GMR
Genetics and Molecular Research
ISSN 16765680

Gene regulatory network identification from the yeast cell cycle based on a neuro-fuzzy system

B.H. Wang, J.W. Lim and J.S. Lim

Artificial Intelligence Lab, Computer Science Department, IT College,
Gachon University, Seongnam, South Korea

Corresponding author: J.S. Lim
E-mail: jslim@gachon.ac.kr

Genet. Mol. Res. 15 (3): gmr.15039002
Received July 21, 2016
Accepted August 1, 2016
Published August 30, 2016
DOI http://dx.doi.org/10.4238/gmr.15039002

ABSTRACT. Many studies exist for reconstructing gene regulatory networks (GRNs). In this paper, we propose a method based on an advanced neuro-fuzzy system, for gene regulatory network reconstruction from microarray time-series data. This approach uses a neural network with a weighted fuzzy function to model the relationships between genes. Fuzzy rules, which determine the regulators of genes, are very simplified through this method. Additionally, a regulator selection procedure is proposed, which extracts the exact dynamic relationship between genes, using the information obtained from the weighted fuzzy function. Time-series related features are extracted from the original data to employ the characteristics of temporal data that are useful for accurate GRN reconstruction. The microarray dataset of the yeast cell cycle was used for our study. We measured the mean squared prediction error for the efficiency of the proposed approach and evaluated the accuracy in terms of precision, sensitivity, and F-score. The proposed method outperformed the other existing approaches.

Key words: Gene regulatory network; Neuro-fuzzy network; Microarray; Gene selection

INTRODUCTION

Many studies have been undertaken to reconstruct a gene regulatory network (GRN), using time-series gene data, to discover the dynamic and uncertain functional relationships among genes. In particular, these studies have been further advanced with the progress in high-throughput technology. However, these biological experiments are either sensitive to the experimental conditions or difficult to accurately analyze low gene expression values (Cheng et al., 2013). As such, many studies have been conducted using machine learning or statistical methods. These studies have strived to reconstruct GRNs, using Boolean networks, Bayesian networks, and neuro-fuzzy algorithms, among other methods (Manshaei et al., 2012).

GRN reconstruction, using a Boolean network, is one of the most popular methods. It reconstructs GRN, using binary elements representing gene expression data. Consequently, it is limited in terms of the potential loss of information for gene interactions (Mehra et al., 2004), even if GRN can be reconstructed simply.

The method that uses a Bayesian network reconstructs GRNs depending on the probability of interactions among genes. The limitation of this method is that it cannot reveal if a cycle is involved in the network; the cyclic interaction of genes happens frequently in biological systems (Kim et al., 2003).

The advantage of the neuro-fuzzy algorithms is that sufficient interactions can be extracted based on rules, even with a small number of samples. However, the rules increase exponentially with a slight increase in the number of samples or genes, which leads to high calculation costs (Manshaei et al., 2012).

This paper proposes a GRN reconstruction method using a neural network, with weighted membership functions (NEWFM), which was reported by Lee and Lim (2011). Our suggested approach has the following characteristics: i) It is not constrained by the number of rules, because we incorporate the weighted membership function (Son et al., 2015). ii) The original gene expression data are preprocessed to extract the features that can best find the time-series interactions. iii) A gene that will give the best effect and another that will give the worst effect to predict the target gene are selected among input genes at each learning iteration. The two selected genes are used to classify activators and repressors. The best or worst effect is calculated by the sum of the weighted membership functions.

The cost for calculation is independent of the number of samples because of the first characteristic. The second and third characteristics provide the advantage of constructing a highly accurate GRN.

Time-series gene data from the yeast cell were used for the GRN reconstruction. The mean square error (MSE) was measured with cdc15, cdc18, and alpha datasets after determining the GRN accuracy. The sensitivity, precision, and F-score were calculated after GRN reconstruction. The proposed algorithm showed better performance than the other algorithms, in terms of sensitivity and F-score. The remainder of this paper is as follows: We propose the GRN reconstruction algorithm in section 2. We present the experimental results to demonstrate the reliability and improved contributions of our algorithm, in terms of mean square error (MSE), sensitivity, and F-score, in section 3. Our conclusions are presented in section 4.

MATERIAL AND METHODS

This section describes the method for modeling GRN. The proposed approach has four stages, which are illustrated in Figure 1.

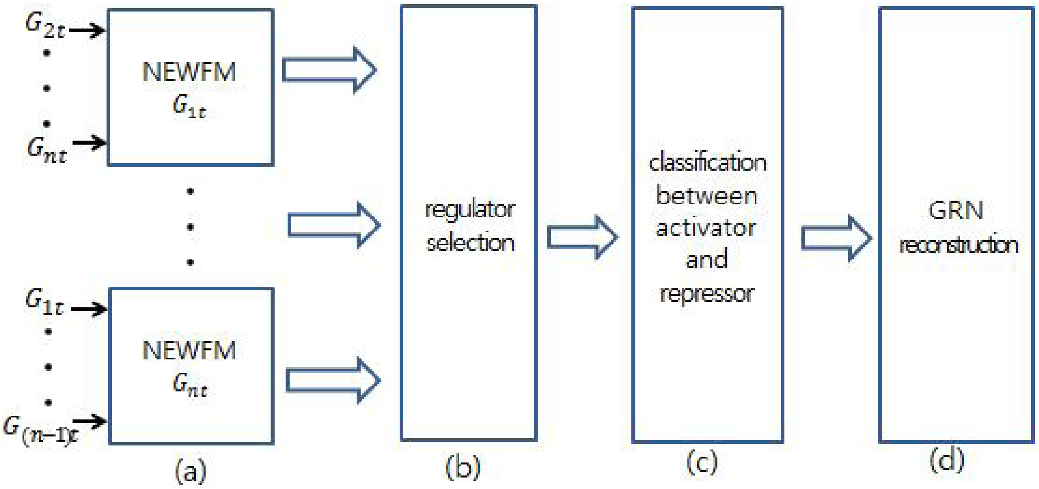

Figure 1. Proposed GRN reconstruction method. In G_{nt}, n means the number of genes and t means time point. Stage (a) is the learning step using NEWFM. If G_{1t} is the target gene, G_{2t} to G_{nt} become input genes. Every gene becomes the target gene by turn. The learning is performed by the number of genes.

The genes were learned for the target gene prediction in stage (a). Regulators were selected in stage (b) after the learning was finished for all target genes. The activators and repressors were classified in stage (c), using the proposed procedure. Finally, GRN was reconstructed. Figure 1 is described as follows:

(a) Learning stage

The genes were learned for the target gene prediction in stage (a). Gene data from *Saccharomyces cerevisiae* (yeast) were used to reconstruct GRN. The yeast cell cycle data had four types of datasets: alpha, cdc28, cdc15, and elu. The dataset used for learning in this experiment was cdc15, which comprised 12 genes and 24 time points for each gene. Alpha and cdc28 were used for the test datasets comprising 18 and 17 time points, respectively.

Figure 2 provides details of stage (a). It shows the learning of target gene G_1, where n denotes the number of genes and t indicates time point. Each n gene is respectively used as the target gene in turn. Thus, the learning stage is performed n times. It proceeds in two steps.

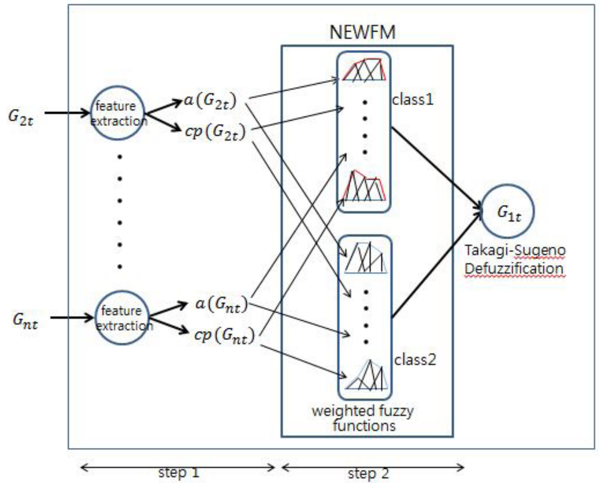

Figure 2. Learning stage. Input genes are learned for target gene prediction.

The original gene expression data were preprocessed in step 1 of stage (a), to extract the features that could best find the time-series interactions. Two kinds of features were extracted from the original data, using the following formulae.

$$a(G_{it}) = \frac{G_{i(t\text{-}1)} + G_{it}}{\sqrt{2}} \qquad t = 2, 3, \ldots.., 24 \qquad \text{(Equation 1)}$$

$$\mathrm{cp}(G_{it}) = \frac{G_{it} \text{-} \left(\frac{G_{i(t-1)} + G_{i(t-2)}}{2}\right)}{\frac{G_{i(t-1)} + G_{i(t-2)}}{2}} \qquad t = 3, 4, \ldots.., 24 \qquad \text{(Equation 2)}$$

G_i is *i*th gene. Feature *a* uses the result of the first-level Haar wavelet to reflect the gene state of the prior time point in the current time point; whereas, the feature *cp* uses the values from 2 prior time points to reflect the impact of the past on the present (Son et al., 2015). We could find more accurate gene-to-gene interactions through these features, because the past and current states of the gene were reflected in order to predict the target gene.

The learning in step 2 was performed using NEWFM, which is a neuro-fuzzy system with a weighted fuzzy membership function (Lee and Lim 2011; Son et al., 2015). Three fuzzy functions per input feature were created in two classes in NEWFM. The heights and widths of three fuzzy functions were adjusted by weight during the learning.

$$class(G_{it}) = \begin{cases} 1, & if \quad G_{it} < average\ (G_i) \\ 2, & otherwise \end{cases} \qquad \text{(Equation 3)}$$

The class was determined by the target gene expression value obtained from Equation 3, where G_{it} denotes target gene *i* of time point *t*; *t* was from 1-24 in this experiment (Soinov et al., 2003). If G_{it} was smaller than the average of all time point values of G_i, then the class was 1. This indicated a lower than average expression. If it was the same as the overall average or higher, then the class was 2; this indicated a higher expression than the average.

A fuzzy function, which is the sum of the three weighted fuzzy functions, was determined as the result of learning; it was referred to as the bounded sum fuzzy function. Red and blue line functions in step 2 were the bounded sum fuzzy functions of two classes. Three weighted fuzzy functions were simplified as the bounded sum fuzzy function. That meant that rules were simplified. The fuzzy values of the bounded sum fuzzy function were defuzzified, using the Takagi-Sugeno method (Takagi and Sugeno, 1985).

(b) Regulator selection

The two features, which gave the best effect and the worst effect for predicting the target gene, were selected by two bounded sum fuzzy functions at each iteration of learning. The information about the number selected, as the feature giving the best effect or the worst effect, was stored in the two variables. We assigned the names *H* and *L* to the variables, respectively. *H* and *L* were used for finding regulators.

Input features were ranked by Equation 4 in Stage (b). The number of learning iterations was represented by *r*, $H(IF_i)$ was *H* value of input feature *i*, and $L(IF_i)$ was L value of input feature *i*. *ED* was the extent of the effect that the input feature gave to the target gene. A greater *ED* value was indicative of a better effect.

$$ED = \frac{H(IF_i) - L(IF_i)}{r} \quad \text{(Equation 4)}$$

Figure 3 shows the procedure for selection of regulators. Input feature ranking for a target gene was dependent upon *ED*. The calculated ED ranking resulted in removal of half of the features to a small *ED* order, because they were regarded as noise. The remaining features for all target genes were merged, and then a second rank was determined, depending on *ED*. The second ranked features were divided into 3 parts. The first part meant that a regulator and a target gene moved with the same pattern over the course of time. In other words, the input gene changed in a pattern similar to the change in the target gene expression value at each time point. The last part meant that the regulator gene expression decreased, but the target gene expression increased. The middle part meant that a regulator and target gene moved in an irregular pattern over the course of time.

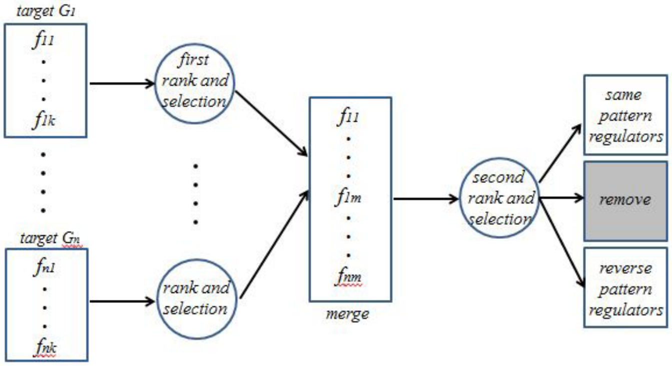

Figure 3. Regulator selection process, where k is the number of input features, n is the number of genes, and m means remaining features for each target gene, after deleting noise features.

A change in the same direction or in reverse meant that there was high predictability. Meanwhile, the change of the input gene, in accordance with the change in the target gene, lacked regularity in the irregular pattern section. The prediction capability was low; therefore, we removed the middle part features. The remaining regulators were final.

(c) Classification between activator and repressor

In Stage (c), the following process was undertaken to identify the activator and repressor genes in the discovered regulators.

Process: Identification of activator or repressor genes

```
// RG[]: regulator
// TG[]: target gene
// VRG[]: variable to save increasing or decreasing state of next time point value, compared with prior time point value of the regulator gene
// VTG[]: variable to save increasing or decreasing state of next time point value, compared with prior time point value of the target gene
// STP: the number of identical values in VRG[] and VTG[]
// tp: the number of time points
// t: current time point
// TS: threshold to classify activators and repressors
$t = 2$
$VRG[t-1] = 2$
$VTG[t-1] = 2$
While $t <= tp$
{
    If $RG[t] >= RG[t-1]$
        $VRG[t] = 2$
    Else
        $VRG[t] = 1$

    If $TG[t] >= TG[t-1]$
        $VTG[t] = 2$
    Else
        $VTG[t] = 1$
}

$t = 1$
While $t <= tp$
{
    If $VRG[t] == VTG[t]$
        $STP = STP + 1$
}
If $STP >= TS$
    RG is activator of TG
Else
    RG is repressor of TG
```

Variables, from lines 1-15 of the process, memorized the increase or decrease state in each time interval for the regulator gene and target gene. The variable, that denoted the state of increase or decrease, was marked as 1 when the expression value decreased in the next time interval. If the expression value increased, then the value was marked as 2. The variable that stored the state of increase or decrease for the regulator was *VRG*; the variable that stored the state of increase or decrease for the target gene was *VTG*.

The number of occurrences, from line 17-22, when *VRG* and *VTG* were the same for a time point, was counted and stored in variable *STP*. This variable represented the number of intervals in which the changes in the regulator gene and target gene were the same. If *STP* was the same as or greater than *TS*, as determined from lines 23-26, then gene *RG* was determined to be an activator; otherwise, it was a repressor. The activator and repressor classes were distinguished, based on the number of intervals with the same changes.

RESULTS

The results of the reconstruction and prediction are now presented, and the proposed method is compared with other approaches. The results are then discussed.

Prediction

The next step, after completing learning for all target genes, was to define error criterion to test for the prediction accuracy of the data set of the remaining genes. We measured the prediction error using the method presented in Equation (5), which was also used by Manshaei et al. (2012).

$$E = \frac{1}{N}\sum_{i=1}^{N}(G_{ip} - G_{ir})^2 \qquad \text{(Equation 5)}$$

where N is the number of time points, G_{ip} is the predicted value, and G_{ir} is the real value of the gene.

Cdc15 was used for learning. Equation 5 was applied to calculate the mean square error (MSE), in which the Takagi-Sugeno value (Takagi and Sugeno, 1985) acquired from learning was used as a prediction value. Cdc28 and alpha datasets were used for the test.

Table 1 shows MSE for the three datasets. The alpha dataset shows the smallest average MSE for all genes. This meant that this dataset was the one that was best predicted using the proposed method.

Table 1. MSE of three datasets.

Genes	Datasets		
	cdc15	cdc28	alpha
SIC01	0.55	0.1614	0.2682
CLB05	0.1182	0.2546	0.0917
CDC20	0.3646	0.4475	0.2129
CLN03	0.1437	0.1336	0.0990
SWI06	0.1740	0.0396	0.0311
CLN01	0.5547	0.2696	0.3782
CLN02	0.3290	0.2711	0.1869
CLB06	0.6360	0.6632	0.4390
CDC28	0.0538	0.0629	0.0643
MBP01	0.5168	0.0715	0.06
CDC06	0.1815	0.1786	0.1192
SWI04	0.0461	0.1421	0.4271
Average	0.3057	0.2246	0.1981

GRN Reconstruction

The activators and repressors were discovered by the GRN reconstruction method, as proposed. Figure 4 shows the reconstructed GRN. We evaluated the present GRN, by comparing it to the GRN established in the Kyoto Encyclopedia of Genes and Genomes (KEGG) database (Manshaei et al., 2012; HTML). The KEGG database has a tremendous amount of information for S. cerevisiae cell cycle regulatory protein-DNA interactions (Kanehisa et al., 2010).

Black arrow lines in Figure 4 represent activators, while red circle lines represent repressors. Thick solid lines denote correct interactions, while dashed lines mean incorrectly extracted interactions. Thin lines represent interactions that exist in the KEGG database, but were not found during reconstruction. The grey, solid, thin lines refer to interactions that were found in this approach, but do not exist in KEGG.

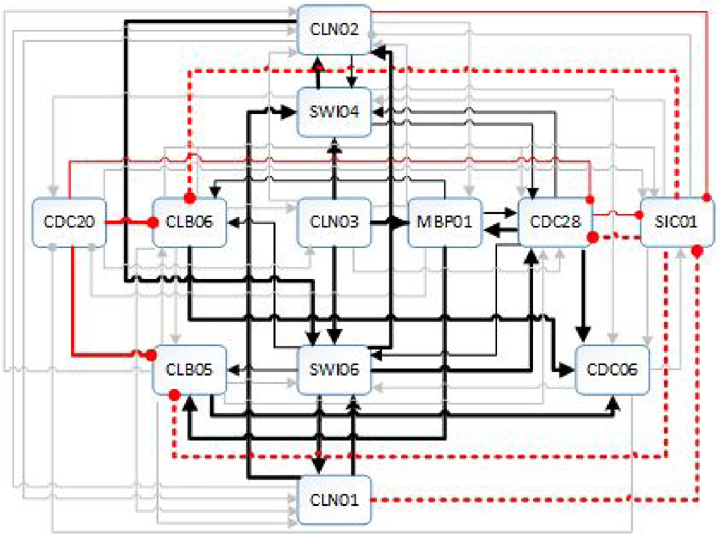

Figure 4. Reconstructed GRN.

DISCUSSION

We evaluated the proposed approach by comparing HRBNF (Manshaei et al., 2012) and Time-delay ARACNE (I et al., 2007). HRBNF is the hybrid rule-based neuro-fuzzy network proposed by Manshaei et al. (2012); it is used for extracting GRN information from gene expression data. Time-delay ARACNE uses a three step algorithm, based on the statistical way of time course gene interaction, for reconstructing GRN (Zoppoli et al., 2010). HRBNF and Time-delay ARACNE showed the best performances in Manshaei et al. (2012), when they extracted activators and repressors from genes. Thus, we selected these two algorithms as comparison targets. We compared three factors of sensitivity, precision, and F-score, which were computed from the following formulas (Manshaei et al., 2012):

$$\text{Sensitivity} = \frac{TP}{TP + FN} \qquad \text{(Equation 6)}$$

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{(Equation 7)}$$

$$\text{F-Score} = \frac{1}{\alpha\left(\frac{1}{Precision}\right) + (1-\alpha)\left(\frac{1}{Sensitivity}\right)} \qquad \text{(Equation 8)}$$

The true-positives (TPs) in Equations 6 and 7 represent the accurately discovered activators or repressors, whereas the false-positives (FPs) represent the inaccurately discovered activators or repressors. A false negative (FN) represents an undiscovered regulator. We assume to be 0.5 in Equation 8, which indicates that precision and sensitivity are given equal weights (Manshaei, et al., 2012).

Table 2 shows a comparison of the sensitivity, precision, and F-score measures, according to the threshold. We used the number of time intervals, in which a regulator and the target gene had the same status, as the threshold that determined whether the selected gene was the activator or repressor for the target gene. If the number of time intervals, that have the same status of the two genes, were greater than threshold or equal to threshold, then the selected gene is an activator; however, if the selected gene did not fall in this range, then it was a repressor. We calculated the sensitivity, precision, and F-score when the threshold ranges were from 8-12. As a result, if it had a median value of 10, then the three values were the highest.

Table 2. Result comparison according to threshold classifying activators and repressors.

Criteria	Threshold: The number of identical statuses among 23 time intervals of regulator and target gene				
	8	9	10	11	12
TP	16	17	17	16	15
FP	5	4	4	5	6
FN	17	16	16	17	18
Sensitivity	48.4%	51.5%	51.5%	48.4%	45.5%
Precision	76.2%	81%	81%	76.2%	71.4%
F-score	59.3%	63%	63%	59.3%	55.6%

Table 3 shows a comparison of the results obtained with the proposed algorithm and the two other algorithms. The number of false positives, for the proposed algorithm, was larger than that of the other algorithms. Nevertheless, several true positives existed, which indicated that the FN decreased. This suggested that the proposed algorithm improved the discovery of correct regulators. Consequently, the sensitivity and F-score were higher than for the other algorithms. This result can be explained by the fact that the NEWFM, used in the proposed algorithm, determined the most optimal and simple rules through the weighted fuzzy functions on its own. It extracted the features that best reflected the dynamic features of the time series for learning.

Table 3. Comparison of proposed algorithm and other algorithms.

Algorithms	Criteria					
	TP	FP	FN	Sensitivity	Precision	F-score
Time delay-ARACNE	10	2	23	30.3%	83.3%	44.4%
HRBNF	13	3	20	39.4%	81.3%	53.1%
The proposed approach	17	4	16	51.5%	81%	63%

CONCLUSIONS

These data show that we learned yeast cell time-series gene data using NEWFM, based on the weighted neuro-fuzzy function for GRN reconstruction. Two kinds of features were extracted from the original dataset through gene preprocessing; this information was used for learning. Next, genes with the best or worst effects on the target genes were selected, depending on the degree of effect. The activators and repressors were distinguished according to the number of time points, in which the finally discovered regulator and target gene showed the same state. The GRN was then constructed. We used the Takagi-Sugeno value to predict the target gene expression value, once we identified the repressors and activators.

The proposed method was compared with two existing algorithms, based on precision, sensitivity, and F-score. The results showed that sensitivity increased by approximately 25%. The F-score increased by approximately 20%, compared to those of HRBNF in the proposed method. The discovery rate for the repressors was low in the reconstruction of GRN, despite this improved performance. This indicates a need for further studies on algorithms that can improve the discovery rate of repressors.

ACKNOWLEDGMENTS

Research supported by Basic Science Research Program through the National Research Foundation of Korea (NRF), which was funded by the Ministry of Education, Science and Technology (#2015R1D1A1A09057409).

REFERENCES

Cheng C, Fu Y, Shen L and Gerstein M (2013). Identification of yeast cell cycle regulated genes based on genomic features. *BMC Syst. Biol.* 7: 70 http://dx.doi.org/10.1186/1752-0509-7-70.

Kanehisa M, Goto S, Furumichi M, Tanabe M, et al. (2010). KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Res.* 38: D355-D360 http://dx.doi.org/10.1093/nar/gkp896.

Kim SY, Imoto S and Miyano S (2003). Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Brief. Bioinform.* 4: 228-235 http://dx.doi.org/10.1093/bib/4.3.228.

Lee S-H and Lim JS (2011). Forecasting KOSPI based on a neural network with weighted fuzzy membership functions. *Expert Syst. Appl.* 38: 4259-4263 http://dx.doi.org/10.1016/j.eswa.2010.09.093.

Manshaei R, Sobhe Bidari P, Aliyari Shoorehdeli M, Feizi A, et al. (2012). Hybrid-controlled neurofuzzy networks analysis resulting in genetic regulatory networks reconstruction. *ISRN Bioinform.* 2012: 419419 http://dx.doi.org/10.5402/2012/419419.

Mehra S, Hu WS and Karypis G (2004). A Boolean algorithm for reconstructing the structure of regulatory networks. *Metab. Eng.* 6: 326-339 http://dx.doi.org/10.1016/j.ymben.2004.05.002.

Soinov LA, Krestyaninova MA and Brazma A (2003). Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biol.* 4: R6 http://www.biomedcentral.com/content/pdf/gb-2003-4-1-r6.pdf. http://dx.doi.org/10.1186/gb-2003-4-1-r6

Son SY, Lee SH, Chung KY and Lim JS (2015). Feature selection for daily peak load forcasting using a neuro-fuzzy system. *Multimedia Tools Appl.* 74: 2321-2336 http://dx.doi.org/10.1007/s11042-014-1943-0.

Takagi T and Sugeno M (1985). Fuzzy identification of systems and its applications to modeling and control. *System Man Cybernetics IEEE Trans*. SMC-15: 116-132 http://dx.doi.org/10.1109/TSMC.1985.6313399.

Zoppoli P, Morganella S and Ceccarelli M (2010). TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics* 11: 154 http://www.genome.jp/kegg/ http://dx.doi.org/10.1186/1471-2105-11-154. PubMed