac{11!}{8!\cdot 3!} = 165$
$C_{11}^{4} = \binom{11}{4} = \frac{11!}{(11-4)!\cdot 4!} = \frac{11!}{7!\cdot 4!} = 330$

rules included in the matrix R_C (Eq. 8).

$$R_C = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \tag{9}$$

$$\widetilde{F_{i,j,k}} = F_{C_{i,j},R_{C_{i,k}}} \tag{10}$$

$$D_{1,j,k} = \sum_{i=1}^{n} \widetilde{F_{i,j,k}} \tag{11}$$

$$D = \{D_{low}, D_{medium}, D_{high}\} \tag{12}$$

As a result, the fuzzy output set D is created.

2.5 Dealing with the Fuzzified Output

To obtain ultimate modeling results, the fuzzy result of inference is transmitted for defuzzification according to equations (Eq. 13 and 15). Equation (Eq. 14) is the graphic interpretation of the conclusion mechanism presented in (Fig. 3).

$$\tilde{O}_{1,i} = \frac{D_{1,i,3} - D_{1,i,1}}{D_{1,i,1} + D_{1,i,2} + D_{1,i,3}} \tag{13}$$

$$\tilde{O} = \frac{D_{high} - D_{low}}{D_{low} + D_{medium} + D_{high}} \tag{14}$$

$$O = \tan(\tilde{O} \cdot \frac{\pi}{2}) \tag{15}$$

The expression model O obtained through the presented algorithm on the output gene is compared with the original expression vector of the output gene G_O according to the following formula:

$$E = \frac{\sum_{i=1}^{N} (G_{O_i} - O_i)^2}{\sum_{i=1}^{N} (G_{O_i} - \overline{G_O})^2} \tag{16}$$

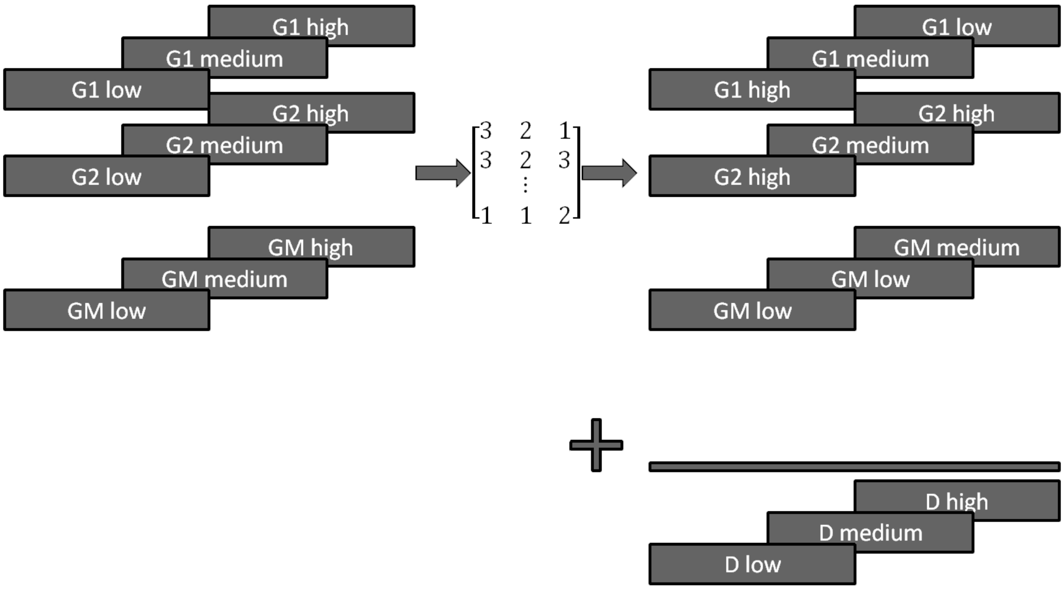

Figure 3 Inference scheme using the exemplary rule matrix. Inputs – genes $G_1, G_2, \ldots, G_M$. Fuzzified output – D matrix.

where G_{O_i} is the i–th expression measurement of the original output gene, O_i is the i–th expression vector measurement obtained as a modeling result and $\overline{G_O}$ represents the mean of the expression vector obtained as a result of modeling. In the comparison between the obtained identical model and the real expression vector of the output gene, the error coefficient E takes a value of zero. Hence, the better the choice of input genes and their respective linguistic rules, the lower the value of error coefficient is.

3 Time Optimization

As shown in (Sec. 2.3), computational complexity of the algorithm results mainly from the necessity for the continuous repetition of the fuzzy conclusion procedure for the huge number of combinations of the applied input data. To reduce that amount in the conclusion mechanism, several modifications are applied using three additional error coefficients constructed analogously to the main error coefficient (Eq. 16). However, they work inside the fuzzy system and on the fuzzified data of input genes, as well as in the fuzzy interpretation of the original output gene.

$$E_L = \frac{\sum_{i=1}^{N}(F_{LO_i} - D_{low})^2}{\sum_{i=1}^{N}(F_{LO_i} - \overline{F_{LO}})^2} \tag{17}$$

$$E_M = \frac{\sum_{i=1}^{N}(F_{MO_i} - D_{medium})^2}{\sum_{i=1}^{N}(F_{MO_i} - \overline{F_{MO}})^2} \tag{18}$$

$$E_H = \frac{\sum_{i=1}^{N}(F_{HO_i} - D_{high})^2}{\sum_{i=1}^{N}(F_{HO_i} - \overline{F_{HO}})^2} \tag{19}$$

The modified algorithm has four steps. In the first three, the k best linguistic rules with respect to the smallest coefficients E_L, E_M and E_H are stored in particular fuzzy sets. Next, in the fourth step, the analysis of the original algorithm is subsequently performed, taking only k^3 of all the best combinations of the linguistic rule vectors stored in the first three steps.

For comparison, the number of fuzzy conclusion procedure calls in case of the optimized algorithm for the described yeast microarray is equal to

$$L_{opt} = 12 \cdot C_{11}^1 \cdot (3 \cdot 3 + k^3) + C_{11}^2 \cdot (3 \cdot 3^2 + k^3) + C_{11}^3 \cdot (3 \cdot 3^3 + k^3) + C_{11}^4 \cdot (3 \cdot 3^4 + k^3).$$

The number for $k = 25$ takes the value of

$$L_{opt}\Big|_{k=25} = 106329168.$$

Therefore, the reduction of the fuzzy conclusion mechanism calls is:

$$(1 - \frac{L_{opt}}{L_{oryg}}) \cdot 100\% = 95.04\%$$

The same percentage of time reduction is also observed.

4 Results

The introduced modifications allow for calculations in a much shorter time. Table (Tab. 2) presents the comparison of the calculation times between the original algorithm and the modified one depending on the constant k of the best rules in the fuzzy sets. As can be seen in the case of the 15 rules, it is possible to obtain the first solution for more than half of the genes using only 1.12% of the original calculation time. For 25 best rules, the calculation time is reduced to 4.9% of the exhaustive search time. Thus, we can obtain the best results for three-fourths of the analyzed genes.

Table 2 Results for the time optimized algorithm

Gene	**5 best rules**	**10 best rules**	**15 best rules**	**20 best rules**	**25 best rules**
	k = 5	**k = 10**	**k = 15**	**k = 20**	**k = 25**
	time: 0.1%	**time: 0.39%**	**time: 1.12%**	**time: 2.55%**	**time: 4.9%**
SIC1	1	1	1	1	1
CLN1	24	11	1	1	1
CLN2	2	2	1	1	1
CLN3	236	18	18	18	5
SWI4	3359	1047	12	12	10
SWI6	293	23	1	1	1
CLB5	121	121	121	121	121
CLB6	7	7	2	1	1
CDC6	1	1	1	1	1
CDC20	4	4	1	1	1
CDC28	58579	12313	722	246	49
MBI1	14	1	1	1	1

References

1 Sokhansanj B., Fitch J., Quong J., Quong A. Linear Fuzzy Gene Network Models Obtained from Microarray Data by Exhaustive Search 5:108 doi: 10.1186/1471-2105-5-108 BMC Bioinformatics, 2004,

2 Fitch J., Sokhansanj B. Genomic Engineering: Moving Beyond DNA Sequence to Function Proc IEEE 88:1949-1971, 2000